Surrogate Branching: Parametric Relaxation for Mixed Integer Optimization

Fred Glover
School of Engineering and Leeds School of Business,
University of Colorado
fred.glover@colorado.edu

Said Hanafi
Mechanical engineering and Computer Science
University of Valenciennes
Said.Hanafi@univ-valenciennes.fr

Abstract

Surrogate Branching (SB) methods in mixed integer optimization provide a staged parametric relaxation of customary branching methods used in branch-and-bound and branch-and-cut algorithms. SB methods operate by forming surrogate constraints composed of non-negative linear combinations of component inequalities of three types: (1) ordinary branching inequalities, (2) redundant inequalities involving bounds on variables, and (3) the strictly redundant inequality $0 \leq 1$. The usefulness of surrogate constraint relaxations and their associated duality theory in mixed integer optimization acquires a new scope through these surrogate branching inequalities, by allowing branching decisions to be progressively compounded, and parametrically staged in strength, as a function of the degree of separation desired.

1. Introduction

We write the mixed integer programming (MIP) problem in the form

$$\begin{array}{lll} \textbf{MIP}: & \text{Maximize} & x_o = cx \\ & \text{subject to} & Ax \leq b \\ & & U \geq x \geq 0 \text{ and } x_j \text{ integer, } j \in N \end{array}$$

where N is a non-empty subset of the index set for the x vector. As in the case of branch-and-bound (B&B) and branch-and-cut (B&C) methods generally, we are interested in solving a linear programming relaxation of MIP, and employ a strategy of adjoining inequalities to partition the solution space or to generally produce a new version of MIP whose LP relaxation is closer to being integer feasible. In customary B&B and B&C approaches this is done in a manner that assures an optimal integer solution to the original problem will remain accessible at all stages (in at least one of the partitions created). Although the methods we consider can be implemented to provide such assurance, they are conceived with the goal of finding very high quality (optimal or near optimal) integer solutions with a very efficient expenditure of effort. Thus, the strategies we propose, which we call Surrogate Branching (SB) methods, are designed to be implemented as metaheuristics as well as exact algorithms.

Surrogate Branching methods have their origins in two developments: surrogate constraint relaxations for mixed integer programming and Local Branching (LB) strategies for 0-1 integer programming. (See Glover (2002) for a tutorial survey on surrogate constraint methods, and see Fischetti and Lodi (2002) for an introduction to local branching.) Combining and extending these origins, SB methods provide a useful supplement to customary B&B and B&C strategies. They also give a foundation for more general forms of LB strategies, and provide a natural framework for creating a coordinated SB/LB procedure for MIP problems.

This paper is organized as follows. Section 2 gives the general structure of SB inequalities and identifies the conditions under which they provide separating inequalities of varying degrees of restrictiveness. Section 3 identifies the connection between SB and LB inequalities, and the relation between alternative strategies based on these inequalities. Section 4 then introduces a series of hypotheses about the nature of methods that are likely to prove effective for solving MIPs, accompanied by general strategies for testing and exploiting these hypotheses through the application of SB methods.

2. Fundamental Surrogate Branching Structures

The surrogate branching methods derive from the creation of surrogate constraints that provide a form of compound branching structure, composed of a union of ordinary branching inequalities, bounds on variables, and the strictly redundant inequality $0 \le 1$. Notationally, we represent these components as follows. Let $x_j = x_j'$, $j \in N$, denote the values assigned to the integer variables by a solution to an LP relaxation of MIP (where MIP may constitute a sub-problem of the original problem at a current node of a B&B tree). Define d_j and u_j respectively to be the customary "down" and "up" adjacent integer values that bracket x_j', i.e., $d_j \le x_j' \le u_j$ where $u_j = d_j + 1$. Then for selected (disjoint) subsets UP and DN of N, we may refer to a corresponding collection of branching possibilities denoted by

$$x_j \le d_j \quad j \in DN \tag{1}$$
$$x_j \ge u_j \quad j \in UP \tag{2}$$

We allow DN and UP to be drawn from chosen subsets N(F) and N(I) of N over which x_j' is respectively fractional and integer-valued (hence $d_j < x_j' < u_j$ for $j \in N(F)$ and $x_j' = d_j$ or $x_j' = u_j$ for $j \in N(I)$). Although customary B&B methods do not include consideration of variables for branching that are integer-valued, situations arise in the creation of compound surrogate constraint branching inequalities where it is important to include such variables. The reason is that the enforcement of SB inequalities over fractional variables may move a variable from N(I) to N(F)[1] and yet we wish to impose a new *revised* SB inequality relative to the LP relaxation in which the variable still belonged to N(I). This strategy of creating revised branching inequalities is one of the important features of SB methods.

