Chapter 1

METAHEURISTIC AGENT PROCESSES (MAPS)

Fred Glover[1] and Gary Kochenberger[2]

[1] *Leeds School of Business, University of ColoradoBoulder,Colorado 80309-0419*
fred.glover@colorado.edu

[2] *School of Business Administration, University of Colorado Denver, Colorado 80217-3364*
gary.kochenberger@cudenver.edu

Abstract: Agent-based models have had a remarkable impact in many areas of science, engineering and business. To achieve their full potential, however, these models must be extended to meet challenges of optimization that have so far been sidestepped or left unattended. Because classical optimization procedures are incapable of handling the complex problems that give rise to this challenge, a need arises for agent-based models to draw support from the field of metaheuristics.

Accordingly, this situation motivates the creation of Metaheuristic Agent Processes (MAPs) that integrate agent-based models with metaheuristic procedures, and thereby offer a means for achieving further advances through the use of agent-based technology. In this paper, we demonstrate that fundamental metaheuristic strategies already encompass inherent agent-based components, providing a natural foundation for the form of integration necessary to produce MAPs. In addition, we identify a particular class of discrete optimization models that exhibits useful links to agent-based systems, and whose successful applications invite further exploration within the MAP context.

1.1 INTRODUCTION: AGENT-BASED MODELS AND OPTIMIZATION

Agent-based Models (ABMs) are gaining widespread recognition for their role in analyzing complex activities. The underlying structure of ABMs varies, but they are generally conceived to consist of autonomous software objects that interact within an environment. Agents are often described as having behaviors and missions by which they may affect the environment as

well as each other, and they are notably subject to being combined to create interactive simulations and models.
Abstract characterizations of ABMs, however, may be viewed chiefly as "after-the-fact" attempts to group together ideas that are intuitively conveyed by the agent terminology. While a thoroughly precise and universally agreed-upon definition of agent-based models may not exist, the relevance of ABMs in science and industry is manifested in its diverse applications. These include explorations into the transmission of diseases, the operation of ecosystems, the dynamics of supply chains, the fluctuations and trends of financial markets, the behavior of economic sectors, the patterns of migrations, the flows of metropolitan traffic and the interactions of chemical and bio-physical processes.

Within these numerous and varied systems, two essential elements come conspicuously to the fore: the need for complex simulations and the need for highly adaptive optimization. The relevance of complex simulations[1] has been long been recognized, and has received extensive attention in agent-based modeling – as evidenced by the existence of public domain libraries for generating simulations from an agent-based perspective.[2] On the other hand, the relevance of optimization has been significantly underplayed. No doubt this is because the structure of many agent-based systems cannot easily be captured by classical optimization models, due to conditions of non-linearity, discreteness and/or uncertainty that are often present in these systems.

In fact, the current role of optimization in agent-based modeling is entirely analogous to the role it assumed within the general field of simulation only a few years ago, as a result of these same factors – inapplicability of classical models, non-linearity, combinatorial complexity and uncertainty. Within the simulation industry, practitioners struggled for years in an attempt to handle the compelling issues of optimization simply by means of trial-and-error analysis. The simulation literature often claimed to have "optimized" various system components, based on nothing more than a series of guesses and brute force re-trials. Today, this picture has dramatically changed, thanks to the newly-emerged metaheuristic procedures that are now routinely being used in the simulation industry, and that are creating solutions of vastly greater quality and utility than were previously possible. The leading

---

[1] Complexity in this case is manifested in the outcomes of the simulation, though not necessarily in the elements and operations that compose it.

[2] A prominent example is the Swarm Simulation System (www.swarm.org). An award-winning commercial authoring tool for creating agent-based models is provided by AgentSheets (www.agentsheets.com).

provider of this technology to the simulation industry, OptTek Systems, reports over 50,000 applications of its metaheuristic search software (e.g., see www.opttek.com).

In the same way as occurred in the earlier applications of optimization in the simulation area, optimization within agent-based models is still approached for the most part by resorting to a series of educated guesses about the values of various input control parameters and decision variables. There is no globally coordinated mechanism for identifying parameter values that yield outcomes of high quality. In particular, the possibility of conducting an intelligent search for high quality solutions by using an appropriate metaheuristic framework is still largely unrecognized.

1.2 METAHEURISTIC AGENT PROCESSES

A significant opportunity exists to expand the scope and power of agent-based models by integrating them with metaheuristics. We refer to the result of such integration as *Metaheuristic Agent Processes (MAPs).* From a strictly technical point of view, the creation of MAPs involves nothing revolutionary, since it corresponds to the same type of advance already made in the realm of simulation. Such a development is all the more natural because of the close alliance between simulation and agent-based models, where simulation is pervasively used to capture the dynamics and investigate the implications of many forms of ABMs. Taking advantage of this fact by creating metaheuristic agent processes to improve the quality and value of information derived from agent-based models would mark a significant step forward.

The integration required to produce effective MAPs rests on principles already well-known and applied within many segments of the metaheuristic community. Indeed, some metaheuristic procedures are founded on metaphors that call to mind the notions and terminology of agent-based systems, and some proponents of these metaheuristics have already sought to have their work viewed as a contribution to the ABM area.[3] However, such contributions are still limited in scope, and contributions of a more

---

[3] These metaheuristics are grouped by the label of “swarm intelligence” or “particle swarm optimization,” and widely portrayed by the metaphor of bees swarming about a hive. Interesting and perhaps unexpected bonds to certain other types of methods are evidenced by the fact that this search metaphor was originally introduced in the literature of tabu search and adaptive memory programming (see, e.g., Glover (1996) and Glover and Laguna (1997)). A website on particle swarm optimization can be found at www.particleswarm.com.

substantial nature are not only possible but greatly needed. The opportunity to make gains by the creation of MAPs rests on the same types of metaheuristic advances that have made it possible to handle the complex conditions of non-linearity, discreteness and uncertainty in other realms. Our thesis is that MAPS include agent-based processes of solving problems (i.e., agent-based algorithms) and also agent-based representatives of complex systems we try to optimize.

The next sections set out to accomplish three things. First, we demonstrate an intimate connection whereby certain long-standing metaheuristic strategies may be viewed as instances of agent-based processes themselves. From this standpoint, there are compelling precedents for a broader integration of metaheuristics and agent-based models to produce MAPs. Second, within this development we also identify recent innovations that hold promise for further enriching the realm of metaheuristic agent processes. Within this context, we discuss the opportunity for next steps that can usefully expand the application of agent-based models by their integration with metaheuristics. Finally, we demonstrate that a class of 0-1 quadratic optimization models has close ties to agent-based systems, and observe that the highly successful application of these models motivates a fuller exploration of their connection with ABMs.

1.3 METAHEURISTIC PROCESSES CONCEIVED AS AGENT-BASED SYSTEMS

We illustrate a few selected metaheuristic strategies that have conspicuous interpretations as agent-based systems. Notably, the first two of these strategies we discuss emerged long before the notions of agent-based models were popularized. On the basis of these interpretations, it will be clear that many other metaheuristic strategies can likewise be viewed as instances of an agent-based framework. Thus, while agent-based modeling and optimization have up to now remained somewhat insulated from each other, the two fields can productively be viewed as interrelated through the design of metaheuristics and, in particular, through the realm of MAPs.

