Formulating Good MILP Models

J. N. Hooker

February 7, 2010

Abstract

Writing good models is essential to harnessing MILP solution technology. Due to an underlying theorem that characterizes MILP representability, MILP modeling involves identifying two types of structure in a problem: knapsack constraints and choices between discrete alternatives. This article illustrates the modeling process with a series of examples. They include capital budgeting, freight transport, set packing/covering/partitioning, airline crew rostering, facility location, fixed charges, lot sizing, and package delivery. Piecewise linear modeling, logical conditions, symmetry, special ordered sets, indicator constraints, and semi-continuous variables are also discussed.

Mixed integer/linear programming (MILP) provides a highly developed technology for solving a wide variety of optimization problems with continuous and discrete variables. To harness this technology, however, one must know how to formulate a given problem as an MILP model. In many cases, it is not enough simply to write a correct model. One must also write a "good" model, meaning that the formulation is amenable to fast solution. A poor formulation can require orders of magnitude more solution time than a good one.

MILP problem formulation is generally regarded as an art rather than a science, but it need not be unprincipled. An underlying theory tells us precisely which problems can be given an MILP formulation, and this in turn leads to useful guidelines for writing good models.

The guidelines are based on the fact that MILP formulations involve two modeling ideas: knapsack constraints and choices between discrete alternatives (disjunctions). By identifying these two elements in a given problem, one can obtain practical MILP formulations in a reasonably systematic way. The resulting models incorporate many of the "tricks" that one might otherwise learn only through familiarity with the folklore of modeling.

The characteristics of a good model have never been precisely defined, but in many cases it is useful to have a "tight" continuous relaxation. The relaxation is important because its solution provides a bound on the optimal value of the original problem. A bound from a tight relaxation is closer to the optimal value and can accelerate solution, because all general-purpose MILP solvers make essential use of bounds to prune the search tree.

1 Basic Definitions

An MILP problem consists of linear inequality constraints and a linear objective function. At least some of the variables must take integer values, while the remaining variables are continuous. Such a problem can be written in the form

$$\begin{array}{ll} \min \ (\text{or max})\ cx & \\ Ax \geq b & (a) \\ x \in \mathbb{R}^n, \ \ x_j \in \mathbb{Z} \text{ for } j \in J & (b) \end{array} \tag{1}$$

where $x = (x_1, \ldots, x_n)$ and A is an $m \times n$ matrix. Variables x_j for $j \in J$ must take integer values. A value of x is *feasible* for (1) if it satisfies the constraints (a) and (b), and the *feasible set* is the set of feasible values. The objective is to find a feasible x that minimizes (or maximizes) cx.

The *continuous relaxation* of (1) is obtained by dropping the integrality condition on x_j for $j \in J$. The continuous relaxation is generally much easier to solve than (1), because it is a linear programming problem. Its optimal value is a lower bound on the optimal value of (1) if the objective is to minimize, and an upper bound if the objective is to maximize.

2 Knapsack Modeling

Mixed integer formulations frequently involve counting ideas that can be expressed as *knapsack inequalities*. For present purposes we can define a knapsack inequality to be one of the form $\sum_j a_j x_j \leq \beta$, or $ax \leq \beta$ for short, where some (or all) of the variables x_j may be restricted to integer values. We also regard $ax \geq \beta$ as a knapsack inequality.

2.1 Knapsack Problems

The term "knapsack inequality" derives from the fact that the *integer knapsack problem* can be formulated with such an inequality. The problem is to pack a knapsack with items that have the greatest possible value while not exceeding a maximum weight β. There are n types of items. Each item of type j has weight a_j and adds value c_j. If x_j is the number of items of type j put into the knapsack, the problem can be written

$$\begin{aligned} &\max\ cx \\ &ax \leq \beta \\ &x_j \in \mathbb{Z},\ \text{all } j \end{aligned} \tag{2}$$

This is an MILP model because it has the form (1).

A wide variety of modeling situations involve this same basic idea. A classic example is the capital budgeting problem, in which the objective is to allocate a limited amount of capital to projects so as to maximize revenue. Here β is the amount of capital available. There are n types of projects, and each project j has initial cost a_j and earns revenue c_j. Variable x_j represents the number of projects of type j that are funded.

Typically an MILP model contains many knapsack constraints. There may also be purely linear constraints in which all the variables are continuous. This article focuses on modeling ideas that require integer variables, while the article on Linear Programming Models discusses purely linear modeling.

2.2 Example: Freight Transport

A product can be manufactured at several plants, from which it can be shipped to n customers (Fig. 1). Each customer j requires a quantity D_j of the product, and each plant i can manufacture at most C_i. The product must be transported in vehicles, which have capacity K. The cost of sending a vehicle along the route from factory i and to customer j is c_{ij}, regardless of how much cargo it carries. The objective is to assign vehicles to routes to minimize cost while meeting customer demand.

The movement of product from the plants to customers is described by a standard network flow model, which is purely linear. If x_{ij} is the quantity shipped from plant i to customer j, the constraints are

$$\begin{aligned} &\sum_j x_{ij} \leq C_i,\ \text{all } i \quad (a) \\ &\sum_i x_{ij} = D_j,\ \text{all } j \quad (b) \end{aligned} \tag{3}$$

Constraints (a) ensure that plant capacity is not exceeded, and constraints (b) meet customer demand. The vehicles assigned to each route must have

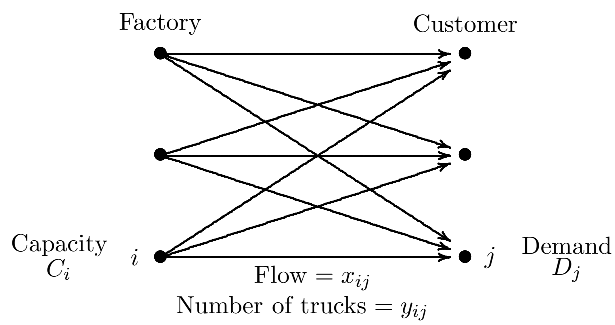

Figure 1: *A freight transport problem.*

enough capacity to carry the flow x_{ij} along that route. If y_{ij} is the number of vehicles assigned to the route from i to j, this condition is captured by the knapsack inequalities

$$Ky_{ij} \geq x_{ij}, \quad \text{all } i, j \tag{4}$$

where y_{ij} takes integer values. These are knapsack inequalities because they can be written $x_{ij} - Ky_{ij} \leq 0$. The objective is to minimize cost:

$$\min \sum_{ij} c_{ij} y_{ij} \tag{5}$$

The MILP model consists of (3)–(5), plus the condition that $x_{ij} \geq 0$ and $y_{ij} \in \mathbb{Z}$ for all i, j.

