18th World IMACS / MODSIM Congress, Cairns, Australia 13-17 July 2009
http://mssanz.org.au/modsim09

Solving Environmental Problems with Integer Programming: Recent Experience and Challenges

N. Boland[1]
[1]*School of Mathematical and Physical Sciences*
The University of Newcastle
Callaghan, NSW 2308, Australia
Email: Natashia.Boland@newcastle.edu.au

Abstract: For most real-world problems, especially those arising in environmental decision-making, natural models are nonlinear. In optimization, the complexity of solving nonlinear problems can be reduced by introducing some appropriate problem-dependent simplification that transforms the nonlinear problem to a more easily solved integer linear programming problem. Such techniques are increasingly being utilized in the modelling and solution of environmental problems, not least because the resulting formulations can often be solved in practice: progress in linear and integer programming solvers and software tools over the last ten years or so has meant more reliable and rapid solution of even large-scale problems. This talk will describe two cases of environmental problems tackled with integer progamming, highlighting its modelling power. The first case concerns river systems, and decisions about environmental flows, addressing questions such as how much to release, and when. The second case concerns forestry. Harvest scheduling in forestry has for some time been planned with the aid of integer programming tools; now environmental considerations, such as habitat preservation, are being incorporated in such models. Solution approaches used, and future challenges, will also be discussed.

Keywords *Integer programming, environmental flows, forestry, habitat preservation*

1 Introduction

Linear programming has long been a valuable modelling tool in business and industry. It can provide assistance to evaluate what-if scenarios, and can also act “behind-the-scenes”, embedded in decision support technology, to inform decision making in a range of planning settings. The ubiquity of linear programming can be observed from the presence of a linear programming solver in the standard desktop spreadsheet, (Microsoft Excel’s Solver) as well from the plethora of linear programming software platforms available (see [8] for a recent survey of such software).

Many aspects of human decision-making in the business world are naturally linear (e.g. workers are paid per hour, establishing proportionality between cost and hours worked), so linear programming provides a natural model. The applicability of linear programming to the world of biology, which is often nonlinear, is less obvious. Somewhat surprisingly, the option of making some variables integer-valued in a linear programme, i.e turning to *integer* programming, enriches the modelling range greatly. In contrast to general nonlinear programming, integer programming has also enjoyed a rapid increase in the effectiveness of solution techniques and technology. This combination of modelling flexibility, and solution capability, is leading to increasing uptake of integer programming to address environmental questions.

In this paper, we illustrate the power of integer programming as a modelling paradigm based on two case studies. The first is very recent work by researchers based at the University of Melbourne on modelling environmental water flows in the Golbourn River. The second is a recently published application of integer programming in forestry, where the modelling power of integer linear programming is used to increase habitat preservation. We discuss how integer programming can be used to model nonlinear functions that arise in environmental modelling, and how it can be used to capture logical relationships between areas of forest and habitat construction. We then give a brief review of the state-of-the-art in integer programming solver technology, with reference to the convenient and easy-to-use modelling languages that are now available to accelerate model production. We conclude with a discussion of some of the challenges that remain, particularly for environmental modelling using integer programming.

2 Case 1: Optimization of Environmental Water Release Decisions

This case is based on the the work of Horne et al. [9], which will form part of the PhD thesis of the first author, expected to be submitted soon at the University of Melbourne[1]. The work addresses environmental release flows of water, for a river which is shared by multiple users, for example, agricultural, and/or recreational/tourism-related. The decision-making time frame addressed in this study is that of monthly decision-making, planned over an annual time horizon. The model assumes that the storage level of water in the river's storage facilities is known at the start of the year, and that the "type" of year it can be expected to be can be reliably predicted at the outset, so that expected rainfalls in each month of the year are assumed known. The authors acknowledge this is a big assumption, and intend relaxing it in future work.

For simplicity, we describe parameters and variables in terms of a river with only a single storage facility; we are concerned with the reach below the facility, and with controlling water release from the facility to address environmental needs. This is easily extended to multiple facilities (indeed the work of [9] considers two such).

Take the total capacity of the storage facility to be C (GL). We assume that the water to be released from storage for purposes such as irrigation (i.e. all non-environmental purposes) has been planned and is known, so that for each month t, the passing flow in the reach downstream from the storage facility, is known, and is denoted by f_t, for each $t = 1, \dots, T$. We let H denote the storage level of the facility at the start of the year, (in GL), and h_t denote the GL of water expected to enter the storage facility during each month t of the coming year, $t = 1, \dots, 12$, taking account of the planned releases (the f_t values). Note h_t could be negative, if rainfall in that month is not expected to exceed the combination of natural losses, such as through evaporation, and the planned releases contributing to the f_t values. Note also that if there is heavy rainfall, it is possible the dam could overflow. Since we assume that water entering the dam is known with certainty, we might also assume that the f_t values have included dam overflow. Alternatively, we could ask that the combined environmental and planned releases include any flows forced by dam overflow;

[1]The model presented here is not precisely that of the cited work; some modifications have been made to assist in the exposition.

we show how to do that below. In the meantime, we note that if there are *no* environmental releases, the water level in the storage facility in each period t, (ignoring the facility's capacity), can be calculated to be $H + \sum_{s=1}^{t} h_s$, and the passing flow in the downstream reach is f_t. It is assumed that releases have been planned so that $0 \leq H + \sum_{s=1}^{t} h_s$, and, in the case that planned releases are assumed to absorb any dam overflow, it is also assumed that $H + \sum_{s=1}^{t} h_s \leq C$ for all $t = 1, \ldots, 12$. (In fact the lower limit could reasonably be set to a positive "safety stock" level, rather than zero, but of course deciding what that value should be requires a body of research in its own right.)