We begin by stepping back in time to examine a set of strategies from the 1960s that has motivated the development of more recent ideas. In their original form, these strategies were designed to generate solutions within the setting of job shop scheduling by creating improved local decision rules. The first of these approaches (Crowston, et al., 1963) sought to create improved rules by selecting probabilistically from a collection of known rules so that different rules will be applied at different decision points throughout the process of generating a schedule constructively. As complete solutions (schedules) are produced by this approach, the decision rules that are more often used to create the solutions of higher quality receive greater increases in the probabilities for applying them in future passes.

The process can be viewed from as a metaheuristic agent process as shown in Fig. 1.1. The decision rules (of which there may be many) operate as agents, and at each step of constructing a solution the agents enter into a "probabilistic competition" to determine which rule is allowed to augment the current solution to create an expanded solution for the next stage. The process repeats until completing the generation a new solution, whereupon the updated probabilities are calculated and the procedure begins once more with the null solution, to launch another construction. (For simplicity, we do not try to show all connections in this or subsequent figures, or to identify stopping rules, as typically based on numbers of iterations and/or quality of solutions obtained.)

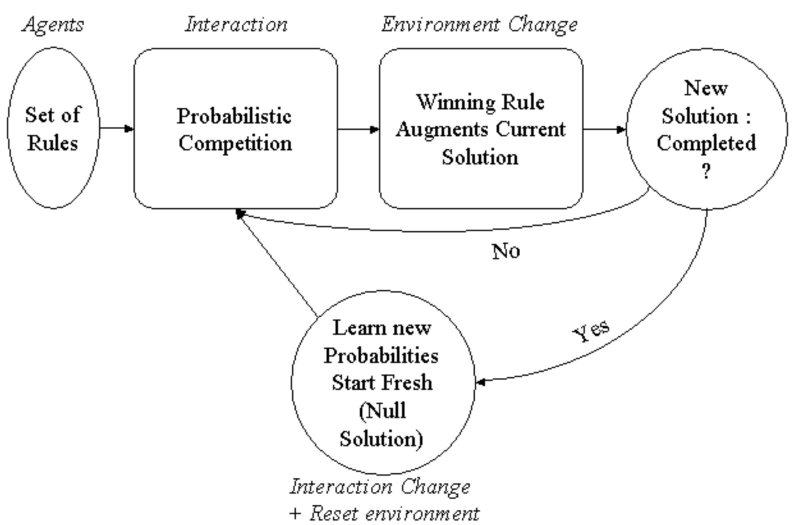


Figure 1.1. Probabilistic Decision Learning MAP

It is to be noted that this type of approach can readily be applied in many other settings, as a multi-start metaheuristic. Also, in the case where different decision rules are used to choose among alternative neighborhoods, the approach can be envisioned as an instance of a probabilistic form of strategic oscillation (Glover and Laguna, 1997; Gendreau, 2003) or as a variable neighborhood search procedure (Hansen and Mladenovic, 2003).

A related, but somewhat more effective method (Glover, 1963) replaces the approach of probabilistically choosing among a basic collection of rules by instead creating new rules that are explicitly different from all members of the collection, using a process of parametric combination. The basic rules are first re-expressed to yield an evaluation metric compatible with the notion of creating a weighted combination, and then each new pass systematically modifies the weights used to combine rules on preceding passes. The design of this approach later became encapsulated in surrogate constraint methods, by combining constraints instead of decision rules, and also more recently embodied in scatter search procedures (see the surveys respectively of Glover (2003) and Glover, Laguna and Marti (2000)).

From the perspective of a metaheuristic agent process, the rules to be combined may again be viewed as the agents. The diagrammatic outline in Fig. 1.2 also refers to the more general form of the process that includes surrogate constraint and evolutionary scatter search approaches, by allowing constraints and solutions to be agents instead of decision rules. In this type of process we may conceive of an additional agent entering the picture (a "marriage agent") as the means for creating the weighted combination of the components. Another generalization operates here, because the procedure is not only concerned with augmenting partial solutions (in successive stages of construction), but also with transforming complete solutions directly into other complete solutions. The augmentation repeats until creating a complete solution, while the direct transformation creates a new solution at each step. At this point, the new weights are produced for the next pass, and the procedure iterates until satisfying a chosen stopping criterion. An instance of this approach called a "ghost image process" has produced the best known solutions for an extensive test-bed of fixed charge transportation problems (Amini, 2003).

There are evidently a variety of possible variations that likewise fit within the same agent-based design, such as permitting different weights to be applied at different stages of construction or according to different environments. Likewise, as in the case of the process depicted in Fig. 1.1, the decision rules can refer to rules for choosing neighborhoods and the approach can also be used as a schematic for a multi-start method. Finally, we observe that the approaches of Fig. 1.1 and Fig. 1.2 can be merged, to create a probabilistic variation of a parametric combination process.

Within the framework of evolutionary approaches, an important extension of scatter search procedures is represented by path relinking methods (see, e.g.,

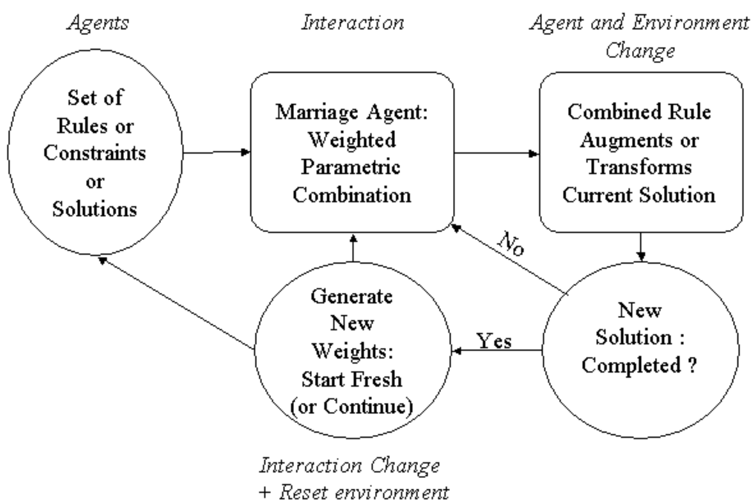


Figure 1.2. Parametric Decision Combination MAP

Glover, Laguna and Marti (2000) and Yagiura, Ibaraki and Glover (2002), Ribeiro and Resende (2005)). Path relinking combines solutions by generating paths in neighborhood spaces instead of Euclidean spaces as occurs in scatter search. To provide fuller generality, we treat rules and solutions alike as agents, thereby encompassing path relinking methods that use both transitional and constructive neighborhoods. (Transitional neighborhoods define moves that transform complete solutions into other complete solutions, while constructive neighborhoods define moves that transform incomplete (partial) solutions into more nearly complete solutions.) The same representation, under appropriate qualification, can also capture the approach of referent domain optimization (Glover and Laguna (1997) and Mautor and Michelon (1998)). The depiction of these approaches as metaheuristic agent processes is given in Fig. 1.3.