A variation of this problem allows several types of vehicles to be used. A vehicle of type t has capacity K_t and costs c_{ijt} to operate on the route from i to j. Then if y_{ijt} is the number of vehicles of type t traveling from i to j, the knapsack constraint (4) becomes

$$\sum_t K_t y_{ijt} \geq x_{ij}, \quad \text{all } i, j$$

and the objective (5) becomes

$$\min \sum_{ijt} c_{ijt} y_{ijt}$$

2.3 Set Packing, Set Covering, and Set Partitioning

Special cases of knapsack inequalities occur in set packing, set covering, and set partitioning problems. The *set packing* problem begins with a collection of finite sets S_j for $j = 1, \ldots, n$ that may partially overlap. It seeks a largest subcollection of sets that are pairwise disjoint.

Suppose, for example, that there are n surgeries to be performed, and the objective is to perform as many as possible this morning. Surgery j requires a specific set S_j of surgeons and other personnel. Because the surgeries must proceed in parallel, no two surgeries with overlapping personnel can be performed. This is a set packing problem.

The set packing problem can be formulated with 0-1 knapsack inequalities. Let $A_{ij} = 1$ when item i belongs to set S_j, and $A_{ij} = 0$ otherwise. Let variable $x_j = 1$ when set j is selected. The knapsack inequality $\sum_{j=1}^{n} A_{ij} x_j \leq 1$ prevents the selection of more than one set containing item i. Thus the system $Ax \leq e$ of knapsack inequalities, where e is a vector of ones, prevents the selection of any two overlapping sets. The objective is to maximize $\sum_{j=1}^{n} x_j$ subject to $Ax \leq e$ and $x \in \{0,1\}^n$, which is an MILP problem.

The *set covering* problem likewise begins with a collection of sets S_j but seeks the minimum subcollection that contains all the elements in the union of the sets. For example, one may wish to buy a minimum collection of songbooks that contains all the songs that appear in at least one book. Here S_j is the set of songs in book j.

If A_{ij} and x_j are as before, the knapsack inequality $\sum_{j=1}^{n} A_{ij} x_j \geq 1$ ensures that item i is covered. The set covering problem is to minimize $\sum_{j=1}^{n} x_j$ subject to $Ax \geq e$ and $x \in \{0,1\}^n$. The objective function in this or the set packing problem can be generalized to cx by attaching a weight c_j to each set S_j, to represent the cost or benefit of selecting S_j.

The *set partitioning* problem seeks a subcollection of sets such that each element is contained in exactly one of the sets selected. The constraints are therefore $Ax = e$, which are a combination of the knapsack constraints $Ax \leq e$ and $Ax \geq e$. The problem is to minimize or maximize cx subject to these constraints.

2.4 Example: Airline Crew Rostering

An important practical example of set partitioning is the airline crew rostering problem. Crews must be assigned to sequences of flight legs while observing complicated work rules. For example, there are restrictions on the number of flight legs a crew may staff in one assignment, the total duration of the assignment, the layover time between flight legs, and the locations of the origin and destination.

Let S_j be a set of flight legs that can be assigned to a single crew, where j

indexes all possible such sets. There may be millions of sets S_j and therefore millions of variables x_j. A set S_j is selected ($x_j = 1$) when it is assigned to a crew, incurring cost c_j. This is a partitioning problem because each flight leg must be staffed by exactly one crew and must therefore appear in exactly one selected S_j.

There are ways to formulate the problem with a much smaller model that explicitly represents the work rules with constraints. Yet the set partitioning model has the advantage that it can be solved by *branch-and-price* methods. These methods solve the continuous relaxation by *column generation*; that is, by adding variables (and the associated columns of A) to the problem only when they can improve the solution. Typically, only a tiny fraction of the columns are generated. This is an instance in which a large model is "better" than a small one in the sense that it permits faster solution.

2.5 Clique Inequalities

A collection of set packing constraints in a model can sometimes be replaced or supplemented by a *clique inequality*, which substantially tightens the relaxation. For example, the 0-1 inequalities

$$\begin{aligned} x_1 + x_2 \phantom{{}+x_3} &\leq 1 \\ x_1 \phantom{{}+x_2} + x_3 &\leq 1 \\ x_2 + x_3 &\leq 1 \end{aligned} \tag{6}$$

are equivalent to the clique inequality $x_1 + x_2 + x_3 \leq 1$. One can see that the clique inequality provides a tighter relaxation from the fact that $x_1 = x_2 = x_3 = 1/2$ violates it but satisfies (6). It should therefore replace (6) in the model.

To generalize this idea, define a graph whose vertices j correspond to 0-1 variables x_j. The graph contains an edge (i, j) whenever $x_j + x_j \leq 1$ is implied by a constraint in the model. If the induced subgraph on some subset C of vertices is a clique, then $\sum_{j \in C} \leq 1$ is a valid inequality and can be added to the model.

2.6 Logical Conditions

Logical conditions on 0-1 variables can be formulated as knapsack inequalities that are similar to set covering constraints. Suppose, for example, that either plants 2 and 3 must be built, or else plant 1 must not be built. For the moment, regard x_j as a boolean variable that is true when plant j is built, and false otherwise. The condition can be written

$$\neg x_1 \vee x_2 \vee x_3 \tag{7}$$

where $\vee$ means "or" and $\neg$ means "not." Such a condition is a *logical clause*, meaning that it is a disjunction of *literals* (boolean variables or their negations). Because the clause states that at least one of the literals must be true, it can be written as an inequality $(1 - x_1) + x_2 + x_3 \geq 1$ by viewing x_j as true when $x_j = 1$ and false when $x_j = 0$. This is equivalent to the knapsack inequality $-x_1 + x_2 + x_3 \geq 0$.

Any logical condition built from "and," "or," "not," and "if" can be converted to a set of clauses and given an MILP model on that basis. One need only observe three equivalences.

- $A \Rightarrow B$ (i.e., "A implies B," or "B if A") is equivalent to $\neg A \vee B$.
- $\neg(A \wedge B)$ is equivalent to $\neg A \vee \neg B$ (De Morgan's Law), where $\wedge$ means "and."
- $A \vee (B \wedge C)$ is equivalent to $(A \vee B) \wedge (A \vee C)$, a form of distribution.

Consider, for example, the condition, "If plants 1 and 2 are built, then plants 3 and 4 must be built." It can be written

$$(x_1 \wedge x_2) \Rightarrow (x_3 \wedge x_4)$$

The implication is first eliminated to obtain $\neg(x_1 \wedge x_2) \vee (x_3 \wedge x_4)$. The negation is then brought inside using De Morgan's Law, resulting in the expression $\neg x_1 \vee \neg x_2 \vee (x_3 \wedge x_4)$. The conjunction is now distributed:

$$(\neg x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_4)$$

This conjunction of two clauses can be written as two knapsack constraints:

$$\begin{aligned} -x_1 - x_2 + x_3 &\geq -1 \\ -x_1 - x_2 + x_4 &\geq -1 \end{aligned}$$

A model can sometimes be further tightened by generating *resolvents* of the clauses. Suppose, for example, that a model contains clauses $x_1 \vee x_2$, $\neg x_1 \vee \neg x_3$. Because exactly one variable (x_1) occurs in the two clauses with different signs, one can infer the resolvent $x_2 \vee \neg x_3$, which contains all the literals in either clause except x_1 and $\neg x_1$. The resolvent is converted to knapsack form and added to the MILP model. The process can be repeated as desired, and clauses implied by others (if any) can be dropped.

