

Genetic algorithms and scatter search: unsuspected potentials

FRED GLOVER

School of Business, CB 419, University of Colorado, Boulder, CO 80309, USA

We provide a tutorial survey of connections between genetic algorithms and scatter search that have useful implications for developing new methods for optimization problems. The links between these approaches are rooted in principles underlying mathematical relaxations, which became inherited and extended by scatter search. Hybrid methods incorporating elements of genetic algorithms and scatter search are beginning to be explored in the literature, and we demonstrate that the opportunity exists to develop more advanced procedures that make fuller use of scatter search strategies and their recent extensions.

Keywords: Genetic algorithms, scatter search, optimization, mathematical relaxation

1. Introduction

Genetic algorithms (GAs) were initially proposed as hyper-plane sampling approaches rather than as optimization methods. Over time, however, GAs have increasingly come to be viewed as a means for solving optimization problems, and today optimization stands as a principal area of focus for these methods. Simultaneously, researchers are coming to realize that links exist between GAs and other approaches for solving optimization problems. This dawning recognition has more than casual significance.

The area of mathematical relaxation, customarily perceived to be entirely different from GAs, has significant overlaps with GAs that have previously gone unnoticed. Principles underlying heuristic methods that have descended from mathematical relaxation approaches give a foundation for creating new forms of GAs that hold promise for increased success in the optimization area.

In this paper we will show how a class of heuristics called scatter search methods, which originated as counterparts to approaches from mathematical relaxation, provide strategies that both reinforce and extend strategic ideas for improving GA methods. The compatibility of these lines of development, which have appeared so different on the surface, underscores the value of the themes that arise from their roots.

In the following sections, we trace the evolution of relaxation methods to disclose similarities between

their concepts and those of GA methods. A number of coincidences emerge. The scatter search approach appeared in the same time period that witnessed the publication of the early GA formulations. Some of the notable differences between the two approaches at the time, occasioned largely by the fact that scatter search was intended for solving optimization problems while GAs sought other objectives, have dissolved in more recent years. As will be seen, a number of the key departures from the original GAs, which notably improved their empirical performance for solving optimization problems, were basic elements of the initial scatter search framework. More importantly, a few researchers in the GA community (and outside) have recently begun to propose ways to exploit further these connections and obtain additional advantage from them.

In the development that follows, we begin by surveying fundamental ideas from mathematical relaxation that have intriguing correspondences with ideas that come alternately from genetic algorithms. Building on this, we show how these ideas were transported from the ‘dual framework’ of relaxation approaches to create the ‘primal framework’ of scatter search. This provided a method whose underlying notions contain additional similarities to GA ideas, as well as other elements that are not yet fully assimilated into GA approaches, and suggest possibilities for creating new hybrid methods. Finally, we discuss recent extensions of these ideas that offer an opportunity to create still more advanced procedures based on integrating scatter search and GA methods.

2. Ties to mathematical relaxation

Three features of genetic algorithms are often observed as essential to their fundamental character, consisting of the following operations:

1. Working with a population of elements (coded as vectors);
2. Creating new elements by combining existing elements;
3. Determining which new elements are retained (i.e. are allowed to 'survive') based on a measure of quality or 'fitness'.

Each of these features is also basic to the approaches that come from the area of mathematical relaxation. The degree to which these approaches share correspondences with GAs at deeper levels (falling short of the GA conception or providing possible enhancements), depends on the type of mathematical relaxation considered. We begin by examining two main types of mathematical relaxation. Lagrangean relaxation and surrogate constraint relaxation, that share correspondences with GAs at different levels.

2.1. Lagrangean relaxation

Lagrangean relaxation has made a large impact on the field of optimization, and represents one of the key strategies for solving problems from this domain: see, for example, Fisher (1973), Greenberg (1973a), Geoffrion (1974), and the surveys by Fisher (1985) and Beasley (1993).

Lagrangean relaxation was first introduced by Everett (1963). The basic idea is to overcome the difficulty of working directly with a system of constraints by absorbing them into an objective function to be minimized. This is done in such a way that violating the constraints will cause the objective function to become larger. An attempt to solve the modified (presumably easier) problem with the constraints removed, therefore, will tend toward the discovery of solutions that satisfy the constraints. Since the modified problem accepts solutions as feasible that do not satisfy the absorbed constraints, and employs an objective function that underestimates (can be no larger than) the value of the true objective function when the problems are solved to optimality, the modified problem is called a 'relaxation' of the original, and in general its creation and solution are the subject of the area broadly known as *mathematical relaxation*.

At first blush the idea underlying this relaxation approach seems to have little in common with the three features previously identified as fundamental to genetic algorithms. However, the mechanism in Lagrangean relaxation for taking the constraints into the objective function is to unite them, in particular by forming a non-negative linear combination (which multiplies each

constraint by a non-negative weight and sums the result). Further, in the case where the constraints consist of linear inequalities, each constraint is summarized by a vector (identifying the coefficients of problem variables and an associated right hand side). Consequently, Lagrangean relaxation is customarily applied by seeking an appropriate combination of vectors—and more precisely, a best combination, identified as one that makes the relaxed objective function value the largest at optimality. Since this optimal value underestimates the true optimal value, a best combination therefore is one that produces the smallest underestimation. (If the underestimation can be reduced to zero, which is the ideal outcome, the relaxed problem is said to yield a *zero duality gap*. More often, notably in combinatorial problems, the underestimation cannot be reduced to zero, and the relaxed problem is said to yield a *positive duality gap*.)