Path relinking using transitional neighborhoods is embodied in this diagram by focusing on solutions as agents. At each step, a subset selection process is applied (which can also be viewed as performed by an agent), and the solutions in the subset are joined by moving from selected *initiating solutions* through neighborhood space toward other members of the subset, which provide *guiding solutions* for determining the trajectory. New solutions are culled from this procedure by an intermediate selection step and subjected to an improvement process. Finally, an evaluation filter decides which of the resulting solutions enters the set of agents to be considered for the next round, by replacing previous agents that are

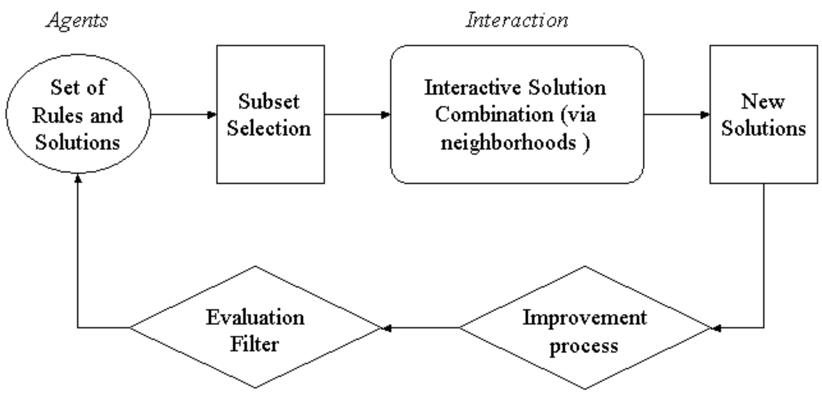


Figure 1.3. Path Relinking / Referent Domain MAP

dominated according to intensification and diversification criteria. (As previously noted, not all connections are shown, in order to keep the diagrams from being cluttered.) Applications of this approach are surveyed in Glover, Laguna and Marti (2000, 2003).

A slightly altered focus where rules also take the role of agents occurs in a form of path relinking involving the use of constructive neighborhoods. In this case, solutions and rules are intermingled, by a design where the guiding solutions provide "votes" (i.e., components of an overall evaluation) that determine which solution element is the next in sequence to be added by the constructive process. Destructive as well as constructive neighborhoods are typically incorporated in such designs. This type of approach has recently been applied effectively in the context of satisfiability problems by Hao, Lardeux and Saubion (2003), yielding new best solutions to a number of benchmark problems. A variant applied in conjunction with ejection chain neighborhoods by Yagiura et al. (2002) succeeds in generating solutions for generalized assignment problems that are not matched by any other procedure.

Referent Domain Optimization is captured by this diagram under the condition where the new solutions produced by the solution combination mechanism are subdivided into components (domains), and the improvement process tightly constrains some of these components while subjecting the

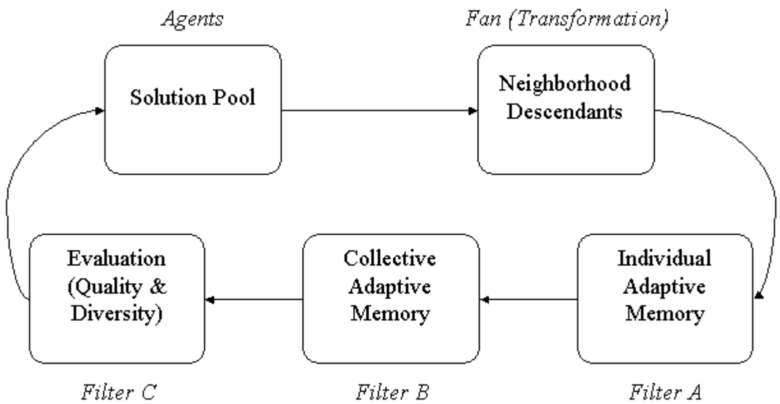


Figure 1.4. Filter-and-Fan MAP

remaining problem to an intensified improvement procedure, typically an exact solution method. Other forms of referent domain optimization sketc hed in Glover and Laguna (1997) can be represented by straightforward modifications of the diagram.

Our final illustration of a metaheuristic from the standpoint of an agent-based process concerns the Filter-and-Fan procedure (Glover, 1998; Greistorfer, Rego and Alidaee, 2003). This approach operates with solutions as agents to produce the MAP representation shown in Fig. 1.4, which portrays a single iterative stage of the procedure. Each iteration begins by performing a *fan step* whereby each agent (solution) generates a subset of descendants in its neighborhood. The solutions carry individual memories (of the type customarily used in tabu search) which are used to separate and remove certain descendants (Filter A). The system as a whole also carries an associated global memory that removes additional descendants (Filter B). Finally, evaluation criteria based on intensification and diversification eliminate certain remaining solutions (Filter C), and those solutions that survive the gauntlet generate an updated solution pool for re-applying the process. After a chosen number of steps, the method recovers a restricted subset of solutions from earlier steps that resulted in the best solution(s) in the final pool. The recovered solutions then compose the starting solution pool for the next stage of the process. The investigation of Greistorfer, Rego and Alidaee (2003) discloses that this approach proves exceedingly effective for solving facility location problems.

1.4 GENERAL OBSERVATIONS ON MAP REPRESENTATIONS

The representations of metaheuristics as MAPs illustrated in the preceding section are intended to be suggestive rather than exhaustive. Evidently, their form can be adapted to create agent-based representations of a variety of other metaheuristic approaches. For example, we may conceive associated constructions for representing processes that might be called Neural MAPs, Genetic MAPs, Evolutionary MAPs, Ant Colony MAPs, Variable Neighborhood MAPs, and so forth.

The depiction of metaheuristics as MAPs has the benefit of clarifying the connection that already exists between metaheuristic procedures and the realm of agent-based models. On the other hand, these representations also suffer a significant limitation, by embodying a level of abstraction that loses track of details that are critical for producing the most effective methods. Among a wide range of procedures that might be portrayed within the same representational framework, only a small subset will reach or surmount the level of performance achieved by the methods that have motivated these MAP diagrams.

The foregoing representations are also incomplete in another respect, resulting from their restricted focus on metaheuristics in isolation from other types of agent-based processes. The methods produced by this restricted focus might be called "antecedent MAPs," or α-MAPs, to distinguish them from the more ambitious MAPs that integrate metaheuristics with agent-based models of other forms. It is worth re-emphasizing that this integration is essential to accomplish the goal of bringing optimization to bear on ABMs.