3 MILP Representability

Knapsack-based modeling is useful but harnesses only a fraction of MILP modeling capability. The full resources of MILP modeling appear when one introduces auxiliary 0-1 variables to represent discrete alternatives. A choice among alternatives can be represented by a disjunction in which each term describes one of the alternatives. If each term is a system of knapsack inequalities, the problem has an MILP model—provided the systems satisfy a fairly innocuous technical condition. In fact, the problem has an MILP model *if and only if* it can be written as such a disjunction of knapsack systems.

3.1 Example: Fixed-Charge Function

A simple fixed charge problem illustrates disjunctive modeling. Suppose the cost x_2 of manufacturing quantity x_1 of some product is to be minimized. The cost is zero when $x_1 = 0$ and is $f + cx_1$ otherwise, where f is the fixed cost and c the unit variable cost.

The problem can be viewed as minimizing x_2 subject to $(x_1, x_2) \in S$, where S is the set depicted in Figure 2(a). There are two discrete alternatives: the cost is zero, or the cost is a fixed cost plus a variable cost. They are captured by the disjunctive formulation

$$\begin{aligned} &\min\ x_2 \\ &\begin{pmatrix} x_2 \geq 0 \\ x_1 = 0 \end{pmatrix} \vee \begin{pmatrix} x_2 \geq cx_1 + f \\ x_1 \geq 0 \end{pmatrix} \end{aligned} \tag{8}$$

The disjuncts respectively define two polyhedra P_1 and P_2, illustrated in the figure. In general there would be additional constraints in the problem, but the focus here is on the fixed-charge formulation.

It is impossible to write an MILP model for the disjunction in (8) because it fails the technical condition mentioned earlier. The condition is that the polyhedra defined by the disjunctions (ignoring the integrality condition) must have the same *recession cone*. The recession cone is the set of directions in which one can go forever without leaving the polyhedron. That is, the recession cone of a polyhedron P is the set of vectors r such that for any $x \in P$, $x + \alpha r \in P$ for all $\alpha \geq 0$. If the polyhedron is bounded, its recession cone is the origin.

The recession cone of P_1 in the example is P_1 itself (see Fig. 2a), and the recession cone of P_2 is the set of all vectors (x_1, x_2) with $x_2 \geq cx_1 \geq 0$.

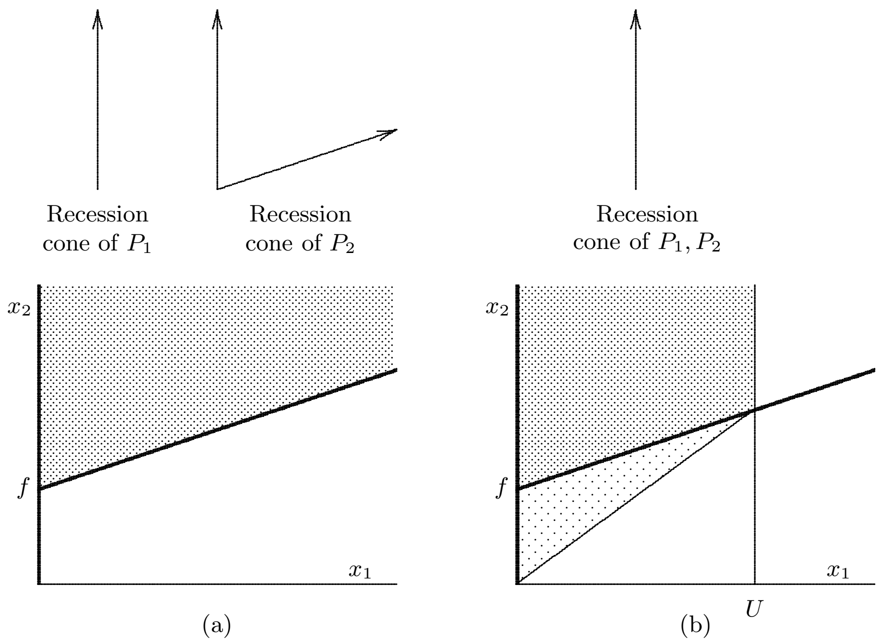

Figure 2: *(a) Feasible set of a fixed-charge problem, consisting of the union of polyhedra P_1 (heavy vertical line) and P_2 (shaded area). (b) Feasible set $P_1 \cup P_2'$ of the same problem with the bound $x_1 \leq U$, where P_2' is the darker shaded area. The convex hull of the feasible set is the entire shaded area.*

However, if an upper bound U is placed on x_1, the problem becomes

$$\begin{array}{l} \min\ x_2 \\ \begin{pmatrix} x_1 = 0 \\ x_2 \geq 0 \end{pmatrix} \vee \begin{pmatrix} x_2 \geq cx_1 + f \\ 0 \leq x_1 \leq U \end{pmatrix} \end{array} \tag{9}$$

The recession cone of each of the resulting polyhedra P_1, P_2' (Figure 2b) is the same (namely, P_1), and the problem is therefore MILP representable. In fact, it will be shown that the problem has the following MILP model:

$$\begin{array}{l} \min\ x_2 \\ 0 \leq x_1 \leq U\delta,\ \ x_2 \geq f\delta + cx_1,\ \ \delta \in \{0,1\} \end{array} \tag{10}$$

which can be simplified to

$$\begin{array}{l} \min\ f\delta + cx_1 \\ 0 \leq x_1 \leq U\delta,\ \ \delta \in \{0,1\} \end{array} \tag{11}$$

The 0-1 variable δ encodes whether the quantity produced is zero or positive, in the former case ($\delta = 0$) forcing $x_1 = 0$, and in the latter case incurring the fixed charge f.

3.2 The Representability Theorem

A problem is *MILP representable* if there is some MILP model for it. To make this more precise, suppose one wishes to model an optimization problem $\mathcal{O}$, which can in general be conceived as the problem of minimizing some variable x_j subject to $x \in S$, where $x = (x_1, \ldots, x_n)$. Any objective function $f(x)$ can be accommodated by adding variable x_j and constraint $x_j \geq f(x)$ to S, and minimizing or maximizing x_j.