No way is known for creating a best combination of vectors in a single step. Consequently, Lagrangean relaxation proceeds iteratively in successive stages, changing the weights of the vectors in the objective function based on the degree to which the associated constraints are violated or satisfied. If the new combination of vectors proves worse than before, it may be discarded or combined with the previous one, or simply utilized as it is, as a basis for the next stage. The best combination overall is retained to become the final choice.

Although there are apparent similarities to some of the basic GA notions, there are clearly significant points of contrast. Apart from evident differences between modes of combination (non-negative linear combinations versus genetic crossover), a major difference lies in the fact that the populations for Lagrangean relaxation are determined once and for all from the original vectors to be combined, and do not dynamically alter. We have also stated the approach in terms of generating and evaluating a single combination in each generation. This is obviously an unnecessary limitation, however. In some implementations, strategies for re-weighting the vectors are applied in parallel, and the results are culled to produce a best generational outcome. Nevertheless, at this point, the correspondences appear to be mostly in the nature of an intellectual curiosity, and their implications are not entirely clear. In the next class of mathematical relaxations, however, we will see the beginnings of more intriguing forms of correspondence.

2.2. Surrogate constraint relaxations

Surrogate constraint relaxations were introduced as a strategy for discrete optimization problems in Glover (1965), sharing with Lagrangean relaxations the notion of combining vectors by forming non-negative linear combinations. However, while Lagrangean relaxations have the goal of eliminating the constraints by absorbing them into the

objective function, surrogate constraint relaxations have the goal of generating new constraints that may stand in for the original constraints (hence the name ‘surrogate’ constraints). Consequently, surrogate constraint relaxations undertake to create new vectors with the same form and function as the original vectors, and which additionally contain desirable features not found in members of the original system.

The surrogate constraint approach also introduces another element that is relevant to our present concerns. Constraints formed by non-negative linear combinations of the original constraints are validly implied by the original constraints, but they are not the only ones that may be so implied. In combinatorial and non-convex optimization, valid constraints can also be composed by transformations to produce inequalities called *cutting planes*. Accordingly, valid constraints derived by such additional transformations likewise provide a basis for creating new surrogate constraints. By this aspect the surrogate constraint approach acquires the ability to operate with a system of vectors that is dynamic and changing. (A surrogate constraint system formed in this way using standard cutting planes is theoretically assured to be sufficient for determining optimal solutions to integer programming problems (more precisely, for determining the convex hull of integer solutions: see, for example, Nemhauser and Wolsey, 1988).)

The differences in perspective introduced by surrogate constraint relaxations, in contrast with Lagrangean relaxations, have a number of strategic consequences. One of the more notable consequences is to view the creation of surrogate constraints as a process for generating information. As expressed in Glover (1968): ‘A surrogate constraint is an inequality ... designed to capture useful information that cannot be extracted from the parent constraints individually, but is nevertheless a consequence of their conjunction.’ (A novel coincidence occurs in the use of the ‘parent’ terminology, which also has become popular in the GA literature.)

The information orientation has important implications for designing procedures to combine vectors. Information is context dependent, and is subject to distortion or to refinement, according to the treatment of considerations such as redundancy, logical consequences and criteria for evaluation. Consequently, the early surrogate constraint approaches incorporated the following additional elements:

- (a) Culling out and eliminating redundant vectors (accounting for information conditional on the solution stage);
- (b) Using surrogate constraints as a basis for inferring useful auxiliary information (such as bounds on variables and sums of variables);
- (c) Introducing different criteria of quality as a basis for generating different vector combinations;

- (d) Providing methods to produce combinations that are best in the probabilistic sense of mathematical expectation;
- (e) Identifying subsets of parents that exhibit particular characteristics, and seeking best vector combinations over each subset.

These developments contain strategic notions clearly resembling ideas presented in the early GA literature, and in some cases incorporate features that were not considered in this literature. Today most of these notions are thoroughly familiar to those who apply genetic algorithms to optimization.

These connections offer the possibility to exploit several productive relationships that emerge from additional developments that followed. Before describing these developments, we take a closer look at the concept of mathematical relaxations, and the way in which this concept is woven into the fabric of mathematical duality theory.

2.3. Mathematical duality

Mathematical duality is intimately related to mathematical relaxation by associations that were first made apparent with the study of Lagrangean relaxation. In fact, Lagrangean relaxation gives rise to a duality theory that is equivalent to an older duality theory for mathematical optimization called Fenchel duality. The importance of this duality is illustrated by its ability to give the well known duality results for linear programming as a special case.

A *Lagrangean dual problem* is the same as the problem of finding a best Lagrangean relaxation—i.e. a best way to weight constraint vectors to yield the smallest underestimation of the original problem objective, when the vectors are combined and absorbed into the Lagrangean objective. This correspondence between duality and relaxation gives rise to the terminology that refers to the degree of underestimation as a ‘duality gap’, as noted earlier. In this way, the goal of optimizing over a *dual* problem also clearly becomes relevant to optimizing over a *primal* (or original) problem.