Notably, an important step toward creating fully integrated MAPs has been effected by the creation of a library of functions, called the OptQuest Engine, that integrates optimization with simulations that have an entirely general form. The library is not limited to serving as a tool for academic research, but has been widely used in practical applications and has been adopted by nearly all major providers of commercial simulation software. (For background on these developments, see April et al. (2003a, 2003b), Laguna and Marti (2002), Kelly and Laguna (1999).) Consequently, it is a natural next step to structure such metaheuristic processes to handle the specific manifestations of simulation that occur in agent-based modeling. Adaptations of this form can be tailored to yield integrated MAPs that exploit the features of different classes of applications, thereby increasing their usefulness. The resulting higher-order MAPs afford the opportunity to significantly extend the capabilities of current agent-based methods, by making it possible to deal more effectively and comprehensively with environments attended by uncertain events, and to perform improved analyses of systems involving agents that behave according to probabilistically defined parameters. The ability of metaheuristics to handle complex nonlinearities and combinatorial relationships provides additional motivation for creating MAPs that go beyond the current "grope-in-the-dark" applications of what-if analysis in ABMs, and opens the door to the benefits of advanced optimization capabilities for agent-based modeling.

1.5 A FUNDAMENTAL AGENT-BASED OPTIMIZATION MODEL

We briefly sketch an optimization model that has a natural link to agent-based models, and that can be used to capture interactions of a variety of agent-based systems. The model, called the binary quadratic programming (BQP) problem, can be expressed as follows. Given an n x n matrix Q of constants, we seek an n-dimensional vector x of variables to

BQP: Minimize (or maximize) $x_o = xQx$

subject to

x binary

Although there are no constraints other than the binary restriction on x, standard transformations make it possible to include many types of constraints (including linear equality constraints) directly in the objective function by modifying the entries of Q. Well known applications of the BQP model include Quadratic Assignment Problems, Capital Budgeting Problems, Multi-dimensional Knapsack Problems, Task Allocation Problems (distributed computer systems), Maximum Diversity Problems, P-Median Problems, Asymmetric Assignment Problems, Symmetric Assignment Problems, Side Constrained Assignment Problems, Quadratic Knapsack Problems, Constraint Satisfaction Problems (CSPs), Set Partitioning Problems, Fixed Charge Warehouse Location Problems, Maximum Clique Problems, Maximum Independent Set Problems, Maximum Cut Problems, Graph Coloring Problems, Graph Partitioning Problems and a variety of others.

Note that in speaking of the BQP model as a *MAP model*, we are introducing a concept that goes beyond the customary notion of an agent-based process, which is restricted to refer to a type of algorithm or computational design.

From the current point of view, we consider that a model deserves (or benefits from) an agent-based interpretation if it can be expressed in a natural way as the outcome of an interaction among agents. The BQP model provides an “idealized form” of a MAP model in the following sense. First, in a simplified form, the model is concerned with possible behaviors of agents we index by the symbol j, whose actions are represented by a policy set that consists of two alternatives, coded by the values 0 and 1. The choice among these alternatives in this situation is represented by the assignment $x_j = 0$ and $x_j = 1$, respectively. The result of interaction among agents is captured in the Q matrix. More precisely, the Q matrix is the evaluator for all possible interactions. We seek a policy by each agent from its policy set that yields the best outcome according to the objective function.

Actually, the interactions specified in the Q matrix imply that the policy set for each agent is more general that suggested by the preceding simplified description. To identify this broader interpretation, we may consider a policy set P(j) for each agent j that consists of multiple options i ε P(j), where agent j chooses policy i if and only if $x_{ij} = 1$. In this case the x vector consists of the binary variables x_{ij} and for each j the problem contains the additional multiple choice restriction

$$\sum (x_{ij} : i \in P(j)) = 1.$$

The choice by agent j of a specific policy from P(j) determines the response to each possible policy choice by every other agent according to the entries of the Q matrix. The effects of all possible pairwise interactions among policy choices receive consideration in this manner. The resulting model can also be viewed in the context of game theory as a Discrete Option Game. The additional multiple choice restrictions, to handle the situation where some players can choose among more than two policies, have a form that permits them to be embedded within the Q matrix in a straightforward way to yield an instance of the BQP model that can be solved highly efficiently.

The interpretation of the BQP model in the agent-based setting, where it provides a fundamental class of agent-based models, is useful in relation to the discussion of MAPs in the preceding sections. (We caution that the term “agent based model” has acquired the connotation of not being a model at all in the sense we speak of here, since it refers simply to a way of characterizing a computational scheme.) Specifically, the agent-based interpretation attached to the BQP model establishes a connection that enhances the relevance of integrating metaheuristic processes with agent-based designs to produce MAPs for several reasons. First, metaheuristic methods have proved by far the most effective methods for solving BQP problems (see, e.g., Kochenberger et al., (2004)). Moreover, the typical perspective regarding agent-based approaches, which does not envision the possibility of optimizing over the range of agent behaviors, becomes greatly broadened by means of the BQP model, for which optimization is meaningful and achievable within a practical sense. Finally, we note that the BQP model provides a framework that captures key problem areas that many efforts previously described as "agent-based modeling," have sought to address. Applications in such diverse areas as organization change, team building, and the study of international conflicts (see for example the works of Levinthal (1997), Solow, et. al. (2002) and Axelrod and Bennett (1993)) which have typically been modeled in terms of Rugged Landscapes can alternatively be modeled and analyzed via BQP. Finally, there is another way in which the agent-based interpretation of the BQP model is relevant to our present concerns. As we will show shortly, the method we have used to solve the BQP model itself has a convenient description as a metaheuristic agent process. Before proceeding to such a description, however, we elaborate on features of the BQP model (and the outcomes of solving it) that further motivate its consideration as a fundamental class of models.

Robustness of BQP

The application potential of BQP is extraordinarily robust due to reformulation methods that enable certain constrained models to be re-cast in the form of BQP. Boris and Hammer (1991, 2002), Hammer and Rudeanu (1968), Hansen (1979), and Hansen et. al. (1993) show that any quadratic (or linear) objective in bounded integer variables and constrained by linear equations can be reformulated as a BQP model. This recasting into BQP is accomplished by imposing quadratic infeasibility penalties in place of the linear constraints as described below:

Transformation to the BQP Form

Many practical combinatorial optimization problems can be modeled as constrained optimization problems of the form

$$\min x_0 = xQx$$

subject to

$$Ax = b, \ \ x \ binary$$

The foregoing model accommodates both quadratic and linear objective functions since the linear case results when Q is a diagonal matrix (observing that $x_j^2 = x_j$ when x_j is a 0-1 variable). Problems with inequality constraints can also be put into this form by introducing bounded *slack variables* to convert the inequalities into equations, and representing these slack variables by corresponding binary expansions. The constrained quadratic optimization models are then converted into equivalent BQP models by adding a quadratic infeasibility penalty function to the objective function as an alternative to explicitly imposing the constraints $Ax = b$. The general approach to such re-casting, which we call Transformation 1, is given below:

Transformation 1. Let P be a positive scalar penalty value, to yield

$$\begin{aligned} x_0 &= xQx + P(Ax-b)^t(Ax-b) \\ &= xQx + xDx + c \\ &= x\hat{Q}x + c \end{aligned}$$

where the matrix D and the additive constant c result directly from the matrix multiplication indicated. Upon dropping the additive constant, the equivalent unconstrained version of our constrained problem becomes

$$BQP(PEN): \min x\hat{Q}x,\ x\ binary$$

From a theoretical standpoint, a suitable choice of the penalty scalar P can always be chosen so that the optimal solution to BQP(PEN) is the optimal solution to the original constrained problem. As reported in Kochenberger, et. al. (2004), valid and computationally stable penalty values can be found without difficulty for many classes of problems, and a wide range of such values work well.

