OR Spektrum (1995) 17:125–137

OR Spektrum

Scatter search and star-paths: beyond the genetic metaphor

Fred Glover

School of Business and Program of Applied Mathematics, University of Colorado, Campus Box 419, Boulder, CO 80309-0419, USA

Received: 2 September 1993 / Accepted in revised form: 17 November 1994

Abstract. Scatter search and genetic algorithms have originated from somewhat different traditions and perspectives, yet exhibit features that are strongly complementary. Links between the approaches have increased in recent years as variants of genetic algorithms have been introduced that embody themes in closer harmony with those of scatter search. Some researchers are now beginning to take advantage of these connections by identifying additional ways to incorporate elements of scatter search into genetic algorithm approaches. There remain aspects of the scatter approach that have not been exploited in conjunction with genetic algorithms, yet that provide ways to achieve goals that are basic to the genetic algorithm design. Part of the gap in implementing hybrids of these procedures may derive from relying too literally on the genetic metaphor, which in its narrower interpretation does not readily accommodate the strategic elements underlying scatter search. The theme of this paper is to show there are benefits to be gained by going beyond a perspective constrained too tightly by the connotations of the term "genetic". We show that the scatter search framework directly leads to processes for combining solutions that exhibit special properties for exploiting combinatorial optimization problems. In the setting of zero-one integer programming, we identify a mapping that gives new ways to create combined solutions, producing constructions called *star-paths* for exploring the zero-one solution space. Star-path trajectories have the special property of lying within regions assured to include optimal solutions. They also can be exploited in association with both cutting plane and extreme point solution approaches. These outcomes motivate a deeper look into current conceptions of appropriate ways to combine solutions, and disclose there are more powerful methods to derive information from these combinations than those traditionally applied.

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

Zusammenfassung. Scatter Search (gestreute Suche) und genetische Algorithmen weisen eine Anzahl einander komplementärer Eigenschaften auf. Trotz verschiedenen Ursprungs haben sich in den letzten Jahren, insbesondere auch aufgrund zahlreicher Modifikationen genetischer Verfahren, zunehmend mehr Gemeinsamkeiten herausgeschält, die in erster Linie auch durch die Übertragung von Scatter Search Features in genetische Algorithmen entstanden. Einige grundlegende Aspekte von Scatter Search sind bisher jedoch in genetischen Algorithmen – im engeren Sinne – nicht berücksichtigt. Es zeigt sich, daß mittels Scatter Search Kombinationen von Lösungen generiert werden können, deren Eigenschaften entscheidend die kombinatorische Struktur der zugrundeliegenden Optimierungsprobleme widerspiegeln. Im Falle binärer Optimierungsprobleme werden durch Projektionen Lösungen zu sog. Sternpfaden (star-paths) kombiniert, von denen aus jeweils optimale Lösungen erzeugt werden können. Mögliche Ergänzungen durch Schnittebenen zur Exploration des Lösungsraumes legen nahe, der Kombination von Lösungen (vgl. etwa die Rekombination bei genetischen Algorithmen) zur Erzeugung problemspezifischen Wissens mehr Aufmerksamkeit zu schenken als bisher.

Key words: Heuristics, integer programming, genetic algorithms, scatter search

Schlüsselwörter: Scatter search (gestreute Suche), genetische Algorithmen, Sternpfade, Projektion

1. Introduction

The metaphor underlying genetic algorithm (GA) approaches has proved a significant element in their success, and has inspired many useful insights into problem solving. Nevertheless, there are also limitations to this metaphor that inhibit comprehension of key possibilities that are rooted in another tradition for combining population elements, whose beginnings are roughly contemporaneous with those of genetic algorithms. Scatter search, together with its more recent manifestations in path relinking and structured combination strategies, offers a versatile machinery for manipulating vectors and influencing their evolution. Those familiar with the genetic tradition are finding these ideas harmonious with their own.

Newer descendants of genetic algorithms, represented by "parallel genetic algorithms" and "genetic local search", are beginning to incorporate ideas proposed earlier by these alternative approaches, and are coming to embody an evolution that goes somewhat beyond the connotations of the term "genetic". Researchers in the GA community are increasingly recognizing the limitations of the genetic terminology, and are offering alternatives, in some cases making analogies to higher kinds of evolution, such as social or cultural.

The purpose of this paper is to identify important concepts and strategies not yet exploited in the genetic tradition, either in its original or modern form. Useful possibilities exist for developing more effective solution approaches, marrying genetic notions with the complementary framework that derives from scatter search. The power of this framework is indicated by recent computational studies, and we give new theoretical results for carrying this framework further.

In the domain of discrete optimization, we introduce the concept of *star-paths* for linking solutions, based on a class of projections called *directional rounding*. In the scatter search orientation these star-paths may be conceived as creating higher forms of solution combinations (or, in the genetic tradition as establishing higher forms of parent-offspring relationships). We prove these paths are capable of accessing feasible and optimal solutions from any vertex of a polyhedron that circumscribes the feasible space, and hence they yield strategies capable of exploiting discrete optimization relaxation approaches based on linear and convex programming. We also establish connections with cutting plane methods, and observe that parallel computation can be applied advantageously to implement the strategies that arise from our framework.

2. Scatter search and genetic algorithms

Scatter search and genetic algorithms derive from different foundations and perspectives which, viewed with the benefit of hindsight, exhibit several interesting elements in common. Parallels between the methods are more visible today than when these approaches originated in the 1970s, due to the fact that genetic algorithms have undergone a number of critical changes that have given them a character more closely in harmony with scatter search. Researchers from different traditions have begun to notice connections between these approaches, and to establish useful strategies that exploit reinforcing elements of these approaches (Michaelwicz et al. (1991); Mühlenbein (1992); Michalewicz (1993); and Reeves (1993 b)). Hybrid procedures incorporating related ideas have also been developed by Battiti and Tecchiolli (1993); Costa (1992); Dorndorf and Pesch (1995) and Moscato (1993). One of the themes of this paper is that the potential exists to exploid such connections more fully.

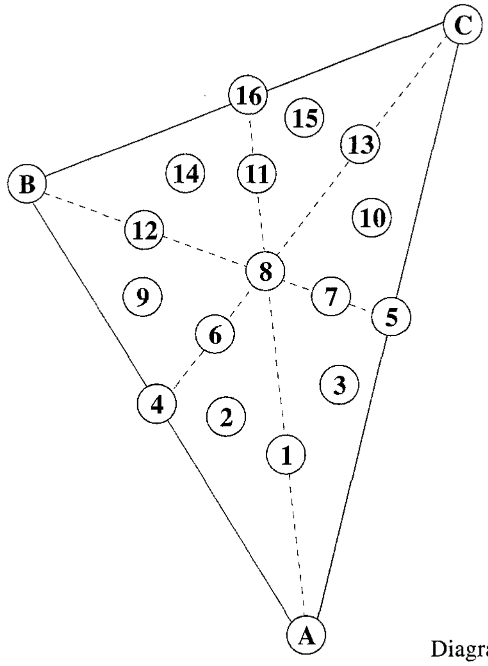

Diagram 1

Because of the widespread coverage of genetic algorithms (e.g., see Goldberg (1989); Davis (1991); Reeves (1993 a) and Whitley (1993)), we will omit an introductory review of this area, and will instead focus on the elements of scatter search, and how they interrelate with GA proposals. The first part of our development draws on Glover, Kelly and Laguna (1992).

Scatter search

Scatter search is designed to operate on a set of points, called reference points, that constitute good solutions obtained from previous solution efforts. The approach systematically generates linear combinations of the reference points to create new points, each of which is mapped into an associated point that yields integer values for variables required to receive such discrete values (Glover 1977). As originally proposed, the mapping consists of rounding or an associated generalized adjacency process, e.g., rounding a selected discrete variable to an integer neighbor, then determining implied value changes for other variables, and repeating.

This idea of using a systematic process to enable solution combinations to meet desired restrictions embodies the principle that such combinations should be influenced by context. (Genetic algorithm concepts proposed at this time, and advocated in some circles even today, instead embrace the notion that combinations should be generated without reference to context.) The adaptive rounding process of scatter search automatically satisfies simple constraints such as mutual exclusivity and precedence relationships, since each rounding step, when carried out by standard updating processes, yields only those remaining options consistent with choices made earlier. (More complex constraints sometimes also can be satisfied this way by updating a linear programming basis representation, as characteristically done in integer programming implementations.)