Surrogate constraint relaxation leads to an associated duality theory with additional features. A *surrogate dual problem* is the same as finding a best surrogate relaxation—a best combination of the original constraints to yield a surrogate problem (embodying the resulting surrogate constraint) whose optimum objective function value underestimates the optimum value for the original problem by the least amount.

By this characterization, results for determining best surrogate constraints also give basic results for surrogate duality theory. Constructive methods underlying these results, as developed in Glover (1965) and recently extended by Fréville and Plateau (1993), make it possible to establish bounded step sizes for finding best surrogate

constraints that have no counterpart in Lagrangean theory, where precise bounds for step sizes are unknown for the goal of finding a best Lagrangean objective.

The constructive surrogate constraint methods also imply that surrogate duals satisfy a quasi-concavity property (by comparison to a simpler concavity property satisfied by Lagrangean duals). These duality implications are set out in fundamental papers by Greenberg and Pierskalla (1970; 1973), and Greenberg (1973*b*). Building on this work, a comprehensive duality theory for surrogate constraint relaxations is given in Glover (1975), including a treatment of composite dual problems that incorporate Lagrangean objectives. Additional fundamental contributions, with special consideration of searching for best multipliers, are provided by Karwan and Rardin (1976, 1979).

An important aspect of surrogate duality is its ability to yield smaller duality gaps than Lagrangean duality—that is, solving the relaxed problem gives a better chance of determining the optimum objective value for the original. Beyond this influence on duality gaps, however, the surrogate dual yields a greater chance of locating trial solutions that are feasible for the original problem. This *primal advantage* of surrogate constraint relaxations has recently been underscored in the findings of Gavish and Pirkul (1985) and Fréville and Plateau (1992; 1993), who obtain highly effective results applying surrogate constraint methods to special classes of integer programming problems.

The primal aspect of surrogate constraint methods holds further implications for our present development, for it led to the step of adapting surrogate constraint ideas to operate entirely on a primal setting. Procedurally, this involved a shift to generating combinations of vectors that represent solutions, in place of those that represent constraints, giving rise to the class of surrogate constraint heuristics called scatter search. This provides our next link to GA approaches and to the possibility of creating useful hybrid methods that unite these two sources.

3. Scatter search

Scatter search starts with a collection of solutions called *reference points*, obtained by the application of preliminary heuristic processes. Instead of utilizing non-negative linear combinations of the initial vectors to create new vectors, as in the approaches of Lagrangean and surrogate relaxations, scatter search employs more general linear combinations that include negative weights.

The approach initially generates weighted centres of gravity which, together with subsets of original reference points, identify subregions as a foundation for generating subsequent points. The allowance for negative weights makes it possible to go outside the area spanned by the reference points, introducing an element of increased diversity in the new solutions that result. The vectors

are additionally subjected to an iterative rounding process, to give integer values to components that are required to be discrete. By such a process, a selected discrete variable is rounded to an integer neighbour, followed by determining implied value changes for other variables, and then repeating.

The purpose of modifying discrete components iteratively is to accommodate interdependencies among the problem variables. This provides an adaptive approach to respond to implications of problem structure and the location of vectors in solution space. We later give an illustration to clarify these points.

New vectors that result from the scatter search combination process are re-submitted to be operated on by heuristic solution procedures, with the goal of transforming them into solutions of still higher quality. This brings the approach full circle, allowing vectors generated along the way to be selected as new reference points, and then the process begins again.

The original scatter search development in Glover (1974; 1977) characterized this approach as ‘conveniently suited to application with learning strategies’. Methods were proposed to support this by introducing memory into the search, compelling the search to cross boundaries of feasibility and local optimality, and exploiting variables that qualified as *strongly determined* and as *consistent*. These memory-based strategies, taken independently, have useful consequences for solving optimization problems, and have evolved into what today is known as tabu search (see, for example, the survey paper of Glover and Laguna (1993), and the volume of Glover *et al.* (1993)). We point out in Section 4 how these strategies can take a valuable role in supplementing those based on generating combinations of solutions.

3.1. An illustration

Visual intuition about the operation of scatter search, before applying its adaptive rounding component, can be gained by reference to Fig. 1 from Glover (1977). Each of the points numbered 1 through 16 in Fig. 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 6, 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

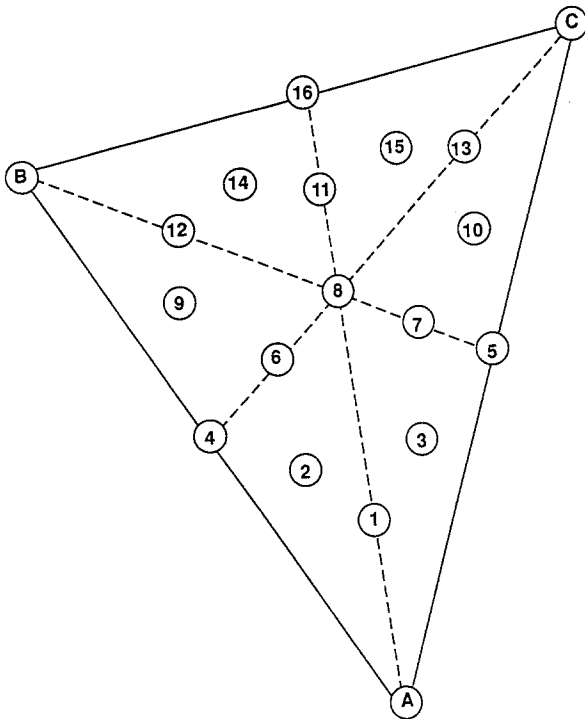


Fig. 1. An illustration of scatter search

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. We examine some of the implications of these characteristics, and their relation to developments in genetic algorithms stemming from the original work of Holland (1975), in the following sections.