In addition to the modeling possibilities introduced by Transformation 1, a very important special class of constraints that arise in many applications can be handled by an alternative approach, given below.

Transformation 2. This approach is convenient for problems with considerations that isolate two specific alternatives and prohibit both from being chosen. That is, for a given pair of alternatives, one or the other but not both may be chosen. If x_j and x_k are binary variables denoting whether or not alternatives j and k are chosen, the standard constraint that allows one choice but precludes both is:

$$x_j + x_k \le 1$$

Then, adding the penalty function $Px_j x_k$ to the objective function is a simple alternative to imposing the constraint in a traditional manner. For problems with a linear objective function having all nonnegative coefficients, at least one positive, the scalar P (with respect to Transformation 2) can be chosen as small as the largest objective function coefficient [5]. This penalty function has sometimes been used by to convert certain optimization problems on graphs into an equivalent BQP model (see Pardalos and Xue (1994)). Its potential application, however, goes far beyond these settings as demonstrated in this paper. Variable upper bound constraints of the form $x_{ij} \le y_i$ can be accommodated by Transformation 2 by first replacing each y_i variable by 1 – y_i', where y_i' is the complementary variable that equals 1 when y_i = 0 and equals 0 when y_i = 1. The opportunity to employ this modeling device in the context of Transformation 2 makes it possible to conveniently model a variety of additional problem types.

The constraint associated with Transformation 2 appears in many important applications which leads us to single it our here as an important alternative to Transformation 1. We note, however, that many other problem-specific special cases exist that yield quadratic equivalent representations. We illustrate this later in the paper when we discuss results we have obtained for the max 2-SAT problem.

Examples

Before highlighting some of the solution methods reported in the literature for BQP, we give two small examples from classical NP-hard problem settings to provide concrete illustrations.

Example 1: Set Partitioning

The classical set partitioning problem is found in applications that range from vehicle routing to crew scheduling. As an illustration, consider the following small example:

$$\min x_0 = 3x_1 + 2x_2 + x_3 + x_4 + 3x_5 + 2x_6$$

subject to

$$\begin{aligned} x_1 + x_3 + x_6 &= 1 \\ x_2 + x_3 + x_5 + x_6 &= 1 \\ x_3 + x_4 + x_5 &= 1 \\ x_1 + x_2 + x_4 + x_6 &= 1 \end{aligned}$$

all x_j binary.

Applying Transformation 1 with P =10 gives the equivalent BQP model:

$$BQP(PEN): \min x\hat{Q}x,\ x\ binary$$

where the additive constant, c, is 40 and

$$\hat{Q} = \begin{bmatrix} -17 & 10 & 10 & 10 & 0 & 20 \\ 10 & -18 & 10 & 10 & 10 & 20 \\ 10 & 10 & -29 & 10 & 20 & 20 \\ 10 & 10 & 10 & -19 & 10 & 10 \\ 0 & 10 & 20 & 10 & -17 & 10 \\ 20 & 20 & 20 & 10 & 10 & -28 \end{bmatrix}$$

This simple example of BQP(PEN) can be solved by any of a variety of methods. (The illustrative problems of this paper are solved by the Tabu Search method of Glover et al. [11,12], and solution statistics for benchmark test problems are cited later.) In this case an optimal solution is given by $x_1 = x_5 = 1$, (all other variables equal to 0) for which $x_0 = 6$. In the straightforward application of Transformation 1 to this example, it is to be noted that the replacement of the original problem formulation by the BQP(PEN) model did not involve the introduction of new variables. In many applications, Transformation 1 and Transformation 2 can be used in concert to produce an equivalent BQP model, as demonstrated next.

Example 2: The K-Coloring Problem

Vertex coloring problems seek to assign colors to nodes of a graph such that adjacent nodes are assigned different colors. The K-coloring problem attempts to find such a coloring using exactly K colors. A wide range of applications ranging from frequency assignment problems to printed circuit board design problems can be represented by the K-coloring model.

Such problems can be modeled as satisfiability problems using the assignment variables as follows:

Let x_{ij} to be 1 if node i is assigned color j, and to be 0 otherwise.

Since each node must be colored, we have

$$\sum_{j=1}^{K} x_{ij} = 1 \quad i = 1,\ldots, n \tag{1.1}$$

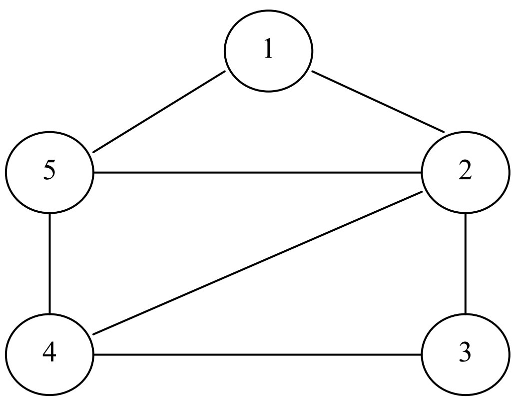
where n is the number of nodes in the graph. A feasible coloring requires that adjacent nodes are assigned different colors. This is accomplished by imposing the constraints

$$x_{ip} + x_{jp} \le 1 \quad p = 1,\ldots, K \tag{1.2}$$

for all adjacent nodes (i,j) in the graph.

This problem can be re-cast into the form of BQP by using Transformation 1 on the assignment constraints of (1.1) and Transformation 2 on the adjacency constraints of (1.2). No new variables are required. Since the model consisting of (1.1) and (1.2) has no explicit objective function, any positive value for the penalty P will do. The following example gives a concrete illustration of the re-formulation process.

Consider the following graph and assume we want find a feasible coloring of the nodes using 3 colors.