In addition to the branching inequalities represented by (1) and (2), we consider the component inequalities identified by reference to upper and lower bounds on the problem variables, over chosen subsets of N denoted by NU and N0, respectively, giving rise to

$$x_j \le U_j \qquad j \in NU \tag{3}$$
$$x_j \ge 0 \qquad j \in N0 \tag{4}$$

We write each of these inequalities in $\le$ form (hence (2) becomes $-x_j \le -u_j$ and (4) becomes $-x_j \le 0$), and give each a non-negative weight w_j, together with giving a non-negative weight of w_o to the strictly redundant inequality $0 \le 1$, to yield the surrogate constraint

$$\sum(w_jx_j: j \in DN) - \sum(w_jx_j: j \in UP) + \sum(w_jx_j: j \in NU) - \sum(w_jx_j: j \in N0) \le z \tag{5}$$

where

$$z = \sum(w_jd_j: j \in DN) - \sum(w_ju_j: j \in UP) + \sum(w_jU_j: j \in NU) + w_o \tag{6}$$

The inequality (5) is the one we call the surrogate branching (SB) inequality.

Within the B&B and B&C contexts we take the sets NU and N0 to be subsets of the variables for which $x_j' = U_j$ and $x_j' = 0$, respectively, and since the other sets are selected rather than arbitrary, we will stipulate that all w_j weights are positive, except for w_o which may be 0. Hence, upon plugging the LP solution $x = x'$ into (5), we see that the resulting left hand side, which we denote by z_o, is strictly greater than z for $w_o = 0$, and hence (5) is a separating inequality in this case. In general, (5) remains a separating inequality for all values of w_o in the range

$$0 \le w_o < z - z_o$$

and w_o thus provides a parameterization of the SB inequality that determines its degree of restrictiveness.

The motivation for creating the surrogate branching inequality is related to the reason for creating surrogate constraints generally: a weighted linear combination of constraints can contain information that is not captured by any of the component constraints individually. In the case of methods for MIP problems, it can be exceedingly important to be able to derive implications from more than a single branching decision at a time. The customary B&B and B&C approaches, which only make the "incremental changes" of introducing a single branching inequality on any iteration, suffer from an

[1] As here, we often refer to variables and their indexes interchangeably.

inability to anticipate consequences of the combined effect of introducing multiple inequalities. Yet, except in rare and fortunate cases, the only way the MIP problem can be solved is by introducing multiple branches, and a lack of a prior appreciation of their mutual interdependencies – a lack of accounting for combined information that may be harbored within a surrogate constraint that embodies such branches – imposes a severe limitation on the decision process.

At the same time, the fact that a surrogate constraint offers a relaxation means that its implications are not as rigid as those of conjunction of its components. Hence, some of the branching inequalities subsumed by the surrogate constraint may be invalid in the sense of rendering all optimal MIP solutions infeasible, and yet the surrogate constraint relaxation offers a degree of *forgiveness* by which the SB inequality may nevertheless be valid. This degree of forgiveness is magnified by including the strictly redundant constraint $0 \leq 1$ as a component of the SB inequality, i.e., by the inclusion of the w_o term. Finally, the special motivation we emphasize in this paper, of introducing an inequality whose implications can be monitored and therefore whose structure can be revised and improved, leads to additional strategic possibilities beyond the realm of ordinary B&B and B&C methods. The following sections elaborate the reasons for creating the SB inequality and detail specific strategies for applying it.