3.2. Relevance of embedded heuristics

The scatter search component of subjecting each generation of solutions to heuristic processes, as a means to improve them and account for problem structure, finds its counterpart in the amended version of GAs called *parallel genetic algorithms*, or PGAs, by Mühlenbein *et al.* (1988), and in the related *genetic local search* approach of Ulder *et al.* (1991). Although some GA researchers view these approaches as inappropriate to be joined to the GA framework, the iterative inclusion of heuristic processes has resulted in solutions dramatically superior to those generated by the original GA proposals. Consequently, the current trend in most applications of GA methods to optimization is to make use of embedded heuristics as fully as possible.

3.3. Combining solutions

The manner of combining solutions in scatter search appears substantially different from the use of genetic

crossover in GAs. Within the context of early GAs, there are solid foundations for this impression. Perhaps surprisingly, combinations generated by scatter search include as a special instance those produced by the original genetic crossover operation, and by many of its later generalizations.

Before demonstrating this, we first observe that limited versions of the linear combination approach have been proposed by researchers in the late 1980s and early 1990s as alternatives to the original crossover ideas. As pointed out by Michalewicz (1993), the operation of taking non-negative linear combinations of two parent vectors has been introduced under a variety of names; e.g. *guaranteed average crossover* in Davis (1989) (for the special case where each parent has an equal weight of $1/2$), *intermediate crossover* in Bäck *et al.* (1991), *linear crossover* in Wright (1990) and *arithmetical crossover* in Michalewicz and Janikow (1991) and Michalewicz (1992).

Linear combinations other than those with non-negative weights (although still restricted to two parents) were introduced by Wright (1990), who applied the name *heuristic crossover* to combinations that extend beyond the line segment spanned by the parents. In this case the extension was made in the direction determined by the 'preferred' parent and the degree of the extension was randomly selected.

3.4. Linear combinations with generalized adjacency rounding

As previously noted, the process for combining solutions in scatter search incorporates two primary elements:

- (a) identifying selected subregions of points, and generating new points by multiparent combinations relative to these subregions;
- (b) differentiating between discrete and continuous vector components, and rounding discrete components iteratively to account for interdependencies among variables.

We focus here on the second of these points, examining the consequences of differentiating between vector components, and of rounding discrete components to integer values. To gain an understanding of the role of these elements, we first stress the relevance of the policy in scatter search of operating with vectors whose components are not restricted to binary values. Although binary representations serve as the foundation of the original genetic encoding and crossover notions, and are still considered fundamental to GA theory, they introduce limitations that are often unrecognized.

To demonstrate the nature of these limitations, consider the goal of creating integer solutions that are combinations of the two solutions $x = 9$ and $x = 26$. Rounded linear combinations as proposed in scatter search 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, (10010) for $x = 9$, and (01011) for $x = 26$ (encoding x as a bit string by the identity $x = 1x_1 + 2x_2 + 4x_3 + 8x_4 + 16x_5$, where x_1, x_2, \dots, x_5 are zero-one integer variables), the possible outcomes are substantially more limited. In particular, the only ways to create rounded linear combinations of the vectors (10010) and (01011) yield the collection of binary vectors ($\star \star 01\star$) where the ' \star elements' can be 0 or 1. Hence instead of producing all possible integer points these combinations produce only the integer values of x satisfying $8 \leq x \leq 11$ and $24 \leq x \leq 27$.

3.5. Comparisons to standard crossover

The same example shows how scatter search combinations relate to combinations created by various standard forms of genetic crossover. (We continue to consider only the two-parent case, to provide a basis for comparison.) The possible outcomes of combining the two previous vectors become quite restricted by applying classical genetic crossover. The only vectors that can be created by the proposals of Holland, for example, are the four vectors (00010), (01010), (10011), (11011), corresponding to $x = 8, 10, 25, 27$. When attention is restricted to binary vectors, which we have seen to be inappropriate, rounded linear combinations used in scatter search in fact give the same set of possibilities as the multipoint 'uniform' crossover operator proposed by Ackley (1987). (In Ackley's proposal, each bit position is assigned a probability of 0.5 for swapping the elements of the parents—identifying the offspring as the modified parents. Later proposals, such as Spears and DeJong (1991), include randomized selections with other probabilities. The approach of making changes strategically instead of by recourse to randomization is not considered.)

By further employing iterative rounding, where values of some variables may change as a result of modifying others, additional possibilities result, since by such a process a vector element with a given value in the parent vectors may come to receive a different value in an offspring.