The vectors that result from the rounded linear combinations of the chosen reference points in turn are allowed to serve as inputs to accompanying heuristic processes. The heuristic procedures then transform these inputs into improved outcomes, thereby bringing the appoach full circle. These outcomes accordingly are screened to provide a new set of reference points, and the process starts again.

By this approach, linear combinations produced at each stage are dispersed across region whose form is biased by the distribution of reference points. Diagram 1 (Glover 1977) shown on the previous page illustrates a simple version of the process. Each of the points numbered 1 through 16 in Diagram 1 is the central point of an apparent subregion of the enveloping region A, B, C. The points A, B and C may or may not constitute the original reference points. (For example, the original points may consist of 5, 7 and 11, or of 4, 5, 12 and 13.) Thus, new points may be created that are not convex combinations of original points, and hence that may contain information that is not contained in these points, in the sense of bits implicit in a solution representation. (At the same time, the original points are also instances of such linear contributions, and hence they are likewise included among the candidate outcomes.)

The mappings that progressively round the resulting linear combinations, modifying fractional components that are required to be discrete, can introduce additional information derived from relationships between problem variables, hence reflecting the influence of problem structure. Problem structure exerts further influence by means of the heuristic processes that take these points as inputs and produce new solutions from them.

Similarities are immediately evident between this approach and the GA formulation of Holland (1975). Both are instances of what are sometimes called "population based" procedures, which start with some collection of elements and progressively evolve those elements to yield new ones that are subjected to the same guiding process. Both also incorporate the idea that a key aspect of producing the new elements is to generate some form of combination of the existing elements. On the other hand, several contrasts between the methods also may be noted. The early GA approaches were predicated on the idea of choosing parents randomly to produce offspring, and further on introducing randomization to determine which components of the parents should be combined (by genetic crossover operations). By contrast, no corresponding recourse to randomization is made in the scatter search approach, although nothing excludes its use as a bias factor (i.e., probabilistically favoring evaluation criteria that would otherwise be applied deterministically). Attention is focused in scatter search on choosing good solutions as a basis for generating combinations, in contrast to the more democratic GA policy of allowing solutions of all types to be combined. This scatter search focus can enhance the generation of relevant outcomes without losing the ability to produce diverse solutions, due to the way the generation process is implemented. Recent "elitist" GA variants also give preference to combining good elements, but without a corresponding ability to produce combinations beyond the region in which these elements lie, except by resorting to auxiliary strategies to overcome this limitation. (Scatter search does not need "mutation" to avoid convergence to a population from which an optimal solution is not reachable.) The different mechanisms used by the two approaches to combine solutions, consisting of rounded linear combinations on one hand and genetic crossover on the other, particularly invite examination.

Significance of rounded linear combinations

Linear combinations provide a somewhat more varied set of possibilities for creating new solutions than crossover as initially introduced in GAs. They also avoid the artificiality of resorting to the binary representations which were the foundation of the original genetic encoding and crossover notions. To see the relevance of this, consider the goal of creating integer solutions that are combinations of the two solutions x=9 and x=26. Rounded linear combinations can generate every integer point from minus to plus infinity on the line joining x=9 and x=26, hence in this case yielding every value x may feasibly be assigned. On the other hand, when these solutions are given a binary representation, (0 1 0 0 1) for x=9, and (1 1 0 1 0) for x=26, then the possible outcomes are substantially more limited. In particular, the only ways to create rounded linear combinations of the binary vectors (0 1 0 0 1) and (1 1 0 1 0) yield the collection of binary vectors (* 1 0 * *) where the "* elements" can be 0 or 1. Hence instead of producing all possible integer points these combinations produce only the integer values of x satisfying $8 \le x \le 11$ and $24 \le x \le 27$.

The possible outcomes are more limited still if genetic crossover is used in the form initially conceived, as sometimes espoused by GA traditionalists. The only vectors that can be created by the proposals of Holland are the four vectors (0 1 0 0 0), (0 1 0 1 0), (1 1 0 0 1), (1 1 0 1 1), corresponding to x = 8, 10, 25, 27. When attention is restricted to binary vectors, which clearly is inappropriate, rounded linear combinations in fact give the same set of possibilities as the "uniform" crossover operator proposed some years later by Ackley (1987) (although the randomized means of generating these possibilities in the GA setting will typically yield different outcomes from the strategic rounding approach of scatter search). By further employing generalized adjacency rounding, where values of some variables may change as a result of modifying others, additional possibilities result.

The significance of rounding to account for interactions between variables is illustrated by the following example (Glover (1964)). Consider the simple integer programming problem

Minimize $9x_1 + 4x_2 + 8x_3$

subject to

$$
\begin{aligned}
&9x_1 - 8x_2 - x_3 \ge 7\\
&-6x_1 + 7x_2 - 2x_3 \ge 6\\
&-x_1 - x_2 + 5x_3 \ge 9\\
&x_1,\ x_2,\ x_3 \ge 0 \text{ and integer.}
\end{aligned}
$$

The linear programming (LP) solution to this problem, which disregards the integer requirement for the variables, gives a solution vector $x=(x_1, x_2, x_3)=(24.43, 25.14, 11.71)$. Successive rounding that respects interactions between the variables (implied by the inequality constraints above, and also manifested in the structure of the LP basis inverse), yields a solution vector $x=(29, 30, 14)$, which turns out to be optimal for this problem. Evidently, the integer values of this final vector could not be anticipated without accounting for the interdependencies among the problem variables. Additional manifestations of such phenomena and processes for exploiting them are given in Nemhauser and Wolsey (1988) and Parker and Rardin (1988), and represent the types of processes that are accommodated naturally within the scatter search framework.

Without contradicting the importance of randomization in GA processes, the fact that scatter search seeks to create new points strategically rather than randomly can represent a useful feature in some settings. The points of Diagram 1, for example, may be generated and scanned in their indicated numerical order, under the condition where this order reflects a ranking determined by the objective function, or more generally by a feasible direction gradient. Scatter search does not prespecify the number of points it will generate or retain, since this can be established adaptively by considering the quality or structure of solutions produced in such a systematic generation.

Scatter search and early GA approaches may also be distinguished by the fact that the reference points are supplied by and in turn supply another heuristic process. This is an orientation that has lately gained strong proponents among a core of researchers in the area of optimization who are seeking to modify GA proposals to make them more effective, particularly as advocated in the Parallel GA approach of Mühlenbein et al. (1988) and the related genetic local search approach of Ulder et al. (1991).

Finally, we observe that the allowance for real-valued weights and vector components (for variables or parameters that are not discrete) anticipates the developing trend in some parts of the GA community to embrace "real-coded" (or floating-point) genes, as represented by the work of Davis (1991), Bäck, Hoffmeister and Schwefel (1991) and Eschelman and Schaffer (1992). Additional connections with current GA developments are provided in the study of Michalewicz et al. (1991), which introduces a nonstandard GA approach using linear combinations in place of genetic crossover.

In this way, the philosophical themes of scatter search and genetic algorithms are being brought closer together by modern efforts to create GA variants with an improved ability to solve optimization problems. These developments have set the stage for the integrative research now underway, notably in the work cited at the beginning of Section 2, that is disclosing the advantage of going beyond piecemeal transformations of GA approaches and creating direct hybrids with scatter search. Relatively simple forms of scatter search have already produced improved solutions in such approaches.

The rest of this paper is devoted to showing how these concepts give a basis for strategies with appealing properties for zero-one optimization, and hence for the broad class of problems that can be represented in this domain. In particular, we will show how a special type of rounding process, called directional rounding, creates an easily implemented form of scatter search whose candidate solutions do not suffer the risk of evolving to a state where an optimal solution is inaccessible (i.e., outside the range of solution combinations made available). Moreover, we will show that this approach gives a way to combine solutions that avoids many combinations that are inherently unproductive. Finally, we show that the embodiment of the approach in the generation of constructions called star-paths yields a highly efficient and adaptive basis for its application.

3. Application to zero-one integer programming

The zero-one integer programming problem encompasses an extensive array of combinatorial applications, ranging from logical design to scheduling and routing, and from graph theory models to resource allocation and financial planning. (Examples of such applications are contained in Hu (1969); Zionts (1974); Murty (1976); Parker and Rardin (1988); Nemhauser and Wolsey (1988).)