3. Connections with Local Branching Inequalities

As a further foundation for seeing the implications of SB inequalities, we point out their connections to the Local Branching (LB) inequalities of Fiscetti and Lodi (2002), which are introduced in the context of providing a primal feasible strategy for 0-1 integer programming problems. In the setting, the LB inequalities have the goal of constraining the admissible solutions to lie within a region that is relatively close to a known feasible solution x*. Specifically, defining $N1 = \{ j \in N: x_j^* = 1\}$ and $N0 = \{ j \in N: x_j^* = 0\}$, then it follows that the 0-1 feasible solutions within a Hamming distance of k from the solution x* will satisfy the inequality

$$\sum(x_j: j \in N0) - \sum(x_j: j \in N1) \leq k - |N1| \qquad (7)$$

which Fiscetti and Lodi call an LB inequality. Thus, the motive is to impose the inequality (7) for an appropriately chosen value of k, and then allow a standard IP solution procedure (such as a B&B or B&C method, or even a metaheuristic method) to explore the region thus produced.[2]

Structurally, we see that (7) is in fact a special instance of an SB inequality, i.e., it arises by setting UP = N1, DN = N0 and by assigning weights of 1 to each of the component branching inequalities $x_j \leq 0$ and $x_j \geq 1$.

The SB framework enlarges the scope of the LB inequalities strategically as well as structurally, in the following important ways:

[2] In this respect, the LB strategy is an instance of a *referent domain strategy* as described in Glover and Laguna (1997).

(i) SB inequalities give a foundation for starting from integer-infeasible solutions and introducing *exploratory separating inequalities*[3] which progressively restrict the feasible space.
(ii) The solution of the LP and MIP sub-problems generated by the exploratory SB inequalities provides information for modifying and improving these inequalities. Such information derives from solution values to the sub-problems that violate component branches of the SB inequality and from reduced cost and penalty information that discloses the attractiveness of alternatives to the component branches.
(iii) The process of solving the indicated sub-problems generated by the SB inequalities simultaneously yields trial solutions that are candidates for the best solution to the original MIP problem, as in the application of surrogate constraint processes generally.

Candidate solutions are also generated with LB inequalities, but based on sub-problems derived by reference to previously identified feasible MIP solutions. The approaches of basing SB inequalities on feasible MIP solutions and of applying them to yield progressively refined exploratory inequalities are usefully complementary, and provide the basis for interweaving the two approaches in a mutually reinforcing pattern.[4]

Implications and specific strategies for taking advantage of these features of the SB framework are examined in the next section.

4. Hypotheses and Associated Strategies.

We begin by observing that an SB inequality can be progressively strengthened as follows:

(1) removing a redundant component branch,
(2) adding a binding component branch,
(3) increasing the weight w_j on a binding component branch,
(4) modifying a redundant branch (by reversal or changing the constant term) to make it binding,
(5) decreasing the value of w_o.

In this paper we focus on processes that make use of such progressive strengthening, together with associated processes that modify component branches by taking advantage of sub-problem information. (By extension of this focus, it is also possible to alternate strengthening steps with weakening steps using strategic oscillation procedures, or by the application of tabu search procedures more generally.)

[3] Such inequalities can also be conceived as instances of pseudo-cuts, as proposed in conjunction with tabu search. Likewise, the memory mechanisms of tabu search can be used to implement strategies for SB inequalities, as discussed in Section 4.
[4] We explore such "interwoven approaches" in a sequel.

Hypothesis 1. (Motivation for Progressive Strengthening) A process that starts by enforcing a weak partial branching inequality and then enforces progressively stronger instances of the inequality discloses patterns that can be exploited to yield better decisions.

It is important to note that progressive strengthening of an SB inequality can be conveniently carried out by LP post-optimization, thereby giving added weight to the motivation suggested by Hypothesis 1. We next examine two main ways to identify useful patterns of the type alluded to in this hypothesis.

4.1 Persistent and Emergent Attractiveness.

The present context motivates an adaptation of the persistent attractiveness and emergent attractiveness notions as introduced in connection with Tabu Search (TS). The following definitions will first be stated very loosely and then will be made more precise.

Persistent Attractiveness: A certain branching direction for a variable x_j is attractive throughout a critical region of solution alternatives as the MIP problem is progressively modified.

Emergent Attractiveness: A certain branching direction for a variable x_j becomes increasingly more attractive while moving through a critical region of solution alternatives as the MIP problem is progressively modified. (Initially, the branching direction can be unattractive, and then gradually alter its status.)

A brief discussion may help to bring these comments into focus. The "critical region of solution alternatives" in the present context refers to the progressively changed solution space that results by the five operations previously mentioned for strengthening an SB inequality, together with the operation of reversing a component branch direction based on solution information that discloses this to be preferable. We recall two classical definitions.