3.6. Heeding interactions among variables

Next we demonstrate the importance of rounding to account for interactions among variables by making use of an example from Glover (1964). Consider the simple integer programming problem:

$$\begin{aligned} &\text{Minimize } 9x_1 + 4x_2 + 8x_3 \\ &\text{subject to} \end{aligned}$$

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

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 which 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 can be found in more recent integer programming literature (see, for example, Schrijver (1986), Nemhauser and Wolsey (1988), Parker and Rardin (1988)).

3.7. Randomization versus systematic generation

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). This orientation continues to dominate most GA implementations. 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 favouring evaluation criteria that would otherwise be applied deterministically, or incorporating probabilistic objectives based on mathematical expectation, as in the earlier surrogate constraint proposals).

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 in the earlier illustration of Fig. 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. Similarly, by its organization, scatter search need 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 during the course of their generation.

3.8. Emerging connections

Explicit connections between the ideas underlying scatter search and those underlying GAs are beginning to be

examined in the work of Michalewicz (1993) and Mühlentbein (1993), and related ideas are investigated in Dorndorf and Pesch (1992), Battiti and Tecchiolli (1992), Moscato and Tinatti (1993), and Reeves (1993a, b). These explorations obtain results that motivate further examination of complementary features of the two procedures that have not yet been considered as a foundation for creating hybrid methods.

The possibilities for establishing useful bridges between these two approaches are enhanced by more recent strategies that have evolved as extensions of the scatter search notions. We examine these next.

4. Scatter search and path relinking

The process of creating linear combinations of two parent vectors, as previously noted, corresponds to generating points on the line segment joining these parents. When, in addition, discrete components are rounded to integer values, the points no longer lie on a simple line segment but describe a more complex path. By extension, we may envision a process of linking parent solutions by other types of paths—paths that may not lie in Euclidean space, but rather lie in a *neighbourhood space* defined by heuristic processes for moving from one solution to another.

The notion of tracing paths between selected solutions in neighbourhood space, as a means of generating other solutions worthy of consideration, was introduced as part of a tabu search strategy in Glover (1989), and is called *path relinking*. An appealing feature of this approach is that path relinking not only generalizes the types of combinations used by scatter search, but establishes deeper connections between *solution combination* strategies and *memory exploiting* strategies, as embodied in tabu search. The following development describes the basic elements of path relinking, building on the exposition of Glover and Laguna (1993).

4.1. Initial steps

We first consider the creation of paths that join two selected solutions x' and x'' , restricting attention to the part of the path that lies 'between' the solutions, producing a solution sequence $x' = x(1), x(2), \dots, x(r) = x''$. To reduce the number of options to be considered, we may create the solution $x(i+1)$ from $x(i)$ at each step by choosing a move that leaves a fewest number of moves remaining to reach x'' . This policy, even if applied without exception, can permit a significant number of alternative choices for generating the next solution at each step. Consequently, additional criteria are relevant to creating the path, as we indicate shortly.

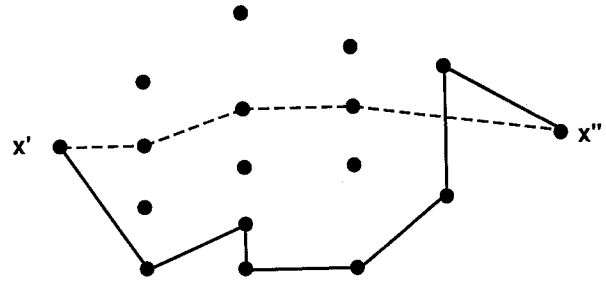


Fig. 2. Path relinking. Original path shown by heavy line; relinked path (one possibility) shown by dotted line

It is possible, as in applying scatter search, that x' and x'' were previously joined by a search trajectory produced by a heuristic method (or by a metaheuristic such as tabu search). In this event, the new trajectory created by path relinking is likely to be somewhat different than the one initially established, representing a 'more direct route' between the solutions. An illustration of this is given in Fig. 2.

To choose among the different paths that may be possible in going from x' to x'' , let $c(x)$ denote an objective function which is to be minimized (by choosing x over some feasible region). Selecting unattractive moves relative to $c(x)$, from the moves that are candidates to generate the path at each step, will tend to produce a final series of strongly improving moves to complete the path. Correspondingly, selecting attractive moves at each step will tend to produce lower quality moves at the end. (The last move, however, will be improving, or leave $c(x)$ unchanged, if x'' is selected to be a local optimum.) Thus, choosing best, worst or average moves, provides options that produce contrasting effects in generating the indicated sequence. An aspiration criterion may be used as in tabu search to override choices in the last two cases if a sufficiently attractive solution is available. (In general, it appears reasonable to select best moves at each step, and then reinitiate the process in the opposite direction by interchanging x' and x'' .)

The choice of one or more solutions $x(i)$ to become reference points for launching a new search phase preferably should depend not only on $c(x(i))$ but also on the values $c(x)$ of those solutions x that can be reached by a move from $x(i)$. We suggest that the process be additionally varied to allow solutions to be evaluated other than those that yield $x(i+1)$ closer to x'' . Aspiration criteria again may be used to decide whether such solutions qualify as candidates for selection.