Our satisfiablity problem is that of finding a feasible binary solution to:

$$x_{i1} + x_{i2} + x_{i3} = 1 \quad i = 1, 5 \tag{1.3}$$

$$x_{ip} + x_{jp} \le 1 \quad p = 1, 3 \tag{1.4}$$

(for all adjacent nodes i and j)

In this traditional form, the model has 15 variables and 26 constraints. To recast this problem into the form of BQP, we use Transformation 1 on the equations of (1.3) and Transformation 2 on the inequalities of (1.4). Arbitrarily choosing the penalty P to be 4, we get the equivalent problem:

$$BQP(Pen): \min x\hat{Q}x$$

where the $\hat{Q}$ matrix is:

$$\hat{Q} = \begin{bmatrix} -4 & 4 & 4 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 \\ 4 & -4 & 4 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 \\ 4 & 4 & -4 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 \\ 4 & 0 & 0 & -4 & 4 & 4 & 4 & 0 & 0 & 4 & 0 & 0 & 4 & 0 & 0 \\ 0 & 4 & 0 & 4 & -4 & 4 & 0 & 4 & 0 & 0 & 4 & 0 & 0 & 4 & 0 \\ 0 & 0 & 4 & 4 & 4 & -4 & 0 & 0 & 4 & 0 & 0 & 4 & 0 & 0 & 4 \\ 0 & 0 & 0 & 4 & 0 & 0 & -4 & 4 & 4 & 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & 0 & 4 & -4 & 4 & 0 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 & 4 & 4 & -4 & 0 & 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 0 & 4 & 0 & 0 & -4 & 4 & 4 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & 0 & 0 & 4 & 0 & 4 & -4 & 4 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 4 & 4 & 4 & -4 & 0 & 0 & 4 \\ 4 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & -4 & 4 & 4 \\ 0 & 4 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 4 & -4 & 4 \\ 0 & 0 & 4 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 4 & 4 & 4 & -4 \end{bmatrix}$$

Solving this unconstrained model, $x\hat{Q}x$, yields the feasible coloring:

$$x_{11}, x_{22}, x_{33}, x_{41}, x_{53}, = 1 \ \ all\ \ other\ \ x_{ij} = 0$$

This approach to coloring problems has proven to be very effective for a wide variety of coloring instances from the literature. An extensive presentation of the xQx approach to a variety of coloring problems, including a generalization of the K-coloring problem considered here, is given in Kochenberger, Glover, Alidaee and Rego (2002)

Solution Approaches to BQP

Due to its computational challenge and application potential, BQP has been the focus of a considerable number of research studies in recent years, including both exact and heuristic solution approaches. Recent papers report on the branch and bound (exact) approaches as well as a variety of modern heuristic methods including simulated annealing, genetic algorithms, tabu search, and scatter search. (See Kochenberger, et. al. (2004) for references to these and other works.) Each of these approaches exhibits some degree of success. However, the exact methods degrade rapidly with problem size, and have meaningful application to general BQP problems with no more than 100 variables. (A notable exception to this for the Ising spin glass problem is discussed in De Simone, et. al. (1995). For larger problems, heuristic methods are usually required. Several proposed heuristics, including the DDT method of Boros, Hammer and Sun (1989) and the “one-pass” procedures of Glover, Alidaee, Rego and Kochenberger (2002) have proven to be effective in certain instances. Two methods we have found to be particularly successful for a wide variety of problems are based on tabu search (see Glover and Laguna (1997), Glover, et. al. (1999) and Glover, et. al. (1998)) and on the related evolutionary strategy scatter search of Glover (1998). In the following section we highlight our tabu search approach which was used to produce the computational results referenced later in this paper.

Although not pursued by us here, we note that an alternative approach is to solve BQP as a continuous non-linear optimization problem within the unit cube. This allows other heuristic/approximation methods based on continuous optimization methodologies to be applied (see Boris and Hammer (1991), Boris and Prekopa (1989) and Rosenberg (1972)).

Tabu Search Overview as a MAP process

Our TS method for BQP is centered around the use of strategic oscillation, which constitutes one of the primary strategies of tabu search. We offer the description below as an example of a MAP solution process that operates via the interaction of several functional agents.

The method alternates between constructive phases that progressively set variables to 1 (whose steps we call “add moves”) and destructive phases that progressively set variables to 0 (whose steps we call “drops moves”). The add moves are created by a “constructive agent” who identifies high quality potential add moves from an environment of available options. In a similar fashion, drop moves are created and managed by a “destructive agent.” To control the underlying search process, we use a memory structure that is updated at *critical events,* characterized by conditions that generate a subclass of locally optimal solutions. Solutions corresponding to critical events are called *critical solutions*. These functions are handled by a "critical event agent" that identifies when critical events occur and then performs the appropriate updates of memory.

A parameter *span* is used to indicate the amplitude of oscillation about a critical event. The operations involving this parameter are managed by a "span agent" that begins by setting *span* equal to 1 and gradually increases it to some limiting value. For each value of *span,* the span agent directs the constructive and destructive agents to perform a series of constructive and destructive phases, in alternation, before progressing to the next value. When *span* reached its limiting point, its guiding agent reverses the process so that *span* is gradually decreased in value, once again accompanied by invoking a series of alternating constructive and destructive phases. When *span* reaches a value of 1, a *complete span cycle* has been completed and the span agent launches the next cycle.

Information stored at critical events is used by a "tabu neighborhood restriction agent" (or simply "tabu agent," for short) to influence the search process by penalizing potentially attractive add moves during a constructive phase and inducing drop moves during a destructive phase. These penalties and inducements are associated with assignments of values to variables in recent critical solutions. The tabu agent also uses cumulative critical event information to introduce a subtle long term bias into the search process by means of additional penalties and inducements similar to those discussed above.

The activities of these agents are orchestrated by the direction of a "macro managing agent" that provides the coordination required for a successful search process. A complete description of the framework for our metaheuristic method is given in Glover, Kochenberger, Alidaee and Amini (1999).

Relevance of the MAP Interpretation for the BQP Problem

Evidently, the tabu search procedure for the BQP problem can be described without reference to a MAP framework, as in the case of other metaheuristic agent processes, and indeed as in the case of agent-based processes generally, which can readily be formulated as distributed computational designs where the activities of the agents are simply the functions of various subroutines. However, there is a virtue in the agent-based formulation that comes from its emphasis on processes that have a natural interpretation as being carried out by certain guiding entities that the literature has come to label with the agent terminology. Such an emphasis invites the designers of the associated methods to organize them in certain modularly structured ways that proves useful for visualizing their function and for extending them to create more advanced versions. For example, a new function can be provided by introducing a new agent, together with rules for interacting with the problem environment and selected other agents. (The agent-based literature often refers to agent-based processes as if the agents operate in complete autonomy from each other. This is an oversimplification that applies only under restricted circumstances.)

The advantages that come from using an agent-based orientation to describe and structure various computational processes lead us to anticipate that advantages may also accrue to ferreting out structures within mathematical models that can be interpreted from an agent-based perspective, as we have done with the BQP model. We now describe the outcomes that further suggest the BQP model may occupy a privileged position among the realm of models to which an agent-based interpretation can be usefully applied.

Computational Experience

Our results of applying the tabu search and associated scatter search metaheuristics to combinatorial problems recast in BQP form have been uniformly attractive in terms of both solution quality and computation times. Although our methods are designed for the completely general form of BQP, without any specialization to take advantage of particular types of problems reformulated in this general representation, our outcomes have typically proved competitive with or even superior to those of specialized methods designed for the specific problem structure at hand. By way of illustration, we present some representative results for a set of max 2-SAT test problems taken from the literature. Details of our experience with other problems will appear in future papers.

Max 2-SAT Results

Several authors (Hammer and Rudeanu (1968), Hansen and Jaumard (1990), Boros and Hammer (2002)) have established the connection between SAT problems and nonlinear penalty functions. The special case of Max 2-SAT is particularly well suited for this approach as it leads naturally to an xQx representation. Our experience, as shown below, indicates that this is a very attractive way to approach this class of problems.