Shadow Prices: The shadow prices for a Linear Programming problem are the solutions to its dual. The ith shadow price is the change in the objective function resulting from a one unit increase in the ith coordinate of b. A shadow price is also the amount that an investor would have to pay for one unit of a resource in order to buy out the manufacturer.

Penalty Costs: Penalty Costs are the amounts the optimal value of the objective function would change for each unit increase in the non-basic variables. They are the negatives of the non-basic bottom row entries in the final tableau.

A natural means to measure the relative attractiveness of alternative branches on a given iteration, in order to provide a foundation for measuring persistent and emergent attractiveness, is provided by a standard penalty cost evaluation (or a "pseudo-cost" evaluation) as applied in B&B. As a simple example, if a particular branch direction originally evaluated as attractive continues to receive a favorable evaluation as w_o is progressively increased, then this direction of change qualifies as persistently attractive. (If the variable eventually receives a value in the solution to the relaxed LP problem that satisfies a particular branching direction, then this can be considered as an evaluation in favor of that direction.)

On the other hand, the case where a component branching direction is inappropriately selected, and therefore should be modified, can be identified by the emergent attractiveness of the counter branching direction. For example, as w_o is progressively decreased toward 0, if the counter branching direction for a variable x_j that originally was unattractive begins to look attractive (as evidenced either by customary branching penalties or by a situation where x_j receives a value that satisfies the counter direction), then this is a signal that the current branch direction may be unsuitably chosen. Thus the emergent attractiveness of a counter branching direction provides a way to amend the SB inequality.

Illustrative Measures.

We amplify the foregoing comments and make them more specific as follows. Assume a standard measure of branching attractiveness (e.g., as derived from a penalty cost) is applied for a selected subset of w_o values, which we denote by $w_o \in R$. We then can define a "preference value" PrefValue(w_o,j) that discloses the relative desirability of an UP branch compared to a DN branch for each x_j and for each $w_o \in R$. By convention, suppose

$$\text{PrefValue}(w_o,j) = \begin{array}{l} + \text{ if UP is preferred to DN} \\ 0 \text{ if there is no preference} \\ - \text{ if DN is preferred to UP} \end{array}$$

where the magnitude of PrefValue(w_o,j) identifies the degree of attractiveness of the preferred branch. It is also useful to consider a related "preference frequency" PrefFrequency(w_o,j), which starts at 0 and is incremented by an amount $\Delta(w_o,j)$ where

$$\Delta(w_o,j) = \begin{array}{l} 1 \text{ if UP is preferred to DN} \\ 0 \text{ if there is no preference} \\ -1 \text{ if DN is preferred to UP.} \end{array}$$

Then we can specify that a persistent attractiveness measure PA(j) for a variable x_j is a function of the two quantities

$$\sum(\text{PrefValue}(w_o,j): w_o \in R)$$

and

$$\sum(\text{PrefFrequency}(w_o,j): w_o \in R)$$

while an emergent attractiveness measure EA(j) for x_j is a function of the two quantities

$$\sum(F(u)\text{PrefValue}(w_o,j): w_o \in R)$$

and

$$\sum(F(u)\text{PrefFrequency}(w_o,j): w_o \in R)$$

where F is a strictly monotone increasing function, i.e., $F(u'') > F(u')$ for $u'' > u' \in R$. (For example, F can implicitly be determined by an exponential smoothing operation.) The underlying functions for determining PA(j) and EA(j) can differ from each other and can be based on thresholds.

Define

$$\delta(j) = \begin{cases} 1 & \text{if } j \in UP \\ -1 & \text{if } j \in DN \end{cases}$$

and let

$$PA^*(j) = \delta(j)PA(j)$$
$$EA^*(j) = \delta(j)EA(j)$$

Then we may hypothesize that as PA*(j) and EA*(j) become larger, the likelihood increases that the presumptive branch for x_j is appropriate. In particular, for a given variable x_p, if the value EA*(p) is negative, or small relative to other EA*(j) values, then x_p may be considered a candidate to *reverse direction*, i.e., to change its presumptive branching direction. This leads to the following speculation.

Hypothesis 2. The SB inequality proves to be increasingly effective as an aid to solving the MIP problem as the number of variables that are candidates to reverse direction is reduced.