To elaborate the process, let $x^*(i)$ denote a neighbour of $x(i)$ that yields a minimum $c(x)$ value during an evaluation step, excluding $x^*(i) = x(i+1)$. If the choice rules do not automatically eliminate the possibility $x^*(i) = x(h)$ for $h < i$, then a simple tabu restriction can be used to do this (e.g. as in Glover and Laguna (1993)). Then the method selects a solution $x^*(i)$ that yields a minimum value for $c(x^*(i))$ as a new point to launch the search. If only a

limited set of neighbours of $x(i)$ are examined to identify $x^*(i)$, then a superior least cost solution $x(i)$, excluding x' and x'' , may be selected instead. Early termination becomes possible upon encountering an $x^*(i)$ that yields $c(x^*(i)) < \min(c(x'), c(x''), c(x(p)))$, where $x(p)$ is the minimum cost $x(h)$ for all $h \leq i$. It should be noted that the procedure will continue without stopping if $x(i)$, in contrast to $x^*(i)$, yields a smaller $c(x)$ value than x' and x'' , since $x(i)$ effectively adopts the role of x' .

4.2. Variation and tunnelling

A variant of the path relinking approach starts with both endpoints x' and x'' simultaneously producing two sequences $x' = x'(1), \dots, x'(r)$ and $x'' = x''(1), \dots, x''(s)$. The choices in this case are designed to yield $x'(r) = x''(s)$, for final values of r and s . To progress toward this outcome when $x'(r) \neq x''(s)$, either $x'(r)$ is selected to create $x'(r+1)$, as by the criterion of minimizing the number of moves remaining to reach $x''(s)$, or $x'(s)$ is chosen to create $x''(s+1)$, as by the criterion of minimizing the number of moves remaining to reach $x'(r)$. From these options, the move is selected that produces the smallest $c(x)$ value, thus also determining which of r or s is incremented on the next step.

The path relinking approach benefits by a tunnelling strategy that allows a different neighbourhood structure to be used than in the standard search phase. For example, moves for path relinking may be periodically allowed that normally would be excluded due to creating infeasibility. Such a practice is protected against the possibility of becoming 'lost' in an infeasible region, since feasibility evidently must be recovered by the time x'' is reached. The tunnelling effect therefore offers a chance to reach solutions that might otherwise be bypassed. In the variant that starts from both x' and x'' , at least one of $x'(r)$ and $x''(s)$ may be kept feasible.

As in tabu search strategies for achieving intensification and diversification, it is appropriate to select the points x' and x'' by reference to clusters of solutions that are created according to criteria of similarity or affinity. Choosing x' and x'' from the same cluster then stimulates intensification, while choosing them from two 'widely separated' clusters stimulates diversification.

4.3. Extrapolated relinking

The path relinking approach goes beyond consideration of points 'between' x' and x'' in the same way that linear combinations extend beyond points that are expressed as convex combinations of two endpoints. In seeking a path that continues beyond x'' (starting from the point x') we invoke a tabu search concept, referring to sets of attributes associated with the solutions generated, as a basis for choosing a move that 'approximately' leaves the fewest moves remaining to reach x'' . Let $A(x)$ denote the set of

solution attributes associated with ('contained in') x , and let A_drop denote the set of solution attributes that are dropped by moves performed to reach the current solution $x'(i)$, starting from x' . (Such attributes may be components of the x vectors themselves, or may be related to these components by appropriately defined mappings.)

Define a *to-attribute* of a move to be an attribute of the solution produced by the move, but not an attribute of the solution that initiates the move. Similarly, define a *from-attribute* to be an attribute of the initiating solution but not of the new solution produced. Then we seem a move at each step to maximize the number of *to-attributes* that belong to $A(x'') - A(x(i))$, and subject to this to minimize the number that belong to $A_drop - A(x'')$. Such a rule generally can be implemented very efficiently by appropriate data structures.

Once $x(r) = x''$ is reached, the process continues by modifying the choice rule as follows. The criterion now selects a move to maximize the number of its *to-attributes* not in A_drop minus the number of its *to-attributes* that are in A_drop , and subject to this to minimize the number of its *from-attributes* that belong to $A(x'')$. The combination of these criteria establishes an effect analogous to that achieved by the standard algebraic formula for extending a line segment beyond an endpoint. (The secondary minimization criterion is probably less important in this determination.) The path then stops whenever no choice remains that permits the maximization criterion to be positive.

For neighbourhoods that allow relatively unrestricted choices of moves, this approach yields an extension beyond x'' that introduces new attributes, without reincorporating any old attributes, until no move remains that satisfies this condition. The ability to go beyond the limiting points x' and x'' creates a form of diversification analogous to that provided by the original scatter search approach. At the same time the exterior points are influenced by the trajectory that links x' and x'' .