Consider an MILP model $\mathcal{M}$ that minimizes x_j subject to constraints that contain the variables $x_1, \ldots, x_n$ (some of which may be restricted to integers), possibly along with auxiliary continuous variables $y_1, \ldots, y_m$ and 0-1 variables $\delta_1, \ldots, \delta_k$. Model $\mathcal{M}$ *represents* problem $\mathcal{O}$ if the projection of $\mathcal{M}$'s feasible set onto x is S; that is, $x \in S$ if and only if (x, y, δ) is feasible in $\mathcal{M}$ for some y, δ. This means that one can solve $\mathcal{O}$ by solving $\mathcal{M}$, because if (x, y, δ) is an optimal solution of $\mathcal{M}$, then x is an optimal solution of $\mathcal{O}$.

MILP representability is completely characterized by the following theorem, which is proved in [4, 6]. (Technically, the proof assumes that A, c consist of rational data.)

Theorem 1 *An optimization problem that minimizes x_j subject to $x \in S$ is MILP representable if and only if S is the feasible set of some constraint of the form*

$$\bigvee_{k \in K} \left(A^k x \geq b^k \right), \; x \in \mathbb{R}^n, \; x_j \in \mathbb{Z} \text{ for } j \in J \tag{12}$$

where K is finite and the polyhedra $\{x \mid A^k x \geq b^k\}$ for $k \in K$ have the same recession cone.

Thus an optimization problem has an MILP model if and only if its feasible set can be described by a disjunction of knapsack systems that define polyhedra with the same recession cone.

If it is difficult to identify the recession cones, one can always ensure they are the same by placing fixed lower and upper bounds on all the variables—bounds wide enough so as not to be violated by any reasonable solution. This results in bounded polyhedra for every disjunct, so that every recession cone is the origin.

3.3 Conversion to an MILP Model

There remains the question as to how to convert the disjunctive problem (12) into an MILP model. There are two standard ways: a big-M formulation and a convex hull formulation. Both require 0-1 auxiliary variables. The convex hull formulation is generally tighter, but at the cost of introducing continuous auxiliary variables as well. First, the big-M formulation.

Theorem 2 *If an optimization problem satisfies the conditions of Theorem 1, it is represented by the big-M formulation*

$$\begin{aligned} &\min\ cx \\ &A^k x \geq b^k - M^k(1-\delta_k),\quad k \in K \\ &x \in \mathbb{R}^n,\ \ x_j \in \mathbb{Z}\ \textit{for}\ j \in J,\quad \delta_k \in \{0,1\}\ \textit{for}\ k \in K \end{aligned} \tag{13}$$

where

$$M^k = b^k - \min_{\ell \neq k}\left\{\min_x \left\{ A^k x \;\middle|\; A^\ell x \geq b^\ell,\ x \in \mathbb{R}^n,\ x_j \in \mathbb{Z}\ \textit{for}\ j \in J \right\}\right\} \tag{14}$$

When auxiliary variable $\delta_k = 1$, the kth linear system $A^k x \geq b^k$ is enforced. When $\delta_k = 0$, the kth system is deactivated by subtracting a vector M^k of large numbers from the right-hand side. The components of M^k are chosen in (14) to be as small as possible, so as to result in a tighter relaxation, but larger values yield a correct model.

The *convex hull MILP formulation* introduces continuous auxiliary variables by disaggregating x into a sum $\sum_{k \in K} x^k$, where each x^k corresponds to the kth disjunct.

Theorem 3 *If an optimization problem $\mathcal{O}$ satisfies the conditions of Theorem 1, it is represented by the formulation*

$$\begin{aligned} &\min\ cx \\ &x = \sum_{k \in K} x^k \\ &A^k x^k \geq b^k \delta_k,\quad k \in K \\ &\sum_{k \in K} \delta_k = 1 \\ &x \in \mathbb{R}^n,\ \ x_j \in \mathbb{Z}\ \textit{for}\ j \in J,\quad \delta_k \in \{0,1\}\ \textit{for}\ k \in K \end{aligned} \tag{15}$$

The convex hull formulation (15) has the advantage that it provides the tightest possible linear relaxation of $\mathcal{O}$, namely a *convex hull relaxation*. That is, its continuous relaxation describes a set that, when projected onto x, is the closure of the convex hull of $\mathcal{O}$'s feasible set. This assumes that the individual systems $A^k x \geq b^k$ provide convex hull relaxations—as they do, for example, when all the x_js are continuous.

The model (15) continues to provide a valid convex hull relaxation of $\mathcal{O}$ when the recession cone condition is not satisfied, even though (15) does not correctly represent $\mathcal{O}$ in this case. In addition, the model (15) is *locally ideal*, meaning that the 0-1 variables δ_k take integer values at every extreme point of the polyhedron described by the model's linear relaxation.

The big-M formulation (13) likewise provides a valid relaxation of $\mathcal{O}$ even when the polyhedra do not have the same recession cone, although not in general a convex hull relaxation. The big-M formulation need not be locally ideal.

It is hard to say in general when a big-M formulation is preferable to a convex hull formulation. It tends to be more attractive when there are a large number of disjuncts, because the convex hull model introduces a vector x^k of auxiliary continuous variables for every disjunct. In some cases, the big-M model provides a convex hull relaxation. On the other hand, the convex hull model may simplify, and some variables may drop out.

In practice, a problem typically has several disjunctive constraints. (A system of knapsack constraints can be regarded as a disjunction with a single disjunct.) The MILP formulations of the disjunctions are pooled to form an MILP model. Note that while convex hull formulations provide convex hull relaxations of the individual disjunctions, when taken together they do not necessarily provide a convex hull relaxation of the problem as a whole.

3.4 Modeling the Fixed-Charge Function

The fixed charge problem can be modeled by writing a big-M or a convex hull formulation of the disjunctive problem (8). In this case, the two formulations turn out to be identical.

The big-M formulation (13) of (8) is

$$\begin{array}{ll} 0 \leq x \leq M_1^1 \delta & 0 \leq x \leq U + M_1^2(1-\delta) \\ z \geq -M_2^1 \delta & z \geq cx + f - M_2^2(1-\delta) \\ \delta \in \{0,1\} & \end{array} \tag{16}$$

where δ_1, δ_2 are written as $\delta, 1-\delta$ because they sum to one. From (14), $(M_1^1, M_2^1, M_1^2, M_2^2) = (U, -f, 0, f)$. So (16) simplifies to (11).

The convex hull formulation (15) of (8) is

$$\begin{array}{ll} x = x^1 + x^2 & z = z^1 + z^2 \\ x^1 = 0 & 0 \le x^2 \le U\delta \\ z^1 \ge 0 & z^2 \ge cx^2 + f\delta \\ \delta \in \{0,1\} & \end{array}$$

Because δ_1 does not appear in the model for the first disjunct, δ_2 is written as δ. The fact that $x^1 = 0$ means that x can replace x^1. Also z^1 can be removed by writing $z \ge z^2$. Because z_2 is minimized, z can now replace z_2, and the model again simplifies to (11). The feasible set of (11)'s continuous relaxation projects onto the convex hull of the feasible set in x_1, x_2-space, as illustrated in Fig. 2(b).