We begin by giving a formulation for this problem that is convenient for our subsequent development, and then introduce the theoretical results relevant to exploiting this problem within the scatter search framework. Specifically, we represent the zero-one integer programming problem in the form

(**IP**) Minimize $z=\Sigma(c_j x_j: j \in N)$

subject to

$$\Sigma(A_j x_j: j \in N) = b$$
$$1 \geq x_j \geq 0 \text{ and } x_j \text{ integer} \qquad j \in I \subseteq N$$
$$U_j \geq x_j \geq 0 \qquad j \in C = N - I$$

The vectors A_j, $j \in N = \{1,\ldots,n\}$ and b are column vectors of constants. The subsets I and C of N respectively constitute the index sets for the integer (zero-one) and continuous variables. We denote the vector consisting of both integer and continuous variables by the symbol x. Solutions (x vectors) that are feasible for problem (IP) will be called IP feasible, and solutions that are feasible for the corresponding linear programming relaxations (dropping the integer requirement for the zero-one variables) will be called LP feasible.

To differentiate this problem from the mixed IP problem, we assume the values of the continuous variables are uniquely determined by the values of the integer variables. This occurs most commonly when the variables x_j, $j \in C$, represent slack variables for associated inequality constraints; that is, the vectors A_j, $j \in C$ compose an identity matrix. Then for any assignment of values to the integer variables x_j, $j \in I$, each continuous variable x_k, $k \in C$ receives the value

$$x_k = b_k - \Sigma(A_{kj} x_j: j \in I) \; k \in C. \qquad (1)$$

If all the problem data are integers, then the continuous variables implicitly receive integer values.

We allow upper bounds U_j for some or all of the continuous variables to be infinite (i.e., redundant) and stipulate that $U_j=1$ for $j\in I$, which gives $x_j\le U_j$, $j\in N$. By this convention, the preceding formulation suggests the use of the bounded variable simplex method for solving the LP relaxation (see, e.g., Dantzig (1963)), and in general as a vehicle for moving from one extreme point to another. (Thus, feasible pivot moves leading to adjacent extreme points include those that change the value of a nonbasic variable from one bound to the opposite bound, and it is unnecessary to refer to slack variables for the upper bound inequalities.)

We have formulated the problem in this manner to make it convenient to exploit this extreme point connection. In particular, it is well known that an optimal solution for the zero-one IP problem may be found at an extreme point of the LP feasible set. Special approaches have been proposed to exploit this fact (Cabot and Hurter (1968); Glover (1968); Balas and Martin (1980); Aboudi and Jörnsten (1992); Lokketangen et al. (1993); Glover and Lokketangen (1994)). Part of our development has implications for modifying and extending such approaches incorporating cutting planes. Beyond this, however, we will give ways to take advantage of extreme point representations as a foundation for constructing new heuristic search processes, creating solution combinations that are targeted to lie in a subspace where an optimal zero-one extreme point can be proved to be found.

4. Linear programming extreme point representations

Let $x(0)$ denote a current basic extreme point solution as obtained by the bounded variable simplex method, let $\{x_j: j\in NB\}$ denote the current set of nonbasic variables and let $\{x_j: j\in B\}$ denote the current set of basic variables $(B=N-NB)$. The extreme points adjacent to $x(0)$ have the form

$$x(h)=x(0)-\theta_h D_h \text{ for } h\in NB \qquad (2)$$

where D_h is a vector associated with the nonbasic variable x_h, and θ_h is the change in the value of x_h that moves the current solution from $x(0)$ to $x(h)$ along their connecting edge. The standard LP basis representation identifies the subset of entries D_{hj} of D_h associated with the current basic variables x_j. The entries of D_h for nonbasic variables x_j are automatically zero, except for x_h. We choose the sign convention for entries of D_h that yields a coefficient for x_h of $D_{hh}=1$ if x_h is currently at its lower bound, and of $D_{hh}=-1$ if x_h is currently at its upper bound. Hence x_h respectively receives the value θ_h or $U_h-\theta_h$ at the extreme point $x(h)$. By this convention, the value θ_h is always nonnegative, and is strictly positive except under degeneracy.

Our following development will identify points $x(h)$, expressed in the form of (2), based on positive values of θ_h that may differ from those that yield extreme points adjacent to $x(0)$. Such points $x(h)$, $h\in NB$, provide a foundation for generating reference points for our scatter search approach. We also will create sets of reference points that do not depend on identifying a basic LP feasible solution $x(0)$. Both of these ways of generating reference points depend on the notion of directional rounding, examined next.

5. Directional rounding in scatter search

We allow the symbols 0 and 1 to refer both to scalars and to vectors (of all 0's and all 1's) according to context. Because of the dependency of the continuous variables on the integer variables by (1), a restriction on x expressed in the form $x\in X$ will be treated as a restriction directly placed on the subvector $(x_j: j\in I)$. In accordance with this convention, we let $X(0,1)$ denote the unit hypercube defined relative to $(x_j: j\in I)$, where $x\in X(0,1)$ indicates $0\le x_j\le 1$, $j\in I$, and let $V(0,1)$ denote the vertices of the hypercube, where $x\in V(0,1)$ indicates $x_j=0$ or 1, $j\in I$.

Directional rounding, identified by the symbol δ, is a mapping from the continuous space $X(0,1)$ to the discrete space $V(0,1)$ by the following rules. We first define δ with respect to components of the vector x.

For a specified $j\in I$, assume x'_j is an arbitrary value of x_j and x^*_j is a value that satisfies $0\le x^*_j\le 1$. Then the directional rounding $\delta(x^*_j, x'_j)$ from x^*_j to (in the direction of) x'_j is given by

$$\begin{array}{r}0 \text{ if } x'_j<x^*_j\\ 1 \text{ if } x'_j>x^*_j\\ x^*_j \text{ if } x'_j=x^*_j \text{ and } x^*_j=0 \text{ or } 1\\ 0 \text{ or } 1 \text{ if } x'_j=x^*_j \text{ and } x^*_j\ne 0,\ x^*_j\ne 1.\end{array}$$

In the last case, the choice of $\delta(x^*_j, x'_j)=0$ or 1 can be based on whether x^*_j is closer to 0 or 1, breaking the tie arbitrarily for $x^*_j=0.5$. Alternative rules for making the choice between 0 and 1 when $x'_j=x^*_j$ are identified later.

We now extend the definition of δ to refer also to vectors by defining the directional rounding $\delta(x^*, x')$, from a vector $x^*\in X(0,1)$ to an arbitrary vector x', to be the point in $V(0,1)$ given by

$$\delta(x^*,x')=(\delta(x^*_j, x'_j): j\in I).$$

Our notation specifies only the effect of δ on the integer components of the vector $x=\delta(x^*,x')$, since the continuous components are determined automatically. (That is, the vector is given by $x_j=\delta(x^*_j, x'_j)$ for $j\in I$, yielding a unique resulting value for each x_k, $k\in C$ by (1).)

Finally, we identify the directional rounding $\delta(x^*,X)$ from the vector x^* to the set X to be the set of points in $V(0,1)$ given by

$$\delta(x^*,X)=\{\delta(x^*,x'): x'\in X\}.$$

It may be noted that directional rounding includes nearest neighbor rounding as a special instance. In particular, $\delta(x^*,x')$ becomes the nearest integer neighbor of x^* whenever each x'_j, $j\in I$, lies in the interval bounded by x^*_j and the integer closest to x^*_j.

The point x^* will be called the base point and the point x' will be called the focal point of the directional rounding $\delta(x^*,x')$. Our definitions apply as well as mixed integer programming problems by solving a residual linear program to determine values of continuous variables, given values for integer variables. The definitions also can readily be extended to the case where arbitrary vectors serve as base points, rather than requiring $x^* \in X(0,1)$.

We apply these ideas in the scatter search context as follows. Recall that the standard scatter search design prescribes: (a) choosing a set of reference points, (b) generating linear combinations of these points, and (c) rounding the integer components of the linear combinations to obtain new points. We specify that the set X, used to create the directionally rounded set $\delta(x^*,X)$, is the set of selected linear combinations of the reference points in this approach. Hence the mapping δ gives the transformation that rounds these linear combinations to yield new points. The following sections are devoted to specifying rules and associated theory related to the choice of reference points, the set X, and the base point and focal point pairs.