4.4. Multiple parents

New points can be generated from multiple parents as follows. Instead of moving from a point x' to (or through) a second point x'' , we replace x'' by a collection of solutions X'' . Upon generating a point $x(i)$, the options for determining a next point $x(i+1)$ are given by the union of the solutions in X'' , or more precisely, by the union A'' of the attribute sets $A(x)$, for $x \in X''$. A'' takes the role of $A(x'')$ in the attribute-based approach previously described, with the added stipulation that each attribute is counted (weighted) in accordance with the number of times it appears in elements $A(x)$ of the collection. Still more generally, we may assign a weight to $A(x)$, which thus translates into a sum of weights over A'' applicable to each attribute, creating an effect analogous to that of

creating a weighted linear combination in Euclidean space. Parallel processing can be applied to operate on an entire collection of points $x' \in X'$ relative to a second collection $x'' \in X''$ by this approach. Further considerations that build on these ideas, and that go beyond the scope of our present development, are detailed in Glover (1991).

This multiparent path relinking approach generates new elements by a process that emulates the strategies of the original scatter search approach at a higher level of generalization. The reference to neighbourhood spaces makes it possible to preserve desirable solution properties (such as complex feasibility conditions in scheduling and routing), without requiring artificial mechanisms to recover these properties in situations where they may otherwise become lost. By expanding the concept of *solution combinations*, and providing new strategies for generating these combinations, path relinking gives a further basis for integrating ideas of genetic algorithms and scatter search.

5. Conclusion

Ideas originating in mathematical relaxations have evolved to become incorporated in the notions that underly scatter search, giving an approach to optimization that is highly compatible with the approach of genetic algorithms. The potential to exploit connections between scatter search and GAs is just beginning to be explored, and opportunities to create hybrid methods are abundant. Recent extensions of scatter search based on path relinking, and the associated use memory exploiting strategies based on tabu search, offer particular promise as a foundation for creating such hybrids, opening the door to classes of procedures that have features complementary to those normally studied in the GA domain. Determining the relative merits of different options, and identifying applications where these options are most effective, provide inviting avenues for future research.

References

- Aarts, E. H. L., Eiben, A. E. and van Hee, K. M. (1989). A general theory of genetic algorithms. *Computing Science Notes*, 89/8, Eindhoven University of Technology.
- Ackley, D. (1987). *A Connectionist Model for Genetic Hillclimbing*. Kluwer, Dordrecht. Academic Publishers.
- Bäck, T., Hoffmeister, F. and Schwefel, H. (1991). A survey of evolution strategies. In *Proceedings of the Fourth International Conference on Genetic Algorithms*, eds R. Belew and L. Booker, pp. 2–9. Morgan Kaufmann, San Mateo, CA.
- Battiti, R. and Tecchiolli, G. (1992). Parallel biased search for combinatorial optimization: genetic algorithms and tabu search. Technical Report No. 9207-02, Istituto Per La Ricerca Scientifica E Tecnologica, Italy.
- Beasley, J. E. (1983). Lagrangean relaxation, In *Modern Heuristic Techniques for Combinatorial Problems*, ed. C. Reeves, pp. 243–298. Blackwell Scientific Publishing, Oxford.
- Costa, D. (1992). An evolutionary tabu search algorithm and the NHL scheduling program. ORWP 92-11, Ecole Polytechnique Fédérale de Lausanne, Switzerland.
- Davis, L. (1989). Adapting operator probabilities in genetic algorithms. In *Proceedings of the Third International Conference on Genetic Algorithms*, pp. 61–69. Morgan Kaufmann, San Mateo, CA.
- Davis, L. (ED.) (1991). *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York.
- Dorndorf, U. and Pesch, E. (1992). Evolution Based Learning in a Job Shop Scheduling Environment. Technical Report, University of Limberg, The Netherlands.
- Everett, H. (1963). Generalized Lagrange multiplier method for solving problems of optimal allocation of resources. *Operations Research*, 11, 399–417.
- Fisher, M. L. (1973). Optimal solution of scheduling problems using generalized Lagrangean multipliers: Part I. *Operations Research*, 21, 1114–1127.
- Fisher, M. L. (1985). An applications oriented guide to Lagrangean relaxation. *Interfaces*, 15, 10–21.
- Fisher, M. L., Northrup, W. D. and Shapiro, J. F. (1975). Using duality to solve discrete optimization problems: theory and computation experience. *Mathematical Programming Study*, 3, 59–94.
- Fréville, A. and Plateau, G. (1992). FPBK92: An implicit enumeration code for the solution of the 0-1 bidimensional knapsack problem based on surrogate duality. *Graphs and Optimization Colloquium*, Grimentz, Switzerland.
- Fréville, A. and Plateau, G. (1993). An exact search for the solution of the surrogate dual of the 0-1 bidimensional knapsack problem. *European Journal of Operational Research*, 68, 413–421.
- Gavish, B. and Pirkul, H. (1985). Efficient algorithms for solving multiconstraint zero-one knapsack problems to optimality. *Mathematical Programming*, 31, 78–105.
- Geoffrion, A. M. (1972). Duality in nonlinear programming: a simplified applications-oriented development. In *Perspectives on Optimization*, ed. A. M. Geoffrion. Addison-Wesley, Reading, MA.
- Geoffrion, A. M. (1974). Lagrangean relaxation and its uses in integer programming. *Mathematical Programming Study*, 2, 82–114.
- Glover, F. (1964). A bound escalation method for the solution of integer linear programs. *Cahiers de Recherche Opérationnelle*, 6, 131–168.
- Glover, F. (1965). A multiphase-dual algorithm for the zero-one integer programming problem. *Operations Research*, 13, 879–919.
- Glover, F. (1968). Surrogate constraints. *Operations Research*, 16, 741–749.
- Glover, F. (1974). Heuristics for Integer Programming Using Surrogate Constraints. MSRS 74-6, College of Business, University of Colorado, July (also referenced in Glover (1975)).
- Glover, F. (1975). Surrogate constraint duality in mathematical programming. *Operations Research*, 23, 434–451.