4 Examples of MILP Modeling

The disjunctive modeling approach may seem a roundabout way to obtain a fairly obvious model for the fixed charge function. However, it provides a general tool for constructing good formulations that are much less obvious for more complex problems. This is illustrated by the following examples.

4.1 Capacitated Facility Location

The transportation problem described earlier assumes that factories already exist at all of the factory sites. The problem can be modified to address a location as well as a routing issue: at which sites should factories be built?

The problem poses two discrete alternatives for each site i. Either a factory is built or it is not. If it is built, the total shipments out of the location must be at most C_i, and a fixed cost f_i is incurred. Otherwise nothing is shipped out of the location. Thus if x_{ij} is again the quantity transported from i to j, the situation at site i is represented by a disjunction of two knapsack systems:

$$\left(\begin{array}{c} \displaystyle\sum_{j=1}^{n} x_{ij} \le C_i \\ 0 \le x_{ij} \le K_{ij} y_{ij}, \text{ all } j \\ z_i = f_i \\ y_{ij} \in \mathbb{Z}, \text{ all } j \end{array} \right) \vee \left(\begin{array}{c} x_{ij} = 0, \text{ all } j \\ z_i = 0 \end{array} \right) \quad (17)$$

where variable z_i represents the fixed cost incurred. Again, each customer j must receive adequate supply:

$$\sum_{i=1}^{m} x_{ij} = D_j, \quad \text{all } j \tag{18}$$

This can be viewed as a disjunction with one disjunct. The problem is to minimize

$$\sum_{i=1}^{m} \left(z_i + \sum_{j=1}^{n} c_{ij} x_{ij} \right) \tag{19}$$

subject to (18), and subject to (17) for all i.

The polyhedra defined by the two disjuncts in (17) have different recession cones. The cone for the first polyhedron is $\{(x_i, y_i) \mid x_i = 0,\ y_i \geq 0\}$ where $x_i = (x_{i1}, \ldots, x_{in})$ and $y_i = (y_{i1}, \ldots, y_{in})$, while the cone for the second is $\{(x_i, y_i) \mid x_i = 0\}$. However, if we add the innocuous constraint $y_i \geq 0$ to the second disjunct, the two disjuncts have the same recession cone and can be given a convex hull formulation:

$$\begin{aligned} & \sum_{j=1}^{n} x_{ij} \leq C_i \delta_i \\ & 0 \leq x_{ij} \leq K_{ij} y_{ij}, \ \text{all } j \\ & z_i = f_i \delta_i, \ \ \delta_i \in \{0, 1\}, \ \ y_{ij} \in \mathbb{Z}, \ \text{all } j \end{aligned} \tag{20}$$

The resulting MILP model for the entire problem is

$$\begin{aligned} & \min \ \sum_{i=1}^{m} \left(f_i \delta_i + \sum_{j=1}^{n} c_{ij} y_{ij} \right) \\ & \sum_{j=1}^{n} x_{ij} \leq C_i \delta_i, \ \text{all } i \\ & 0 \leq x_{ij} \leq K_{ij} y_{ij}, \ \text{all } i, j \\ & \sum_{i=1}^{m} x_{ij} = D_j, \ \text{all } j \\ & \delta_i \in \{0, 1\}, \ \ y_{ij} \in \mathbb{Z}, \ \text{all } i, j \end{aligned} \tag{21}$$

If the big-M formulation (13) of the disjunctions is used, rather than the convex hull formulation, the same MILP model results.

4.2 Uncapacitated Facility Location

An uncapacitated location problem shows how a disjunctive approach can lead to a tighter relaxation than one might obtain otherwise. The problem is the same as the previous one, except that there is no limit on the capacity of each facility.

A beginner's mistake is to model this as a special case of the capacitated problem. Although there is no capacity limit, one can observe that each factory i will ship at most $\sum_j D_j$ units and therefore let $C_i = \sum_j D_j$ in the formulation (21) for the capacitated problem. This is a valid formulation of the uncapacitated problem, but there is a much tighter one.

Let's begin again with a choice of alternatives. If factory i is installed, it supplies at most D_j to each customer j and incurs cost f_i. If it is not installed, then it supplies nothing:

$$\begin{pmatrix} 0 \leq x_{ij} \leq D_j, \text{ all } j \\ 0 \leq x_{ij} \leq K_{ij} y_{ij}, \text{ all } j \\ z_i = f_i \\ y_{ij} \in \mathbb{Z}, \text{ all } j \end{pmatrix} \vee \begin{pmatrix} x_{ij} = 0, \text{ all } j \\ z_i = 0 \end{pmatrix}$$

The convex hull formulation of this disjunction is

$$\begin{aligned} & x_{ij} \leq D_j \delta_i, \text{ all } j \\ & 0 \leq x_{ij} \leq K_{ij} y_{ij}, \text{ all } j \\ & z_i = f_i \delta_i, \quad \delta_i \in \{0,1\}, \quad y_{ij} \in \mathbb{Z}, \text{all } j \end{aligned} \tag{22}$$

This is a tighter MILP formulation than (20) with $C_i = \sum_j D_j$. To see this, note that setting $\delta_i = 1/2$ can satisfy (20) when $x_{ij} > D_j/2$ for some j, provided $x_{ik} < D_k/2$ for some $k \neq j$, whereas this solution cannot satisfy (22).

A rule of thumb for MILP modeling is that a more disaggregated model like (22) is tighter. Yet (22) results simply from using all available information when formulating each disjunct.

4.3 Lot Sizing with Setup Costs

A lot sizing problem with setup costs illustrates how logical relations among linear systems can be captured with logical constraints.

There is a demand D_t for a product in each period t. No more than K_t units of the product can be manufactured in period t, and any excess over demand is stocked to satisfy future demand. If there is no production in the previous period, then a setup cost of f_t is incurred. The unit production cost is p_t, and the unit holding cost per period is h_t. A starting stock level s_0 is given. The objective is to choose production levels in each period so as to minimize total cost over all periods.

Let x_t be the production level in period t and s_t the stock level at the end of the period. In each period t, there are three options to choose from: (1) start producing (with a setup cost), (2) continue producing (with no setup cost), and (3) produce nothing. If v_t is the setup cost incurred in period t, these correspond respectively to the three disjuncts

$$\begin{pmatrix} v_t \geq f_t \\ 0 \leq x_t \leq K_t \end{pmatrix} \vee \begin{pmatrix} v_t \geq 0 \\ 0 \leq x_t \leq K_t \end{pmatrix} \vee \begin{pmatrix} v_t \geq 0 \\ x_t = 0 \end{pmatrix} \tag{23}$$

There are logical connections between the choices in consecutive periods. If we schematically represent the disjunction (23) as $A_t \vee B_t \vee C_t$, the logical connections can be written

$$\begin{aligned} & B_t \Rightarrow (A_{t-1} \vee B_{t-1}) \\ & A_t \Rightarrow (\neg A_{t-1} \wedge \neg B_{t-1}) \end{aligned} \tag{24}$$

The inventory balance constraints are

$$s_{t-1} + x_t = D_t + s_t, \quad s_t \geq 0, \quad t = 1, \ldots, n \tag{25}$$

where s_t is the stock level in period t and s_0 is given. The problem is to minimize

$$\sum_{t=1}^{n} (p_t x_t + h_t s_t + v_t) \tag{26}$$

subject to (23) and (24) for all $t \geq 1$ and (25).