5.1. Fundamental analysis

We let X(R) denote a chosen set of reference points, indexed by the set R; i.e., $X(R) = \{x(r): r \in R\}$. Notationally, for two arbitrary points x' and x'', and a scalar λ, we identify the ray from x' though x'' by

$$\text{Ray}(x',x'') = \{x: x = \lambda x'' + (1-\lambda)x',\ \lambda \geq 0\}.$$

Also, for a set of reference points X(R), and scalars λ_r, $r \in R$, we identify the hyperplane consisting of all normalized linear combinations of these points by:

$$\text{Plane}(X(R)) = \{x: x = \Sigma(\lambda_r x(r): r \in R),\ \Sigma(\lambda_r: r \in R) = 1\}.$$

Our terminology is motivated by the fact that the points of X(R) are linearly independent in the usual case to be considered. We also identify the associated half space as

$$\text{Half\_space}(X(R)) = \{x: x = \Sigma(\lambda_r x(r): r \in R),\ \Sigma(\lambda_r: r \in R) \geq 1\}.$$

Relative to the base point x^*, which we assume does not belong to Plane(X(R)), and hence which identifies a set of affinely independent points when taken together with X(R), we define the polyhedral (half) cone spanned by the rays from x^* through the points of X(R):

$$\text{Cone}(x^*,X(R)) = \{x: x = x^* + \Sigma(\lambda_r\, x(r): r \in R),\ \lambda_r \geq 0,\ r \in R\}.$$

Finally, the set of all convex combinations of the points of X(R) identifies a face of the truncated cone that results from the intersection of Cone(x^*,X(R)) with Half_space(X(R)), i.e., in particular, the face that excludes the point x^*. We accordingly define

$$\text{Face}(X(R)) = \{x: x = \Sigma(\lambda_r x(r): r \in R),\ \Sigma(\lambda_r: r \in R) = 1,\ \lambda_r \geq 0,\ r \in R\}.$$

These definitions are relevant for exploiting the situation where x^* corresponds to an extreme point x(0) of the LP feasible region, and X(R) is a set of points on the rays (edges) from x^* through each of the adjacent extreme points x(h), $h \in NB$. X(R) can be taken to be these adjacent extreme points under conditions of nondegeneracy, but preferably will be chosen to be different from these extreme points. In general, we will state various properties governing x^* and the points of X(R), and conclusions that hold when these properties are satisfied. The following assumptions provide a foundation for these properties.

Assumptions. A1. All optimal IP solutions belong to Cone(x^*,X(R)).

A2. x^* does not belong to Half_Space(X(R)), but belongs to the complementary half space (generated for $\Sigma(\lambda_r: r \in R) \leq 1$).

A3. All optimal IP solutions, exclusing x^*, belong to Half_Space(X(R)).

A4. x^* is a dual feasible point of Cone(x^*,X(R)), relative to the objective of minimizing $z = cx$.

A result that immediately establishes a link between these assumptions is the following.[1]

Lemma 1. *Let* $x^* = x(0)$ *and define* $x(h)$, $h \in NB$ *by* (2) *for any given positive values* θ^*_h for θ_h, i.e.,

$$x(h) = x(0) - \theta^*_h D_h,\ h \in NB \tag{3}$$

Then, for the set $X(R) = \{x(r): r \in R\}$ *determined by taking* $R = NB$, *we obtain:*

$$\text{Cone}(x^*,X(R)) = \{x: x = x(0) - \Sigma(\lambda_r D_r: r \in R),\ \lambda_r \geq 0,\ r \in R\}$$

As a basis for applying the preceding result, we note that assumptions A1 and A2 hold when x^* is any LP feasible extreme point and X(R) is constructed from its adjacent extreme points by Lemma 1 (where the positive θ^*_h values of Lemma 1 may not be the same as the θ_h values that define these adjacent extreme points). Assumption A3 holds under these same conditions when Half_Space (X(R)) is a valid cutting plane that excludes x^* as feasible. Assumption A4 holds if x^* is an optimal linear programming solution that is a primal and dual feasible extreme point (e.g., as obtained by the simplex method).

To build on these observations, we note that every point on the ray from x^* through x' (excluding the point x^* itself) gives the same directional rounding as $\delta(x^*,x')$. Expressed more formally, we obtain:

Lemma 2. *For any base point* $x^* \in X(0,1)$ *and focal point* $x' \neq x^*$:

$\delta(x^*,x') = \delta(x^*,x'')$ *for all* $x'' \in \text{Ray}(x^*,x')$ *such that* $x'' \neq x^*$.

A closely associated result is the following:

Lemma 3. *For any* $x^* \in X(0,1)$ *and* $x' \in V(0,1)$:

$\delta(x^*,x'') = x'$ *for all* $x'' \in \text{Ray}(x^*,x')$ *such that* $x'' \neq x^*$.

This latter result says that all focal points on a ray from x^* within the unit hypercube through a vertex x' of this hypercube will directionally round to x'.

[1] Proofs of results stated in this and the following section are contained in Glover (1993).

In accordance with our previous remarks, we are interested in generating points as linear combinations of the points of X(R), to create a basis for generating rounded candidate solutions as part of a scatter search strategy. We will show that the use of directional rounding, for x^* and X(R) properly chosen, makes it possible to restrict these linear combinations advantageously to convex combinations. More precisely, we can limit attention to the region Face(X(R)) to find a point that can be directionally rounded to yield an optimal solution. Furthermore, there is a convex subregion of Face(X(R)) such that every point in this subregion directionally rounds to give this optimal solution. This is expressed in the following result.

Theorem 1. *Let* x^{opt} *denote an optimal IP solution, distinct from* x^*, *and assume that* A1 *and* A2 *hold. Then there is a convex region* $X \subseteq$ *Face*(X(R)) *such that* $\delta(x^*,x') = x^{opt}$ *for all* $x' \in X$.

As already noted, Lemma 1 and linear programming theory imply that A1 and A2 are satisfied by taking $x^* = x(0)$ and defining X(R) in relation to the current basis representation of x(0). Thus we may state the following associated result which is chiefly a consequence of Theorem 1.

Theorem 2. *For* $x^* = x(0)$ *and* X(R) *as identified in Lemma* 1, *defining* x(h), $h \in NB$, *from* (3) *relative to positive values* θ^*_h, *there is a convex region* $X \subseteq$ *Face*(X(R)) *such that all optimal IP solutions (except* x^*, *if it is optimal) belong to the set of directionally rounded solutions* $\delta(x^*,X)$. *Moreover, if* X *is not polyhedral, there is a polyhedral subset of* X *for which this conclusion is true.*

The preceding theorem is also valid if we replace the word "optimal" by "feasible". These results give both a justification and heuristic motivation for the following form of scatter search.

Scatter search with directional rounding

Step 1. Start with any LP feasible extreme point $x^* = x(0)$ for problem (IP).

Step 2. Identify positive values θ^*_h, $h \in NB$, and identify the points $x(h) = x(0) - \theta^*_h D_h$ relative to the current LP basis representation. Let R = NB, thereby creating the set of reference points $X(R) = \{x(h): h \in NB\}$.

Step 3. Let X be a chosen subset of Face (X(R)), generated by taking selected convex combinations of the points of X(R). Then apply directional rounding by reference to the base point $x^* = x(0)$, creating points of the set $\delta(x^*,X)$ as candidates for seeking an optimal solution to (IP).

The preceding method rests on two critical elements: (i) determining the θ^*_h values and hence the precise set of points to compose the set X(R) in Step 2; and (ii) determining which convex combinations of the points of X(R) to generate in Step 3 (as a foundation for the directional rounding). We examine these issues in the following sections.

6. Determining X(R) and a connection with cutting planes

Although Theorem 2 justified choosing any positive values of θ^*_h in order to generate X(R) by (3), some choices are better than others. From a heuristic orientation, three different options for producing the reference points of X(R) are given in Appendix 1. There also is another way to view the issue of determining X(R), which leads to a connection with cutting planes and additional implications relative to assumptions A1–A4. To develop this connection, consider a current basic LP solution x(0), with its associated set of nonbasic variables identified by NB, and let

$$NB0 = \{j \in NB: x_j(0) = 0\}$$
$$NBU = \{j \in NB: x_j(0) = U_j\}.$$

Any valid cutting plane for problem IP, that exludes x(0) and otherwise is satisfied by all optimal IP solutions, can be expressed in the form

$$\Sigma(d_j y_j: j \in NB) \geq d_o$$

where $d_o > 0$, and where the variables y_j are defined so that $y_j = x_j$ for $j \in NB0$ and $y_j = U_j - x_j$ for $j \in NBU$. (For example, the prototypical cuts of Gomory (1960, 1963) take their coefficients directly from the current basis representation, and can be immediately written in this form.) The preceding inequality remains valid by dividing through by d_o, and by replacing any nonpositive coefficient by a positive coefficient. We express the resulting inequality in the form

$$\Sigma(y_j/\theta^*_j: j \in NB) \geq 1 \qquad (4)$$

where $\theta^*_j = d_o/d_j$ if $d_j > 0$, and θ^*_j is an arbitrary positive value otherwise. (These "arbitrary" values preferably should be chosen relatively large in the present context, e.g., a positive multiple of the largest of the remaining θ^*_j values.) Then we may state the following result.