- Glover, F. (1977). Heuristics for integer programming using surrogate constraints. *Decision Sciences*, **8**, 156–166.
- Glover, F. (1989). Tabu search, Part I. *ORSA Journal on Computing*, **1**, 190–206.
- Glover, F. (1991). Tabu search for nonlinear and parametric optimization (with links to genetic algorithms). To appear in *Discrete Applied Mathematics*.
- Glover, F. and Laguna, M. (1993). Tabu search. In *Modern Heuristic Techniques for Combinatorial Problems*. ed. C. Reeves, pp. 70–141. Blackwell Scientific Publishing, Oxford.
- Glover, F., Kelly, J. and Laguna, M. (1993). Genetic algorithms and tabu search: hybrids for optimization. To appear in *Computers and Operations Research*, ed. Y. Gupta.
- Glover, F., Laguna, M., Taillard, E. and de Werra, D. (EDS) (1993). Tabu search. Special issue of the *Annals of Operations Research*, **41**.
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, MA.
- Greenberg, H. J. (1973a). Bounding nonconvex programs with conjugates. *Operations Research*, **21**, 346–347.
- Greenberg, H. J. (1973b). The generalized penalty function surrogate model. *Operations Research*, **21**, 162–178.
- Greenberg, H. J. and Pierskalla, W. P. (1970). Surrogate mathematical programs. *Operations Research*, **18**, 924–939.
- Greenberg, H. J. and Pierskalla, W. P. (1973). Quasi-conjugate functions and surrogate duality. *Cahiers du Centre d'Etudes de Recherche Operationelle*, **15**, 437–448.
- Held, M. and Karp, R. M. (1970). The traveling salesman problem and minimum spanning trees. *Operations Research*, **18**, 1138–1162.
- Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI.
- Karwan, M. H. and Rardin, R. L. (1976). Surrogate dual multiplier search procedures in integer programming. School of Industrial Systems Engineering, Report Series No. J-77-13, Georgia Institute of Technology.
- Karwan, M. H. and Rardin, R. L. (1979). Some relationships between Lagrangean and surrogate duality in integer programming. *Mathematical Programming*, **17**, 230–334.
- Michalewicz, Z. (1992). *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, New York.
- Michalewicz, Z. (1993). Evolutionary computation techniques for nonlinear programming problems. Paper presented at the IFORS 93 Conference, Lisbon, Portugal.
- Michalewicz, Z. and Janikow, C. (1991). Genetic algorithms for numerical optimization. *Statistics and Computing*, **1**, 75–91.
- Michalewicz, Z., Vignaux, G. A. and Hobbs, M. (1991). A non-standard genetic algorithm for the nonlinear transportation problem. *ORSA Journal on Computing*, **3**, 307–316.
- Moscato, P. and Tinetti, R. (1993). Blending heuristics with a population based approach.
- Mühlenbein, H. (1993). Parallel genetic algorithms in combinatorial optimization. To appear in *Computer Science and Operations Research*. ed. Osman Balci. Pergamon Press, Oxford.
- Mühlenbein, H., Gorges-Schleuter, M. and Krämer, O. (1988). Evolution algorithms in combinatorial optimization. *Parallel Computing*, **7**, 65–88.
- Nemhauser, G. and Wolsey, L. (1988). *Linear and Combinatorial Optimization*. John Wiley, New York.
- Parker, G. and Rardin, R. (1988). *Discrete Optimization*. Academic Press, New York.
- Reeves, C. (1993a). *Modern Heuristic Techniques for Combinatorial Problems*. Blackwell Scientific Publishing, Oxford.
- Reeves, C. (1993b). Diversity and diversification in genetic algorithms: some connections with tabu search. Coventry University, United Kingdom.
- Schrijver, A. (1986). *Linear and Integer Programming*. John Wiley, New York.
- Spears, W. M. and DeJong, K. A. (1991). On the virtues of uniform crossover. In *4th International Conference on Genetic Algorithms*, La Jolla, CA.
- Ulder, N. L. J., Pesch, E., van Laarhoven, P. J. M., Bandelt, H. J. and Aarts, E. H. L. (1991). Genetic local search algorithm for the traveling salesman problem. In *Parallel Problem Solving from Nature*, eds R. Maenner and H. P. Schwefel, pp. 109–116. Springer-Verlag, Berlin.
- de Werra, D. and Hertz, A. (1989). Tabu search technique: a tutorial and an applications to neural networks. *OR Spectrum*, **11**, 131–141.
- Whitley, D. (1993). *Foundations of Genetic Algorithms 2*. Morgan Kaufmann, San Mateo, CA.
- Wright, A. H. (1990). Genetic algorithms for real parameter optimization. In *Foundations of Genetic Algorithms*, ed. G. Rawlins, pp. 205–218. Morgan Kaufmann, Los Altos, CA.