A convex hull formulation for (23) is

$$\begin{array}{lll} v_t^1 \geq f_t \alpha_t, & v_t^2 \geq 0, & v_t^3 \geq 0 \\ 0 \leq x_t^1 \leq K_t \alpha_t, & 0 \leq x_t^2 \leq K_t \beta_t, & x_t^3 = 0 \\ v_t = v_t^1 + v_t^2 + v_t^3, & x_t = x_t^1 + x_t^2 + x_t^3 & \\ \alpha_t + \beta_t + \gamma_t = 1, & \alpha_t, \beta_t, \gamma_t \in \{0, 1\} & \end{array} \tag{27}$$

Thus, $\alpha_t = 1$ indicates a startup, $\beta_t = 1$ continued production, and $\gamma_t = 1$ no production in period t. To simplify (27), first eliminate γ_t, so that $\alpha_t + \beta_t \leq 1$. Because $x_t^3 = 0$, one can set $x_1 = x_1^1 + x_2^2$, which allows one to replace the two capacity constraints in (27) by $0 \leq x_t \leq K_t(\alpha_t + \beta_t)$. Finally, v_t can replace v_t^1 because v_t is minimized. The convex hull formulation (27) becomes

$$\begin{array}{ll} v_t \geq f_t \alpha_t, \quad 0 \leq x_t \leq K_t(\alpha_t + \beta_t) & (a) \\ \alpha_t + \beta_t \leq 1 & (b) \end{array} \tag{28}$$

where $\alpha_t, \beta_t \in \{0, 1\}$. The logical constraints (24) can be formulated

$$\begin{array}{ll} \alpha_{t-1} + \beta_{t-1} - \beta_t \geq 0 & (a) \\ -\alpha_{t-1} - \alpha_t \geq -1, \;\; -\beta_{t-1} - \alpha_t \geq -1 & (b) \end{array} \tag{29}$$

The set packing inequalities in (29b) and $\alpha_{t-1} + \beta_{t-1} \leq 1$ from (28b) can be replaced by the clique inequality

$$\alpha_{t-1} + \beta_{t-1} + \alpha_t \leq 1 \tag{30}$$

The entire problem can now be formulated as minimizing (26) subject to (25) along with (28a), (29a), (30), and $\alpha_t, \beta_t \in \{0, 1\}$ for all $t \geq 1$.

4.4 Package Delivery

A final example, adapted from [1, 12], illustrates how a principled modeling approach incorporates two modeling tricks that one might otherwise miss. A collection of packages are to be delivered by several trucks, and each package j has size a_j. Each available truck i has capacity Q_i and costs c_i to operate. The problem is to decide which trucks to use, and which packages to load on each truck, to deliver all the items at minimum cost.

The decision problem consists of two levels: the choice of which trucks to use, followed by the choice of which packages to load on each truck. The trucks selected must provide sufficient capacity, which leads naturally to a 0-1 knapsack constraint:

$$\sum_i Q_i \delta_i \geq \sum_j a_j, \tag{31}$$

where each $\delta_i \in \{0, 1\}$ and $\delta_i = 1$ when truck i is selected.

The secondary choice of which packages to load on truck i depends on whether that truck is selected. If the truck i is selected, then a cost c_i is incurred, and the items loaded must fit into the truck (a 0-1 knapsack constraint). If truck i is not selected, then no items can be loaded. The disjunction is

$$\begin{pmatrix} z_i \geq c_i \\ \sum_j a_j x_{ij} \leq Q_i \\ 0 \leq x_{ij} \leq 1, \text{ all } j \\ x_{ij} \in \mathbb{Z}, \text{ all } j \end{pmatrix} \vee \left(\; x_{ij} = 0, \text{ all } j \; \right) \tag{32}$$

where z_i is the fixed cost incurred by truck i, and $x_{ij} = 1$ when package j is loaded into truck i. The associated polyhedra have the same recession cone if one adds $z_i \geq 0$ to the second disjunct. Because $\delta_i = 1$ when truck i is selected, the convex hull formulation of (32) is

$$\begin{aligned} & z_i \geq c_i \delta_i && (a) \\ & \sum_j a_j x_{ij} \leq Q_i \delta_i && (b) \\ & x_{ij} \leq \delta_i, \text{ all } j && (c) \\ & \delta_i, x_{ij} \in \{0,1\}, \text{ all } j \end{aligned} \tag{33}$$

Finally, set covering inequalities ensure that each package is shipped:

$$\sum_i x_{ij} \geq 1, \quad x_{ij} \in \{0,1\}, \text{ all } j \tag{34}$$

One can now minimize total fixed cost $\sum_i z_i$ subject to (31) and (34), as well as (33) for all i.

This formulation differs in two ways from a formulation that one might initially write for this problem. One might omit the constraint (33c) because the model is correct without it. Yet this constraint makes the relaxation tighter. Also, one might not include constraint (31), because due to (34) it is the sum of constraints (33b) over all i. Yet the presence of (31) allows the solver to deduce lifted knapsack cuts, which create a tighter continuous relaxation. This results in much faster solution.

5 Piecewise Linear Functions

Piecewise linear functions are a very useful modeling tool because they can be used to approximate separable nonlinear functions in an MILP model. Typically, one wishes to model a function of the form $g(x) = \sum_j g_j(x_j)$ by approximating each nonlinear term $g_j(x_j)$ with a piecewise linear function $f_j(x_j)$. The function $f_j(x_j)$ is set to a value equal to (or close to) $g_j(x_j)$ at a finite number of values of x_j and is defined between these points by a linear interpolation.

It is convenient to drop the subscripts and refer to $f_j(x_j)$ as $f(x)$. For the sake of generality, suppose that $f(x)$ is piecewise linear in the sense that it is linear on possibly disjoint intervals $[a_i, b_i]$ and undefined outside these intervals, as illustrated by Fig. 3. More precisely, $x \in \bigcup_{i=1}^k [a_i, b_i]$ and

$$f(x) = \begin{cases} f(a_i) + \dfrac{x - a_i}{b_i - a_i} \left[f(b_i) - f(a_i) \right] & \text{if } x \in [a_i, b_i] \text{ and } a_i < b_i \\ f(a_i) & \text{if } x = a_i = b_i \end{cases} \tag{35}$$

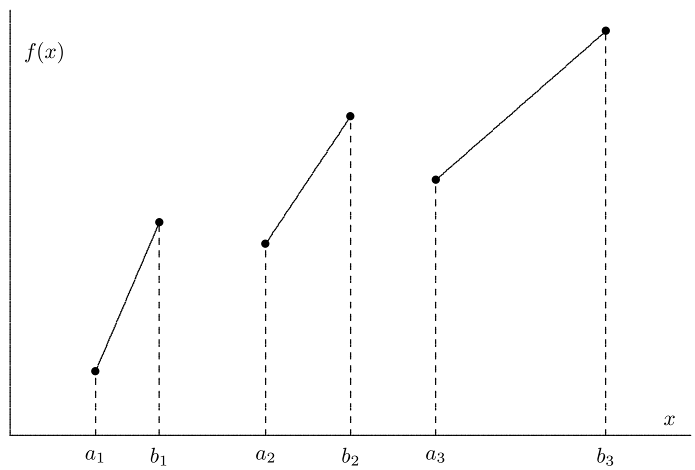

Figure 3: A piecewise linear function.