Theorem 3. *Let* $x^* = x(0)$ *be an LP feasible extreme point, and let* θ^*_h *be a positive value for each* $h \in NB$ *determined by reference to the cutting plane inequality* (4) *(i.e., such that all optimal IP solutions, except possibly* x(0) *(if it is IP optimal), satisfy* (4)). *Finally, let the points* x(h) *be defined by* (3), *i.e.,*

$$x(h) = x(0) - \theta^*_h D_h, \ h \in NB.$$

Then, upon determining X(R) *by specifying* R = NB *as in Lemma* 1, *Cone*(x^*, X(R)) *satisfies* A1, *and the inequality* (4) *defines a half space that satisfies* A2 *and* A3, *when expressed in the form of Half_Space* (X(R)).

Of course it is not necessary to translate (4) into the form of Half_Space (X(R)) to apply this result, since the desired θ^*_h values are given by (4) directly. Special motivation for using cutting planes to determine X(R) in this manner is given by a further result that incorporates assumption A4 and more particularly refers to the situation where x(0) is an optimal extreme point of the feasible LP region (which implies all of the assumptions A1–A4 when X(R) is chosen as in Theorem 3).

Theorem 4. *Let* $x^* = x(0)$ *be an optimal, dual feasible LP extreme point, and let* X(R) *be determined relative to a cutting plane inequality* (4) *as in Theorem* 3. *Then, if problem (IP) has an optimal solution, and* x^* *does not qualify to be this solution, there exists a convex region* $X \subseteq$ *Face*(X(R)) *such that: (a) all optimal IP solutions belong to the set of directionally rounded solutions* $\delta(x^*, X)$.
(b) at least one optimal IP solution is given by $x^{opt} = \delta(x^*, x'')$, *where X is restricted to be polyhedral and* x'' *is an extreme point of* X. *Moreover, the foregoing conclusions remain true when "optimal" is replaced by "feasible" in (a) but not in (b), thus permitting (b) also to hold for a larger convex region* X.

The significance of the theorem is that when x(0) is an optimal LP extreme point and X(R) is generated from a cutting plane, the search for an optimal solution by directional rounding can be shifted to consideration of points that potentially qualify as extreme points of X. However, there can also be points of X other than extreme points that likewise directionally round to optimal solutions.

As an implication of our earlier results, every vertex of the unit hypercube obtained by directional rounding over elements of Face(X(R)), whether feasible or not, can be generated over all elements in a convex subregion of Face(X(R)), and hence we seek a method that avoids unnecessary examination of different parts of the same subregion. We also want to focus the generation of points so that the selected subregion X of Face(X(R)) is heuristically determined. The next introduces a way to achieve these goals.

7. The creation of star-paths

To take advantage of the fact that optimal IP solutions are contained among the directionally rounded solutions derived from focal points on Face(X(R)), we consider the construction of paths within Face(X(R)) and directionally round from x^* to points on these paths. Most precisely, in overview, we first construct specially designed paths between selected points of X(R) and other matched boundary points of Face(X(R)). Then these paths are mapped by directional rounding to create associated paths in V(0,1) called star-paths. The elements of the star-paths provide candidate solutions to check for IP feasibility, and to be used for the phase of scatter search that seeds other heuristic processes.

We differentiate between a given path $P \subseteq$ Face(X(R)) and the set of points obtained by directionally rounding from x^* to P, that is, the set $\delta(x^*, P)$. While P represents a continuous trajectory linking two elements of Face(X(R)), the trajectory represented by $\delta(x^*, P)$ is not continuous but broken, and consists of points that are displaced from the surface defined by Face(X(R)) to produce a projection of P onto the zero-one vertices of V(0,1). We call the collection of points $\delta(x^*, P)$, which depends on x^* as well as P, a star-path.

We focus attention on the case where the path P consists of a line segment joining two points x' and x'', given by

$$P(x', x'') = \{x: x = \lambda x'' + (1-\lambda) x', \ 1 \geq \lambda \geq 0\} \qquad (5)$$

Then the associated star-path $\delta(x^*, P)$ may be identified as the set of points

$$\delta(x^*, P) = \{\delta(x^*, x): x \in P(x', x'')\}.$$

We are particularly interested in the situation where x' and x'' are boundary points of Face (X(R)). Our motivation for this stems from the following result, which is implied by the theorems of the preceding section.

Corollary. *Assume* x^{opt} *is an optimal IP solution distinct from* x^*, *and* x' *is any boundary point of Face*(X(R)). *Then there is another boundary point* x'' *of Face*(X(R)), *such that* x^{opt} *belongs to the star-path derived from* x' *and* x''; *i.e.,* $x^{opt} \in \delta(x^*, P)$, *where* $P = P(x', x'')$ *is the line segment given by* (5).

We next consider the choice of points to generate star-paths in accordance with the observation of the preceding Corollary.

7.1. Choosing boundary points for creating star-paths

The points of X(R) are natural candidates to be included among the boundary points for generating star-paths, since X(R) provides the foundation for generating Face(X(R)). Further, we may initially pair each point $x(r) \in X(R)$ with the point y(r) that is the center of gravity of the remaining points of X(R), that is, where y(r) is the midpoint of the lower dimensional face spanned by the points of X(R-r). By such pairing, the path P(x(r),y(r)) traverses the interior of Face(X(R)) to reach a boundary point "equidistant" from the points of X(R-r). Thus the star-path $\delta(x^*, P)$, for $P = P(x(r), y(r))$, is biased (in a loose sense) toward containing solutions anticipated to satisfy the IP feasibility conditions.

We may improve this bias, and simultaneously incorporate objective function considerations as well as feasibility considerations, as follows. Let z(r) denote a modified objective function value for the point x(r) that includes a penalty for infeasibility; that is, z(r) is the value of z for the IP problem when $x = x(r)$, increased by an amount that measures the relative infeasibility of x(r). To create the point y(r) as a weighted combination of the points of X(R-r), we account for the z(r) values by introducing a positive valued function f that preserves their relative ordering. We stipulate that

$f(z(k)) > 0$ for all $r \in R$, and
$f(z(k)) \geq f(z(h))$ if $z(h) \geq z(k)$ for all $h, k \in R$.

Weights $w_h(r)$, $h \in R\text{-}r$, to generate the y(r) points require a normalizing constant $C(r) = \Sigma(f(z(h)): h \in R\text{-}r)$. Then we define

$$w_h(r) = f(z(h))/C(r) \quad h \in R\text{-}r.$$

As a result, this gives

$$y(r) = \Sigma(w_h(r)x(h): h \in R\text{-}r) \quad r \in R.$$

The properties stipulated for the function f include the case where $f(z(r)) = 1$ for all $r \in R$ (which generates each y(r) as the center of gravity of the points of X(R-r)).

Computationally, it is burdensome to have to generate the set of $|R|-1$ weights $w_h(r)$ for each $r \in R$, which involves on the order of $|R|^2$ weights overall. It is possible to do much better as a result of the following observation.

Remark. Define $F=\Sigma(f(z(r)): r \in R)$ and create associated weights $w_r=f(z(r))/F$, $r \in R$. Then identify the weighted center of gravity y for X(R) given by

$y=\Sigma(w_r x(r): r \in R)$.

The ray from x(r) through y, given by

$\text{Ray}(x(r),y)=\{x: x=\lambda y+(1-\lambda)x(r), \lambda \geq 0\}$

then contains the points of the path P(x(r)y(r)). Moreover, the truncated ray that results by restricting λ to satisfy $\lambda \leq 1/(1-w_r)$ is identical to this path, with $y(r)=(y-w_r x(r))/(1-w_r)$.