This hypothesis evidently depends on the specific rules used to define PA(j) and EA(j). The precise forms of these rules is a topic for research, which over time may be anticipated to provide successive generations of improved methods, beginning with a "first generation" method that uses very simple rules.

The following procedure has the goal of creating an SB inequality that minimizes the number of candidates to reverse direction. Let J denote the union of the sets UP and DN.

Procedure to Exploit Hypothesis 2

Step 0: Identify an initial choice of component branching directions to determine the SB inequality. Solve the LP problems over w_o in R by post-optimization to generate the measures to produce PA*(j) and EA*(j), $j \in J$.

Step 1: Choose one or more x_j variables that are candidates to reverse direction. If none exist, stop. Otherwise, implement the change of direction for these variables (by redefining UP and DN.)

Step 2: Again apply the LP post-optimization process to determine new values of PA*(j) and EA*(j), for $j \in J$. If the number of candidates to reverse direction does not decrease, stop. Otherwise, return to Step 1.

The preceding approach is a simple "hill climbing" procedure, which takes advantage of information produced by the progressive post-optimization process and the concepts embodied in the definitions of PA(j) and EA(j). It may be observed that the process of progressively increasing w_o may result in a situation where no feasible LP solution exists, which thereby also provides information about the identity of inappropriately selected branching directions. Such an outcome yields a limiting value for w_o applicable to the current iteration of the procedure, and in general, a situation where the relative attractiveness of several branches abruptly changes may be taken to indicate that further decreases in w_o are unwarranted – until the composition of UP and DN is modified. (The number of candidates simultaneously permitted to reverse direction on each execution of Step 1 will have a bearing on the outcome.)

The following provides a possible extension of the process.

Extended Procedure

Step 1: When the foregoing procedure stops, if any candidates to reverse direction remain, remove all of these from J.

Step 2: Re-apply the procedure relative to this smaller J, while keeping track of attractiveness measures for variables x_j for j not in J. If any of these variables now has a branching direction that is strongly supported by the successive post-optimization steps, add this variable back to J and repeat until the addition of such a variable creates a candidate to reverse direction.

The outcomes resulting from the foregoing procedures may additionally be used to design branching rules for B&B.

4.2 Sequence Independent Decisions.

An important alternative to the foregoing development is provided by the perspective that underlies tabu branching. The key idea can be expressed as follows.

Hypothesis 3. The choices of a sequential decision process that creates an implicit tree construction can be improved by replacing the sequential order with the less restrictive conditions that result from using tabu search memory and aspiration criteria.

The hypothesis derives in part from the observation that the B&B type of sequential decision process, as considered in previous sections, is based on incomplete information, which is limited to the implications of branching decisions that precede a branch currently under consideration. Although the generation of new sequences in the exploitation of Hypothesis 2 takes advantage of broader implications, the process still retains a degree of "sequential myopia."

The approach of tabu branching, by contrast, suggests the usefulness of changing branching decisions by drawing on the implications available from a more complete set of branches. Thus, an earlier branching decision can be re-evaluated independently of the order in which it was made, by taking account of subsequent branching decisions. In this expanded system of interdependencies, the implications of the earlier decisions are established in the context of additional decisions that followed. The enlarged context is the one in which tabu branching operates. Such an approach can be implemented by changing branches until a selected evaluation criterion signals a local optimum is reached, and then by activating memory mechanisms of tabu search to go beyond local optimality. Alternately, the process can be applied without TS memory and merged with the approach for exploiting Hypothesis 2 to give a multi-start variant.

In applying a non-sequential approach to a set of branches that were first generated sequentially, it is appropriate to note that certain branches made in the initial construction may compel the inclusion of certain other branches that followed them. This knowledge can be used to focus on critical choices in subsequent exploitation of Hypothesis 3. In particular, there is no merit in changing a forced branch until an unforced branch that preceded it is changed. At that point, all forced branches that came later in the sequence lose their forced status. (There can be 'ripple effects,' as manifested in a situation where the change of a given branch induces a change of several others.)

Alternative implementations of such processes provide a rich source of empirical explorations.

References.

Fischetti, M. and A. Lodi (2002). "Local Branching," Research Report, DEI, University of Padova and DEIS, University of Bologna.

Glover, F. (2002). "Tutorial on Surrogate Constraint Approaches for Optimization in Graphs," to appear in *Journal of Heuristics*.

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