In many applications, each $b_i = a_{i+1}$, which means $f(x)$ is continuous.

A disjunctive formulation is natural and convenient for a function of this form:

$$\bigvee_i \begin{pmatrix} x = \lambda a_i + \mu b_i \\ z = \lambda f(a_i) + \mu f(b_i) \\ \lambda + \mu = 1, \ \ \lambda, \mu \geq 0 \end{pmatrix} \tag{36}$$

where disjunct i corresponds to $x \in [a_i, b_i]$, and z is a variable that represents $f(x)$. Because the disjuncts define polyhedra with the same recession cone (all the polyhedra are bounded), (36) has a locally ideal, convex hull formulation:

$$\begin{aligned} & x = \sum_i \lambda_i a_i + \mu_i b_i \\ & z = \sum_i \lambda_i f(a_i) + \mu_i f(b_i) \\ & \lambda_i + \mu_i = \delta_i, \ \text{all } i \\ & \sum_i \delta_i = 1 \\ & \lambda_i, \mu_i \geq 0, \ \delta_i \in \{0,1\}, \ \text{all } i \end{aligned} \tag{37}$$

This is similar to a well-known textbook model that dispenses with the multipliers μ_i but applies only when $f(x)$ is continuous:

$$\begin{aligned}
& x = \sum_{i=1}^{k+1} \lambda_i a_i, \;\; z = \sum_{i=1}^{k+1} \lambda_i f(a_i), \;\; \sum_{i=1}^{k+1} \lambda_i = 1 && (a) \\
& \lambda_i \le \delta_{i-1} + \delta_i, \; i = 2, \ldots, k && (b) \\
& \lambda_1 \le \delta_1, \;\; \lambda_{k+1} \le \delta_k, \; \sum_{i=1}^{k} \delta_i = 1 && (c) \\
& \lambda_i \ge 0, \; i = 1, \ldots, k+1; \;\; \delta_i \in \{0,1\}, \; i = 1, \ldots, k
\end{aligned} \tag{38}$$

where $a_{k+1} = b_k$. The model is designed to require that at most two λ_is can be positive, and if two are positive they must be adjacent. However, this model is not as tight as (37) and is not locally ideal [10]. For continuous functions $f(x)$, one can use the *incremental cost model*, which contains no more variables than (38) but is equivalent to the tight model (37) and locally ideal [11]:

$$\begin{aligned}
& x = a_1 + \sum_{i=2}^{k+1} x_i, \;\; z = f(a_1) + \sum_{i=2}^{k+1} \frac{f'_i}{a'_i} x_i \\
& 0 \le x_i \le a'_i, \; i = 2, \ldots, k+1 \\
& a'_i \delta_i \le x_i \le a'_i \delta_{i-1}, \; i = 3, \ldots, k \\
& a'_2 \delta_2 \le x_2 \le a'_2, \;\; 0 \le x_{k+1} \le a'_{k+1} \delta_k \\
& \delta_i \in \{0,1\}, \; i = 2, \ldots, k
\end{aligned} \tag{39}$$

Here $a'_i = a_i - a_{i-1}$ and $f'_i = f(a_i) - f(a_{i-1})$.

There are recent proposals for modeling piecewise linear functions with a logarithmic number of 0-1 variables [8, 14, 13].

6 Symmetry

Symmetry occurs in an MILP model when values or variables can be exchanged without affecting the problem in any significant way. Symmetries can greatly retard solution of the problem, because the solver may waste time enumerating a large number of equivalent solutions. Symmetries can sometimes be broken by adding constraints to the model, or by reformulating it completely. There are also techniques that deal with symmetries during the solution process.

Value symmetry occurs when the values can be interchanged in a solution without affecting the feasibility or optimality of that solution. For example, any two colors in a graph coloring can be swapped. *Variable symmetry* occurs when variables can be interchanged in a model without affecting the set of feasible or optimal solutions.

6.1 Symmetry-breaking Constraints

Variable symmetry can often by avoided by adding constraints that prioritize symmetric variables. This is illustrated by the package delivery problem described earlier. Suppose that there are four trucks, with capacities 80, 80, 80, and 100, respectively. There are seven packages, having sizes 30, 30, 30, 40, 40, 50, and 50.

Because trucks 1, 2, and 3 are identical, the variables $\delta_1, \delta_2, \delta_3$ can be interchanged with no effect on the problem. This symmetry can be broken by requiring that truck 1 be used first, followed by truck 2 and then truck 3. This is accomplished with the symmetry-breaking constraint $\delta_1 \geq \delta_2 \geq \delta_3$.

6.2 Symmetry-breaking Reformulation

Because some packages are the same size in the example, they give rise to another variable symmetry. Packages 1, 2, and 3 have the same size, which means that variables x_{i1}, x_{i2}, and x_{i3} can be interchanged for any i, and similarly for the other package sizes. These symmetries can be removed by redefining variables. Let y_{ik} be the number of packages of size k loaded on truck i. Then the problem is to minimize $\sum_i c_i \delta_i$ subject to (31) and

$$\sum_k \bar{a}_k y_{ik} \leq Q_i \delta_i, \ \ \delta_i \in \{0,1\}, \text{ all } i; \ \ y_{ik} \geq 0, \ y_{ij} \in \mathbb{Z}, \text{ all } i, k$$

plus the symmetry-breaking constraints for trucks, where $\bar{a}_k$ is the space consumed by packages of size k.