The preceding Remark shows that determining the weighted center y and generating points on the truncated ray from x(r) through y makes it possible to avoid identifying each of the different sets of weights $w_h(r)$, resulting in computational effort of $0(|R|)$ instead of $0(|R|^2)$. Alternately, y(r) can be computed as indicated in the Remark and the path P(x(r),y(r)) can be identified directly.

7.2. *An efficient procedure for generating the star-paths*

Given an appropriate means for determining the paths $P=P(x(r),y(r))$ for generating associated star-paths $\delta(x^*,P)$, as indicated above, it remains to give an approach for identifying solutions generated by these star-paths.

There are an infinite number of λ values between 0 and 1 to generate P by the definition

$P(x(r),y(r))=\{x: x=\lambda y(r)+(1-\lambda)x(r), 0 \leq \lambda \leq 1\}$.

Each x on P gives a directionally rounded solution $\delta(x^*,x)$ of the collection $\delta(x^*,P)$. However, $\delta(x^*,P)$ contains only a limited number of distinct points, consisting of elements of V(0,1), and hence (infinitely) many of the λ values map into the same point of $\delta(x^*,P)$.

We will show that it is possible to generate the star-path elements highly efficiently. At the same time these elements can be checked for feasibility by simple updating calculations that further decrease computational effort. As a basis for this we demonstrate that the star-path can be represented as a mapping of P onto a collection of distinct, successively adjacent, vertices of the zero-one hybercube.

Let $\delta(\lambda)$ identify the points $\delta(x^*,P)$ as a function of the parameter λ. That is, for a given value of λ, which yields the points x of $P(x',x'')$ given by $x=\lambda x''+(1-\lambda)x'$, we define $\delta(\lambda)=\delta(x^*,x)$. Let $\delta_j(\lambda)$ denote the j^{th} component of $\delta(\lambda)$; i.e., $\delta(\lambda)=\delta(x^*_j, x_j)$, for $x_j=\lambda x''_j+(1-\lambda)x'_j$. We observe that the vector x parameterized by λ can be equivalently written as $x=x'+\lambda\Delta$, defining $\Delta=x''-x'$. Define the subsets I(0), I(+) and I(–) of I to consist respectively of those $j \in I$ such that $\Delta_j=0$, $\Delta_j>0$ and $\Delta_j<0$. Finally, for $j \in I(+)$ or $j \in I(-)$, identify the special λ value given by

$\lambda(j)=(x^*_j-x'_j)/\Delta_j$.

These definitions allow a precise characterization of $\delta(\lambda)$ as follows.

Lemma 4. *The elements of* $\delta(\lambda)$ *are given by*

(a) $\delta_j(\lambda)=\delta(x^*_j,x'), \quad j \in I(0)$

(b) $\delta_j(\lambda)=\begin{cases} 0 \text{ if } \lambda<\lambda(j), & j \in I(+) \\ 1 \text{ if } \lambda \geq \lambda(j), & j \in I(+) \end{cases}$

(c) $\delta_j(\lambda)=\begin{cases} 1 \text{ if } \lambda<\lambda(j), & j \in I(-) \\ 0 \text{ if } \lambda \geq \lambda(j), & j \in I(-) \end{cases}$

Lemma 4 does not depend on the assumption that λ is restricted to satisfy $0 \leq \lambda \leq 1$, but applies to the case where $P(x',x'')$ is the infinite line joining x′ and x″, and not just the segment between these points. In addition, this lemma introduces a specific "tie breaking" rule to handle the case $\lambda=\lambda(j)$, where the original definition of directional rounding requires such a rule to choose between a value of 0 or 1.

To take advantage of Lemma 4, let $\theta(1),\ldots,\theta(u)$ be a permutuation of the indexes of $I-I(0)$ so that $\lambda(\theta(1)) \leq \lambda(\theta(2)) \leq \ldots \leq \lambda(\theta(u))$, where $u=|I-I(0)|$. Also let $\lambda(0)$ be any value of λ such that $\lambda(0)<\lambda(\theta(1))$. (It is acceptable to take $\lambda(0)=-\infty$.) By convention, we will suppose that the $\lambda(\theta(h))$ values are all distinct so that $\lambda(\theta(h))<\lambda(\theta(h+1)$ for all $h<u$. This convention allows a maximum number of elements of the star-path $\delta(x^*,P)$ to be created, and also leads to characterizing these elements as adjacent vertices of the unit hypercube defined relative to the integer components of x. We will show that this convention is trivially easy to impose; that is, no explicit perturbation needs to be introduced to allow the $\lambda(\theta(h))$ values to be treated as distinct in case there are tied values.

Theorem 5. *The star-path* $\delta(x^*,P)$*, where* P *is the line segment joining* x′ *and* x″*, contains precisely* u+1 *distinct points, which can be generated by the rule of Lemma* 4 *when* λ *takes the values* $\lambda(0), \lambda(\theta(1)), \ldots, \lambda(\theta(u))$*. The indicated points* $\delta(\lambda)$ *constitute successively adjacent vertices of* V(0,I)*, linked to each other by the following relationship.*

For any arbitrary value of $\lambda<\lambda(\theta(u))$*, let* $\lambda\_next=\lambda(p)$*, where* $p=\theta(h)$ *for* $h=\text{Min}(k:\lambda(\theta(k))>\lambda)$*. Then* $\delta(\lambda)$ *and* $\delta(\lambda\_next)$ *are associated by the rule*

$\delta_j(\lambda\_next)=\delta_j(\lambda)$ *for* $j \neq p, j \in I$
$\delta_p(\lambda\_next)=1-\delta_p(\lambda)$

The preceding result does not require an explicit numerical shift of tied values of $\lambda(\theta(h))$ in order to allow the specified points of the star-path to be generated. The following simple method based on Theorem 5 produces exactly the desired points $\delta(\lambda)$, as λ ranges between in any interval $\lambda\_start \leq \lambda \leq \lambda\_end$. Consequently, this applies to the special case where $P(x',x'')$ is a line segment generated by $0 \leq \lambda \leq 1$, and also applies to the representation of the path given in the Remark of Sect. 7.1, where λ can exceed the value 1.

Star-path generation method

Step 0. Let $\lambda = \lambda\_start - \varepsilon$, for a small positive value of ε, and generate the solution vector $x^o = \delta(\lambda)$ by Lemma 4.

If $\lambda \geq \lambda(\theta u))$, x^o is the only vector to be generated and the procedure stops. Otherwise, identify $p = \theta(h)$, where $h = \text{Min}$ (k: $\lambda(\theta(k)) > \lambda$).

Step 1. Generate the next x^o vector by setting $x^o_p = 1 - x^o_p$, without changing any other elements x^o_j, $j \in I$, $j \neq p$.

Step 2. Set $h = h + 1$. If $h > u$ or if $\lambda(\theta(h)) > \lambda\_end$, stop. Otherwise, set $= \theta(h)$ and return to Step 1.

Setting $\lambda = \lambda\_start - \varepsilon$ in Step 0 generates the maximum number of points for the star-path, in case λ_start coincides with one of the $\lambda(\theta(h))$ values. (Otherwise, the reference to ε is unnecessary.) Note that $\delta(\lambda)$ is only computed once, in Step 0. Thereafter, each new vector x^o results by simply changing the value of the single element x^o_p in Step 1. This corresponds to applying the formula of Theorem 5 for $x^o = \delta(\lambda\_next)$.

Also, there is no requirement in Step 2 that the $\lambda(\theta(h))$ values be strictly increasing. The procedure simply increments h, and all tie breaking is entirely implicit. Finally, the vector $x^o = \delta(\lambda)$ can be checked very efficiently to determine if it is a feasible zero-one solution, due to the fact that exactly one element of x^o changes at each execution of Step 1. Thus, the feasibility check can be based on a marginal calculation, rather than evaluating a complete new vector at each step.

Our earlier results have demonstrated that selecting λ values within the range from 0 to 1 suffice if $x' = x(r)$ and $x'' = y(r)$, for x(r) and y(r) appropriately matched. However, we note it may be useful to allow consideration of points generated over a wider range of λ values, using the preceding method. Also, all of the x(r), y(r) pairs will generate the point $\delta(x^*, y)$, where y is the weighted center of gravity identified in Sect. 7.1. This duplication can easily be avoided, if desired, by accounting for the λ value to be skipped.