For a 2-SAT problem, a given clause can have zero, one, or two negations, each with a corresponding (classical) linear constraint. Each linear constraint, in turn, has an exact quadratic penalty that serves as an alternative to the linear constraint. The three possibilities and their constraint/penalty pairs are:

(a) No negations:

Classical constraint: $x_i + x_j \geq 1$
Exact Penalty: $(1 - x_i - x_j + x_i x_j)$

(b) One negation:

Classical constraint: $x_i + \bar{x}_j \geq 1$

Exact Penalty: $(x_j - x_i x_j)$

(c) Two negations:

Classical constraint: $\bar{x}_i + \bar{x}_j \geq 1$

Exact Penalty: $(x_i x_j)$

It is easy to see that the quadratic penalties shown are zero for feasible solutions and positive for infeasible solutions. Thus, these special penalties can be used to readily construct a penalty function of the form of xQx simply by adding the penalties together. We have found this approach to be very effective. Table 1.1 shows the results we obtained via this approach on a set of test problems from the literature.

As shown in the table, by re-casting each Max 2-SAT instance into the form of xQx and solving the resulting unconstrained quadratic binary program with our Tabu Search heuristic, we were able to find best known solutions very quickly to all test problems considered. By way of contrast, the method of Borchers and Furman took a very long time on several problems and was unable to find best known results for several instances in the allotted 12 hour time limit. In addition to the problems of Table 1.1 above, we have successfully applied this approach to randomly generated problems with as many as 1000 variables and more than 10,000 clauses where best known results are found in roughly one minute of computation time.

The results shown in Table 1.1 above serve as strong evidence of the attractiveness of the xQx approach for the problems considered. Considering both solution quality and the time taken to produce these solutions, this approach is very competitive with special purpose methods constructed specifically for max 2-Sat problems. We note in passing that similar performance relative to special purpose methods has been obtained for the other problem classes singled out earlier in the paper as well.

Table 1.1. Problems from Borchers & Furman (1999)

n	m	Best known solution	xQx solution	xQx time	Maxsat[3] solution	Maxsat time
50	100	4	4	< 1	4	.4
50	150	8	8	< 1	8	1.5
50	200	16	16	< 1	16	116.2
50	250	22	22	< 1	22	652.4
50	300	32	32	< 1	32	8,763
50	350	41	41	< 1	NA	> 12 hr
50	400	45	45	< 1	NA	> 12 hr
50	450	63	63	< 1	NA	> 12 hr
50	500	66	66	< 1	NA	> 12 hr
100	200	5	5	< 2	5	3.2
100	300	15	15	< 2	15	13,770
100	400	29	29	< 2	NA	> 12 hr
100	500	44	44	< 2	NA	> 12 hr
100	600	?	65	< 2	NA	> 12 hr
150	300	4	4	< 3	4	4.1
150	450	22	22	< 3	NA	> 12 hr
150	600	38	38	< 3	NA	> 12 hr

Remarks:

1. All times in seconds unless noted otherwise.
2. Maxsat is an exact method developed by Borchers & Furman
3. Maxsat results obtained on IBM RS/6000-590
4. xQx results obtained on a 1.6 MHZ PC.
5. Each xQx run was for 50 SPAN cycles
6. Problem 100_600 was previously unsolved.

Summary

We have demonstrated how a variety of disparate combinatorial problems can be solved by first re-casting them into the common modeling framework of the unconstrained quadratic binary program. Once in this unified form, the problems can be solved effectively by adaptive memory tabu search metaheuristics and associated evolutionary (scatter search) procedures. We are currently solving problems via BQP with more than 50,000 variables in the quadratic representation and are working on enhancements that will permit larger instances to be solved.

Our findings challenge the conventional wisdom that places high priority on preserving linearity and exploiting specific structure. Although the merits of such a priority are well-founded in many cases, the BQP domain appears to offer a partial exception. In forming BQP(PEN), we destroy any linearity that the original problem may have exhibited. Moreover, any exploitable structure that may have existed originally is "folded into" the $\hat{Q}$ matrix, and the general solution procedure we apply takes no advantage of it. Nonetheless, our solution outcomes have been remarkably successful, yielding results that rival the effectiveness of the best specialized methods.

This combined modeling/solution approach provides a unifying theme that can be applied in principle to all linearly constrained quadratic and linear programs in bounded integer variables, and the computational findings for a broad spectrum of problem classes raises the possibility that similarly successful results may be obtained for even wider ranges of problems. As our methods for BQP continue to improve with ongoing research, the BQP model offers a representational tool of particular promise.

The novel fact that this model has a natural connection with agent-based systems, and provides an idealized instance of a MAP model, invites its exploration within alternative contexts that exploit the links to agent-based processes in additional ways.

REFERENCES

Amini, M. (2003) "Computational Study of Solving Fixed Charge Transportation Problems Using Ghost Image Processes," School of Business, University of Memphis, TN.

Axelrod, R, and Bennett, S, "A Landscape Theory of Aggregation," *British Journal of Political Science*, 1993, 23, 211-233.

April, J., F. Glover and J. Kelly (2003a) "Optfolio – A Simulation Optimization System for Project Portfolio Planning," *Proceedings of the 2003 Winter Simulation Conference*, S. Chick, F. Sanchez, D. Ferrin and D. Morrice, eds.

April, J., F. Glover, J. Kelly and M. Laguna (2003b) "Practical Introduction to Simulation Optimization," *Proceedings of the 2003 Winter Simulation Conference*, S. Chick, F. Sanchez, D. Ferrin and D. Morrice, eds.

Borchers, B. and J. Furman, "A Two-Phase Exact Algorithm for Max-SAT and Weighted Max SAT," *J. of Combinatorial Optimization,* 2, 299-306 (1999).

Boros, E., and P. Hammer, "The Max-Cut Problem and Quadratic 0-1 Optimization, Polyhedral Aspects, Relaxations and Bounds," *Annals of OR*, 33, 151-225 (1991).

Boros, E. and P. Hammer, " Pseudo-Boolean Optimization," *Discrete Applied Mathematics*, 123(1-3), 155-225 (2002).

Boros, E, P. Hammer, and X, Sun, "The DDT Method for Quadratic 0-1 Minimization," RUTCOR Research Center, RRR 39-89, 1989.

Boros, E. and A. Prekopa, "Probabilistic Bounds and Algorithms for the Maximum Satisfiability Problem," *Annals of OR,* 21 (1989), pp. 109-126.

Crainic, T.G. and M. Toulouse (2003) "Parallel Strategies for Meta-Heuristics," Chapter 17 of *Handbook of Metaheuristics*, G. Kochenberger and F. Glover, eds., Kluwer Academic Publishers.

Crowston, W., F. Glover, G. Thompson and J. Trawick (1963) "Probabilistic and Parametric Learning Combinations of Local Job Shop Scheduling Rules," *ONR Research Memorandum No. 117*, GSIA, Carnegie-Mellon University, Pittsburgh, PA.