An alternate approach is to remove both truck and package symmetries by formulating the problem in a manner similar to the crew rostering problem described earlier. A truck of size 80 can be loaded in three ways: $30+30$, $30+40$, $30+50$, and $40+40$, which can be indexed $\ell = 1, \ldots, 4$. A truck of size 100 can be loaded in six ways: $30+30+30$, $30+30+40$, $30+50$, $40+40$, $40+50$, and $50+50$. Incomplete loadings need not be enumerated. Let $d_{t\ell}$ be a vector whose kth component is the number of packages of size k used in loading pattern ℓ on a truck of type t ($t = 1, 2$). Then the four loading patterns for truck type 1 (capacity 80) are encoded by $d_{11}, \ldots, d_{14}$, which are the columns on the left below, while the loading patterns for truck type 2 (capacity 100) appear on the right:

$$\begin{matrix} 2 & 1 & 1 & 0 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 0 \end{matrix} \qquad \begin{matrix} 3 & 2 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 2 & 0 \\ 0 & 0 & 1 & 1 & 0 & 2 \end{matrix}$$

The rows correspond to packages of sizes 30, 40, and 50. If variable $z_{t\ell}$ is the number of trucks of type t loaded with pattern ℓ, the MILP model is

$$\begin{aligned} &\min \sum_t \bar{c}_t \sum_\ell z_{t\ell} \\ &\sum_{t\ell} d_{t\ell} z_{t\ell} \geq b; \;\; z_{t\ell} \geq 0, \; z_{t\ell} \in \mathbb{Z}, \text{ all } t, \ell \end{aligned}$$

where $\bar{c}_t$ is the capacity of truck type t. The requirements vector b contains the number of packages of each type. In the example, $b = (3, 2, 2)$.

There may be a very large number of loading patterns, but as in the case of crew rostering, the problem can be solved by branch and price.

7 Extensions

MILP solvers typically supply features that go beyond MILP modeling and allow more efficient solution. The most common include special ordered sets, indicator variables, and semi-continuous variables.

7.1 Special Ordered Sets

A *special ordered set of type 1* (SOS1) is a variable set $S = \{x_1, \ldots, x_k\}$ for which exactly one variable is allowed to be positive, and the rest must be zero. SOS1 is most commonly used along with a set partitioning constraint $\sum_{j=1}^k x_j = 1$. If the model declares S to be SOS1, there is no need to restrict the variables in the set partitioning constraint to be integer. The solver enforces integrality through a efficient branching scheme that creates a branch for each $x_j \in S$ by fixing all the other variables in S to zero.

Variable set S is a *special ordered set of type 2* (SOS2) when exactly two of the variables are allowed to be positive, the rest must be zero, and any two positive variables must be adjacent (i.e., x_j and x_{j+1} for some j). SOS2 is designed to simplify the model (38) of a piecewise linear function. By declaring $\{\lambda_1, \ldots, \lambda_{k+1}\}$ to be SOS2, only constraints (a) of (38) need be included in the model. SOS2 is also implemented by an efficient branching scheme, but it sacrifices the convex hull relaxation provided by the tighter MILP models (37) and (39). In addition, it is unsuitable for a discontinuous piecewise linear function like that in Fig 3. Further options are discussed in [7].

7.2 Indicator Constraints and Semi-continuous Variables

Indicator constraints are constraints that are enforced only when a 0-1 variable is equal to one. Thus if one wishes to enforce $ax \geq \beta$ only when $\delta = 1$, the pair $(\delta, ax \geq \beta)$ is declared an indicator constraint. This device can obviously be encoded with MILP constraints. The indicator constraint obviates the encoding but sacrifices the continuous relaxation it provides.

A *semi-continuous* variable x is a related idea in which x is forced to zero when $\delta = 0$ and to be within bounds $L \leq x \leq U$ when $\delta = 1$. The MILP encoding is similar to that of a discontinuous piecewise linear function. No encoding is necessary if x is declared a semi-continuous variable.

8 Further Reading

General guidance on MILP modeling can be found in [2, 4, 15, 16]. Modeling of logical conditions is further explained in [3, 17]. The traveling salesman problem presents an interesting challenge for MILP modeling, and several formulations are discussed in [9]. A recent trend is to integrate MILP modeling with constraint programming models, because this provides a richer modeling framework that can allow the solver to exploit special structure in the problem [3, 5].

References

[1] K. Aardal. Reformulation of capacitated facility location problems: How redundant information can help. *Annals of Operations Research*, 82:289–309, 1998.

[2] C. Guéret, S. Heipcke, C. Prins, and M. Sevaux. Applications of optimization with Xpress-MP. White paper, Dash Optimization, 2000.

[3] J. N. Hooker. *Integrated Methods for Optimization*. Springer, 2007.

[4] J. N. Hooker. A principled approach to mixed integer/linear problem formulation. In J. W. Chinneck, B. Kristjansson, and M. Saltzman, editors, *Operations Research and Cyber-Infrastructure (ICS 2009 Proceedings)*, pages 79–100. Springer, 2009.

[5] J. N. Hooker. Hybrid modeling. In P. Van Hentenryck and M. Milano, editors, *Hybrid Optimization: The 10 Years of CPAIOR*. Springer, to appear.

[6] R. G. Jeroslow. Representability in mixed integer programming, I: Characterization results. *Discrete Applied Mathematics*, 17:223–243, 1987.

[7] A. B. Keha, I. R. de Farias, and G. L. Nemhauser. Models for representing piecewise linear cost functions. *Operations Research Letters*, 32:44–48, 2004.

[8] H.-L. Li, H.-C. Lu, C.-H. Huang, and N.-Z. Hu. A superior representation method for piecewise linear functions. *INFORMS Journal on Computing*, 21:314–321, 2009.

[9] A. J. Orman and H. P. Williams. A survey of different integer programming formulations of the travelling salesman problem. In E. J. Kontoghiorghes and C. Gatu, editors, *Optimisation, Economics and Financial Analysis*, Advances in Computational Management Science, pages 933–106. Springer, Berlin, 2006.

[10] M. Padberg. Approximating separable nonlinear functions via mixed zero-one programs. *Operations Research Letters*, 27:1–5, 2000.

[11] H. D. Sherali. On mixed-integer zero-one representations for separable lower-semicontinuous piecewise-linear functions. *Operations Research Letters*, 28:155–160, 2001.

[12] M. Trick. Formulations and reformulations in integer programming. In R. Barták and M. Milano, editors, *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems (CPAIOR 2005)*, volume 3524 of *Lecture Notes in Computer Science*, pages 366–379. Springer, 2005.

[13] J. P. Vielma, S. Ahmed, and G. Nemhauser. A note on: "A superior representation method for piecewise linear functions" by Li, Lu, Huang, and Hu. *INFORMS Journal on Computing*, to appear.

[14] J. P. Vielma and G. L. Nemhauser. Modeling disjunctive constraints with a logarithmic number of binary variables and constraints. In *Integer Programming and Combintorial Optimization Proceedings (IPCO 2008)*, volume 5035 of *Lecture Notes in Computer Science*, pages 199–213, 2008.

[15] H. P. Williams. *Model Building in Mathematical Programming, 4th Ed.* Wiley, New York, 1999.

[16] H. P. Williams. The formulation and solution of discrete optimization models. In G. Appa, L. Pitsoulis, and H. P. Williams, editors, *Handbook on Modelling for Discrete Optimization*, pages 3–38. Springer, 2006.

[17] H. P. Williams. *Logic and Integer Programming.* Springer, Berlin, 2009.