7.3. Adaptive re-determination of star-paths

There exists a useful diversity of possibilities for generating star-paths. These derive from the different alternatives for generating the set of points X(R) by the cutting plane connection of Sect. 5.2 (and the approaches to Appendix 1), and also derive from the different choices of a function f for generating weights to produce the x(r), y(r) pairs. We show that the outcomes of implementing such approaches can be embedded in a higher level process that determines new star-paths adaptively. In this way the scatter search method effectively learns how to modify itself to take advantage of information generated from previous efforts.

The basic notion is the following. Each star-path trajectory is based on focal points in Face(X(R)) that are directionally rounded to produce the star-path points. As shown in the preceding section, these focal points do not have to be explicitly identified (except for the first). Nevertheless, the λ value that produces each star-path point is always known, and can be used to identify the associated focal point of Face(X(R)). (We have taken liberties by allowing some focal points to lie outside of Face(X(R)) in Plane(X(R)).) Moreover, in general, a range of λ values is known that produces each star-path point. A midpoint of this range can be selected to give a representative focal point.

As a result, we may consider a collection E of elite elements of V(0,1) that are generated from various star-paths. This gives rise to an associated collection F(E) of focal points that create the elements of E (where each $x \in F(E)$ yields a point $\delta(x^*, x)$ of E). The criteria for identifying E can be based on objective function values penalized for infeasibility or can more broadly include reference to diversification (see, e.g., Glover (1989, 1991), Woodruff and Spearman (1992), and Reeves (1993b)).

The elements of F(E) identify a set of preferred focal points, and their convex combinations may be viewed as defining a preferred focal region. Moreover, the points of F(E) can be treated exactly as the points X(R) in Sect. 7.1 to generate a weighted center of gravity y. Then the point y can be used to generate paths P(x(r),y), which are extended beyond y to create a match between x(r) and an implicit point y(r).

By Theorem 5 and the star-path method accompanying it, there is no need to precisely identify a λ value that will generate a point y(r), since it suffices to take $x' = x(r)$ and $x'' = y$, and to generate a sequence $\lambda(\theta(h))$ that will encompass all relevant possibilities. The extreme ends of the sequence are likely to be irrelevant, and there is no need to duplicate the use of x(r) as a focal point. Hence the star-path generation may be applied by starting at a $\lambda(\theta(h))$ value larger than 0 (since $\lambda = 0$ treats $x' = x(r)$ as a focal point), and to continue until persistent deterioration in the star-path elements occurs. The points x(r) themselves can be replaced on subsequent rounds by shifting them toward the most recent y, in which case negative starting λ values may be relevant.

As the adaptive procedure is repeated, tabu search can be used to avoid duplicating the composition of the collection E, and hence of the focal set F(E) (e.g., following the design for controlling scatter search in Glover (1991)). Moreover, it is possible to apply the approach in a compound manner. We identify a way to do this in Appendix 2, effectively permitting a given collection E to generate alternative collections by a nested form of adaptation.

8. Final considerations

We have seen that solution combinations with special properties for zero-integer programming problems can be produced by focusing on two aspects of the scatter search framework, the characterization of reference points that generate solution combinations and the mechanism for transforming fractional elements into integer elements. A number of options exist for taking advantage of the outcomes of this focus. Parallel processing can take a useful role in generating and coordinating the set of points X(R), and in choosing a function f to generate weights for producing additional matched points y(r). Parallel processing also can take a role in simultaneously determining parameters associated with different elements of X(R), and in determining star-paths associated with the x(r), y(r) pairs.

These processes can be usefully applied by selecting base points from different extreme points of the LP feasible region. Candidates also can be selected from the trial solutions generated from star-paths. Whether feasible or not, such trial solutions can be used to create a modified objective function, weighting costs to produce optimal LP extreme points that lie in the vicinity of these solutions. Likewise, the trial solutions may be subjected to heuristic modification, following the standard scatter search design. An issue that merits empirical examination is whether some types of problem domains are exploitable by creating trial solutions solely from directional rounding and star-paths, without accompanying heuristic modification. The associated intensification strategy of Appendix 2 may be relevant in such cases.

Next steps

Additional possibilities exist for creating combined solutions by extension of the scatter search approach. Instead of relying on spatial structures (based on mappings of extreme points and focal points) to determine paths that link solutions, and hence that generate new solutions, it is possible to use neighborhood structures as a basis for such a linkage. Thus, solutions can be created by designing neighborhood trajectories to join reference points. This type of path relinking approach (Glover (1989, 1994)), can be applied in the present context by allowing the star-path conception to be transported from the spatial setting to the neighborhood setting. The knowledge that there exists a spatial definition of δ with useful properties motivates a quest for a neighborhood definition that also has such properties. In general, the ideas of path relinking and path projections offer many areas for investigation, including the potential to develop procedures that joint the spatial and the neighborhood structures for creating linkages between solutions.

References

Aarts EHL, Eiben AE, van Hee KM (1989) A General Theory of Genetic Algorithms. Computing Science Notes, 89/8, Eindhoven University of Technology

Aboudi R, Jörnsten K (1992) Tabu Search for General Zero-One Integer Programs Using the Pivot and Complement Heuristic. ORSA J Comput

Ackley D (1987) A Connectionist Model for Genetic Hillclimbing. Kluwer, London

Bäck T, Hoffmeister F, Schwefel H (1991) A Survey of Evolution Strategies. Proceedings of the Fourth International Conference on Genetic Algorithms, Morgan Kaufmann, San Mateo, California, 2–9

Balas E, Martin C (1980) Pivot and Complement – a Heuristic for 0–1 Programming. Manag Sci 26(1):86–96

Battiti R, Tecchiolli G (1992) Parallel Biased Search For Combinatorial Optimization: Genetic Algorithms and Tabu Search. Microprocessors and Microsystems 16:351–367

Cabot AV, Hurter AP (1968) An Approach to Zero-One Integer Programming. Oper Res 16:1212–1216

Costa D (1992) An Evolutionary Tabu Search Algorithm and the NHL Scheduling Program. ORWP 92-11, Ecole Polytechnique Fédérale de Lausanne, Switzerland

Dantzig G (1963) Linear Programming and Extensions. Princeton University Press, Princeton

Davis L (ed) (1991) Handbook of Genetic Algorithms, Van Nostrand Reinhold, New York

Dorndorf U, Pesch E (1995) Evolution Based Learning in a Job Shop Scheduling Environment. Comput Oper Res 22:25–40

Eschelman LJ, Schaffer JD (1992) Real-Coded Genetic Algorithms and Interval-Schemata. Technical Report, Phillips Laboratories

Glover F (1964) A Bound Escalation Method for the Solution of Integer Linear Programs. Cahiers de Recherche Opérationelle 6(3):131–168

Glover F (1968) A Note on Linear Programming and Integer Infeasibility. Oper Res 16:1212–1216

Glover F (1977) Heuristic for Integer Programming Using Surrogate Constraints. Dec Sci 8:156–166

Glover F (1989) Tabu Search – Part I. ORSA J Comput 1(3):190–206

Glover F (1991) Tabu Search for Nonlinear and Parametric Optimization (with Links to Genetic Algorithms) (to appear) Discr Appl Math

Glover F (1993) Directional Rounding and Vertex-Cone Projects for Zero-One Programming. University of Colorado, Boulder

Glover F (1994) Tabu Search: Improved Solution Alternatives. Mathematical Programming: State of the Art 1994, J.R. Birge, K.G. Murty (eds) The University of Michigan Press

Glover F, Lokketangen A (1994) Solving Zero-One Mixed Integer Programming Problems Using Tabu Search, Research Report, University of Colorado, Boulder

Glover F, Kelly J, Laguna M (1992) Genetic Algorithms and Tabu Search: Hybrids for Optimization. Comput Oper Res (to appear)

Goldberg DE (1989) Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading, Mass.

Gomory R (1960) An Algorithm for the Mixed Integer Problem, RM-2597, The Rand Corporation

Gomory R (1963) An Algorithm for Integer Solutions to Linear Programs, in Recent Advances in Mathematical Programming, Graves and Wolfe (eds) McGraw-Hill, New York

Hu TC (1969) Integer Programming and Network Flows. Addison-Wesley, Reading, Mass.