De Simone, C. M. Diehl, M. Junger, P. Mutzel, G. Reinelt, and G. Rinaldi, "Exact Ground State4s of Ising Spin Glasses: New Experimental Results with a Branch and Cut Algorithm," *Journal of Statistical Physics,* 80, 487-496 (1995).

Dorigo, M. and T. Stutzle (2003). "The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances," Chapter 9 of *Handbook of Metaheuristics*, G. Kochenberger and F. Glover, eds., Kluwer Academic Publishers.

Gendreau, M. (2003) "An Introduction to Tabu Search," Chapter 2 of *Handbook of Metaheuristics,* G. Kochenberger and F. Glover, eds., Kluwer Academic Publishers.

Glover, F, and M. Laguna, "Tabu Search," Kluwer Academic Publishers, 1997.

Glover, F., "A Template for Scatter Search and Path Relinking," School of Business, University of Colorado, Technical Report, February 1998.

Glover, F., B. Alidaee, C. Rego, and G. Kochenberger, "One-Pass Heuristics for Large-Scale Unconstrained Binary Quadratic Programs," *EJOR 137, pp. 272-287*, 2002.

Glover, F., G. Kochenberger, B. Alidaee, and M.M. Amini, "Tabu with Search Critical Event Memory: An Enhanced Application for Binary Quadratic Programs," In: *MetaHeuristics: Advances and Trends in Local Search Paradigms for Optimization,* (Eds.) S. Voss, S. Martello, I. Osman, and C. Roucairol. Kluwer Academic Publisher, Boston, 1999.

Glover, F., G. Kochenberger., and B. Alidaee, "Adaptive Memory Tabu Search for Binary Quadratic Programs," Management Science, 44:3, 336-345, 1998.

Glover, F. (1963) "Parametric Combinations of Local Job Shop Scheduling Rules," Chapter IV, *ONR Research Memorandum No. 117*, GSIA, Carnegie-Mellon University, Pittsburgh, PA.

Glover, F. (1996) "Tabu Search and Adaptive Memory Programming – Advances, Applications and Challenges," *Interfaces in Computer Science and Operations Research*, R. Barr, R. Helgason and J. Kennington (eds.). Kluwer Academic Publishers, pp. 1-75.

Glover, F. (1998) "A Template for Scatter Search and Path Relinking," *Artificial Evolution*, Lecture Notes in Computer Science, 1363, J.-K. Hao, E. Lutton, E. Ronald , M. Schoenauer and D. Snyers, Eds. Springer, pp. 13-54.

Glover, F. (2003) "Tutorial on Surrogate Constraint Approaches for Optimization in Graphs," *Journal of Heuristics*, Volume 9, No. 3., pp. 175-227.

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

Glover, F., M. Laguna and R. Marti (2000) "Fundamentals of Scatter Search and Path Relinking," *Control and Cybernetics*, Volume 29, No 3, pp. 653-684.

Glover, F., M. Laguna and R. Marti (2003) "Scatter Search and Path Relinking: Advances and Applications," Chapter 1 of *Handbook of Metaheuristics*, G. Kochenberger and F. Glover, eds., Kluwer Academic Publishers.

Greistorfer, P., C. Rego and B. Alidaee (2003) "A Filter-and-Fan Aproach for the Facility Location Problem," *Proceedings of MIC2003*.

Hammer, P., and S. Rudeanu, *Boolean Methods in Operations Research,* Springer-Verlag, New York, 1968.

Hansen, P.B., "Methods of Nonlinear 0-1 Programming," *Annals Discrete Math,* vol. 5, pp.53-70, 1979.

Hansen, P, B. Jaumard., and V. Mathon, "Constrained Nonlinear 0-1 Programming," *INFORMS Journal on Computing*, 5:2, 97-119, 1993.

Hansen, P. and B. Jaumard, "Algorithms for the Maximum Satisfiability Problem," *Computing*, 44, 279-303 (1990).

Hansen, P. and M. Mladenovic (2003) "Variable Neighborhood Search," Chapter 6 of *Handbook of Metaheuristics*, G. Kochenberger and F. Glover, eds., Kluwer Academic Publishers.

Hao, J.K., F. Lardeux and F. Saubion (2003) "Evolutionary Computing for the Satisfiability Problem," *Lecture Notes in Computer Science,* Volume 2611, pp. 258-268, Springer.

Kelly, J. and M. Laguna (1999) "New Advances Wedding Simulation and Optimization," *Proceedings of the 1999 Winter Simulation Conference*, D. Kelton, ed.

Kochenberger, G., F. Glover, B. Alidaee, and C. Rego, "A Unified Modeling and Solution Framework for Combinatorial Optimization Problems," *OR Spectrum,* 26, pp. 237-250, (2004).

Kochenberger, G., F. Glover, B. Alidaee, and C. Rego, " An Unconstrained Quadratic Binary Programming Approach to the Vertex Coloring Problem," Working Paper, University of Colorado at Denver, 2002.

Kochenberger, G., F. Glover, B. Alidaee and C. Rego (2003) "Solving Combinatorial Optimization Problems via Reformulation and Adaptive Memory Metaheuristics," *Revolutionary Visions in Evolutionary Computation*, A. Menon and D. Doldberg, eds., Kluwer Academic Publishers, forthcoming.

Laguna, M. and R. Martí (2002) "The OptQuest Callable Library," *Optimization Software Class Libraries*, Stefan Voss and David L. Woodruff (eds.), Kluwer Academic Publishers, Boston, pp. 193-218.

Levinthal, D (1997) "Adaptation on Rugged Landscapes," Management Science, 43, 934-950.

Mautor, T, and P. Michelon (1998) "A neglected meta-heuristic: Referent Domain Optimization", 16th European Conference on Operational

Research (EURO XVI), Bruxelles.

Pardalos, P, and J. Xue, “The Maximum Clique Problem,” *The Journal of Global Optimization,* 4, 301-328, 1994.

Resende, M.G.C. and C.C. Ribeiro (2005) “GRASP with path relinking: recent advances and applications,” *Metaheuristics: Progress as Real Problem Solvers*, T. Ibaraki, K. Nonobe and M. Yagiura, (Eds.), Kluwer Academic Publishers.

Rosenberg, I.G., “0-1 Optimization and Non-linear Programming.” *Revue Francaise d’Automatique, d’Informatique et de Rescherche Operationnelle,* (Série Blueu) 2, 95-97, (1972).

Solow, D, Vairaktarakis,G, Piderti, S.K, Tsai, M-C, “ Managerial Insights into the Effects of Interactions on Replacing Members of a Team,” Management Science, 2002, 48, 1060-107.

Yagiura, M., T. Ibaraki and F. Glover (2002) “A Path Relinking Approach for the Generalized Assignment Problem,” *Proc. Int. Sym. on Scheduling*, pp. 105-108 (longer version available from http://www-or.amp.i.kyoto-u.ac.jp/~yagiura/papers/prgap_abst.html).