Lokketangen A, Jörnsten K, Storoy S (1993) Tabu Search Within a Pivot and Complement Framework, paper presented at the IFORS 93 Conference, Lisbon, Portugal

Michalewicz Z (1993) Evolutionary Computation Techniques for Nonlinear Programming Problems, paper presented at the IFORS 93 Conference, Lisbon, Portugal

Michalewicz Z, Vignaux GA, Hobbs M (1991) A Non-Standard Genetic Algorithm for the Nonlinear Transportation Problem. ORSA J Comput 3(4)307–316

Moscato P (1993) An Introduction to Population Approaches for Optimization and Hierarchical Objective Functions: A Discussion on the Role of Tabu Search. Ann Oper Res 41:85–122

Mühlenbein H, Gorges-Schleuter M, Krämer O (1988) Evolution Algorithms in Combinatorial Optimization. Parallel Comput 7:65–88

Mühlenbein H (1992) Parallel Genetic Algorithms in Combinatorial Optimization, to appear in Computer Science and Operations Research, Osman Balci, ed., Pergamon Press, New York

Murty K (1976) Linear and Combinatorial Programming. John Wiley, New York

Nemhauser G, Wolsey L (1983) Linear and Combinatorial Optimization. John Wiley, New York

Parker G, Rardin R (1988) Discrete Optimization. Academic Press, New York

Reeves C (1993 a) Modern Heuristic Techniques for Combinatorial Problems. Blackwell, Oxford

Reeves C (1993b) Diversity and Diversity in Genetic Algorithms: Some Connections with Tabu Search. Coventry University, United Kingdom
Salkin H (1975) Integer Programming. Addison-Wesley, Reading, Mass.
Ulder NLJ, Pesch E, van Laarhoven PJM, Bandelt HJ, Aarts EHL (1991) Genetic Local Search Algorithm for the Traveling Salesman Problem. Parallel Problem Solving from Nature, R. Männer, H.P. Schwefel (eds) (Lect. in Computer Science, vol 496) Springer, Berlin Heidelberg New York, pp 109–116
de Werra D, Hertz A (1989) Tabu Search Technique: A Tutorial and an Application to Neural Networks. OR Spektrum 11:131–141
Whitley D (1993) Foundations of Genetic Algorithms 2, Morgan Kaufmann
Woodruff DL, Spearman ML (1992) Sequencing and Batching for Two Classes of Jobs with Deadlines and Setup Times. Product Oper Manag 1(1):87–102
Woodruff DL, Zemel E (1993) Hashing Vectors for Tabu Search. Ann Oper Res 41:123–137
Zionts S (1974) Linear and Integer Programming. Prentice Hall, Englewood Cliffs

Appendix 1

Heuristic options for generating reference points

For the following, we take R = NB, or in the case of a large problem, take R to be a preferred subset of NB. The first option below is a base point, against which the effectiveness of other options can be measured.

Option 1. Let θ_h be defined as in (2) to be the value that generates the extreme point x(h) adjacent to x(0). Let $\theta^*_h = \theta_h$ if $\theta_h \neq 0$, $h \in R$. Otherwise, define

$$\theta = 0.5\ \mathrm{Min}(\theta_h : \theta_h \neq 0,\ h \in R)$$

and set

$$\theta^*_h = \theta \text{ if } \theta_h = 0,\ h \in R.$$

Option 2. Let z′ be the optimal LP value for z and let z* be the best known IP feasible value of z (or a target for this value). Also let d_h denote the current LP reduced cost of x_h at the extreme point x(0). In accordance with the sign convention for the entries of D_h (which gives the nonbasic variable x_h an entry of $D_{hh} = 1$ or -1 according to whether x_h is currently as its lower or upper bound), we may write

$$z(h) = z(0) + d_h\theta_h,\ h \in NB$$

where $d_h \geq 0$ for all $h \in NB$ at an LP optimum. We select two values z(+) and z(–) for z, by the approach subsequently described, related to the target value z*. Then the values θ^*_h are determined so that z(h) = z(+) for all $h \in R$ with $d_h > 0$, and z(h) = z(–) for all $h \in R$ with $d_h < 0$.

In particular, if $z^* > z(0)$ we set $z(+) = z^* + 0.3(z^* - z')$ and $z(-) = z(0) - 0.3(z^* - z')$, while if $z^* < z(0)$ we set $z(-) = z^* - 0.3(z^* - z')$ and $z(+) = z(0) + 0.3(z^* - z')$. (The constant 0.3 can of course be altered.) This outcome results by the following θ^*_h values. Define $\alpha = z^* - z(0)$, and $\beta = z^* - z'$.

Case A: $\alpha \geq 0$
Set $\theta^*_j = (\alpha + 0.3\beta)/d_h$ for $d_h > 0$, $h \in R$
and $\theta^*_h = -0.3\beta/d_h$ for $d_h < 0$, $h \in R$.

Case B: $\alpha < 0$
Set $\theta^*_h = (\alpha - 0.3\beta)/d_h$ for $d_h < 0$, $h \in R$
and $\theta^*_h = 0.3\beta/d_h$ for $d_h > 0$, $h \in R$.

Option 3. This is a feasibility modification of Option 2. Normalize each inequality constraint of the original LP formulation, before adding a slack variable, by dividing through by the sum of absolute values of the constraint coefficients. Then assign an infeasibility measure to each point x(h) created by Option 2:

$$v(h) = \sum \mathrm{Max}(0, -x_j(h), x_j(h) - U_j : j \in N).$$

Replace each value θ^*_h of Option 2 by dividing it by the quantity $1 + v(h)$, thus defining new θ^*_h values for determining associated new points x(h).

Evident variations on the preceding options are possible. We have stated them in a highly concrete form to give a starting point for empirical study.

Appendix 2

Strategy for augmenting a collection of preferred solutions

We identify a simple instance of scatter search as a basis for compounding the adaptive approach for generating star-paths. The goal is to efficiently produce new zero-one solutions as candidates to be included in a preferred collection E. The approach is based on the fact that the members of E can be treated as reference points, and thereby give rise to additional candidates for membership by a process of creating weighted combinations.

Let S denote an index set for a chosen subset of the solutions in E, thus identifying the members of this subset by $X(S) = \{x(s), s \in S\}$. For simplicity, consider the result of rounding in a nearest neighbor sense. If all points are weighted equally, thereby producing a center of gravity of X(S), the result of nearest neighbor rounding yields a point x whose component x_j, $j \in I$ receives the value taken by the majority of the components $x_j(s)$, $s \in S$. Thus if S contains an odd number of elements, the point x is uniquely determined.

This observation suggests the following strategy for generating trial solutions x as candidates to augment the collection E, or to replace its inferior elements.

Simple scatter search approach to augment E

1. Let S(k) be the index set for the "k best" solutions from the collection E.
2. Choose one or more even values of k (e.g., $k=4$ and $k=6$), and consider each of the k subsets of S(k) consisting of k-1 of its elements, (e.g., each of the 4 subsets of 3 solutions from S(4), and each of the 6 subsets of 5 solutions from S(6)).
3. Represent each of the chosen subsets of solutions by $\{x(s), s \in S\}$, where S contains an odd number of elements. From each subset, generate a trial point x where, for each $j \in I$: $x_j = 1$ if $x_j(s) = 1$ for the majority of $s \in S$, and $x_j = 0$ otherwise. (For example, 10 trial points are generated from S(4) and S(6), and 20 trial points are generated by letting S range over all 3 element subsets of the 6 best solutions.)
4. For each trial point x, test whether it passes a threshold of attractiveness to be admitted to the collection E (e.g. whether it has an evaluation better than the average member of E).

While some of the trial points produced by this approach may duplicate others, such duplications are quickly eliminated if the threshold test in step 4 requires each point selected to be better than its predecessor.

Solutions can alternatively be weighted by their objective function values to obtain trial solutions other than by majority vote. If all of the elements of X(S) are feasible, let z' denote the optimum value of z for the original LP solution. Then $\Delta z(s) = z(s) - z'$ is positive for all $s \in S$, or else an optimal solution is known. (If some elements of X(S) are infeasible, simply select z' to be smaller than the minimum z(s) value.) Create a convex combination of the points of X(S) using the weights $\Delta z(s)/D$, where $D = \Sigma(\Delta z(s){:}\ s \in S)$. Then x can be generated by rounding this outcome.

As in standard scatter search, variants can be applied to problems where simple rounding is inappropriate, using generalized rounding processes that are executed sequentially to allow for changes to meet constraint requirements.