The key decisions to be made in this model are the quantities of water to release each month, in addition to the quantity already planned to be released for agricultural and other purposes, specifically to meet environmental needs. These are represented by decision variables x_t, for each $t = 1, \ldots, 12$. Given these variables, the flow in the reach of interest is thus $x_t + f_t$, and the storage level in the facility is given by $H + \sum_{s=1}^{t} h_s - \sum_{s=1}^{t} x_s$, for each $t = 1, \ldots, 12$. Clearly this must be kept non-negative, i.e. we require that

$$\sum_{s=1}^{t} x_s \leq H + \sum_{s=1}^{t} h_s, \qquad \forall t = 1, \ldots, 12. \tag{1}$$

If it is not assumed that dam overflow is accounted for in the f_t values, then we may also ask that

$$\sum_{s=1}^{t} x_s \geq H + \sum_{s=1}^{t} h_s - C, \qquad \forall t = 1, \ldots, 12. \tag{2}$$

As discussed in [9], in Australia expert panels typically decide environmental flow releases over the long term. So the model assumes that the total quantity of water released for environmental purposes over the year must lie within a range specified by such a panel. Thus we required

$$m \leq \sum_{t=1}^{12} x_t \leq M, \tag{3}$$

where $[m, M]$ is the specified range for total environmental water release over the year. Together with nonnegativity of the decision variables, constraints (1) and (3) form the feasible set of a linear programme (and a rather simple one, at that).

The complexity in this model arises from the objective function. The goal of the model is to maximize the ecological benefit of the water released, and to be naturally conservative by seeking to release as little water was possible while still achieving the maximum benefit. The latter requirement is easily addressed (as we discuss later); it is the former that presents the difficulty. The real challenge is to first determine a function $R : \mathbb{R}^{12} \to \mathbb{R}$, so that $R(y)$ measures the environmental benefit achieved by flows $(y_t)_{t=1,\ldots,12}$, and then to model it in a form tractable for optimization. Horne et al. [9] in fact take R to be a measure of total environmental *risk* to be minimized. Total environmental risk is constructed in [9] from a set $\mathcal{E}$ of ecologically significant flow *elements*, via environmental response curves. For example, riffle habitat, wetland inundation, shear stress (summer), and deep water habitat (winter), are four of the nine environmental flow elements considered in [9] (four for the first reach of interest, five for the second). R is taken to be a weighted sum of risk functions for each flow element, i.e.

$$R(y) = \sum_{q \in \mathcal{E}} w_q r_q(y)$$

where w_q is the weight applied to flow element q, and r_q is its risk function, for each $q \in \mathcal{E}$. We note that $r_q(y)$ typically only depends on a subset of the elements of y, as the flow element is only of interest during certain months of the year. We write $\mathcal{T}_q \subseteq \{1, \ldots, 12\}$ to denote the set of months of interest for flow element $q \in \mathcal{E}$. The nature of $r_q(y)$ also depends on the type of year expected, and on the state of the system at the start of the planning period (for example, on the history of flows over recent years).

Horne et al. [9] describe two key classes of flow element risk function. One can be constructed as a simple sum of functions for each period. The other is more complex, but is required to model flow elements such as wetland inundation, in which an inundation event is only required once during the year. In the former case, r_q is expressed as

$$r_q(y) = \sum_{t \in \mathcal{T}_q} \rho_q(y_t)$$

where $\rho_q : \mathbb{R} \to \mathbb{R}$ measures the risk associated with flow element q as a function of monthly flow. In the latter case, monthly risk functions are also used, but since the risk needs only to be measured once, only the "high tide mark" needs to be minimized. For this case, higher water levels equate to lower risk, so the risk associated with an element such as wetland inundation is taken to be the minimum risk level seen across all periods. Thus the risk measured is that associated with the highest tide mark, and the degree of risk reflects how high that mark is. In other words, the risk function in these cases is taken to be

$$r_q(y) = \min_{t \in \mathcal{T}_q} \rho_q(y_t).$$

We will use $\mathcal{E}'$ to denote the set of flow elements using the former objective function structure, and $\mathcal{E} \setminus \mathcal{E}'$ for the latter. Thus the objective function of [9] can be expressed as

$$\min R(y) = \sum_{q \in \mathcal{E}} w_q r_q(y) = \sum_{q \in \mathcal{E}'} w_q \sum_{t \in \mathcal{T}_q} \rho_q(y_t) + \sum_{q \in \mathcal{E} \setminus \mathcal{E}'} w_q \min_{t \in \mathcal{T}_q} \rho_q(y_t),$$

where ρ_q is the monthly environmental response curve for flow element q, and $y = x + f$ is the total flow vector.

Of course, a lot of interesting science goes into constructing monthly environmental response curves. This is discussed in detail in [10]; further discussion is included in [9]. Here, with the permission of the authors, we reproduce in Figure 1 part of Figure 6 of [10], to illustrate the construction of monthly environmental response curves from habitat rating curves (such as the top left plot) and from the natural flow series, (such as the top right plot), which gives the frequency of habitat area under natural conditions. The natural flow series shows that the habitat area is normally between 3 and 6 m^2/m. Between this range it is assumed there is no environmental risk; the habitat rating curve shows the corresponding flow range of 30 to 3400 ML/d, where the environmental response curve takes on value zero. Below habitat area of 1 m^2/m the risk is taken to be 100%, and otherwise risk is taken to be proportional to habitat area lost. We note that the data in this illustration is based on work reported in a series of reports; see for example, Cottingham et al. [6, 7]. We note again that these response curves depend on – and shift according to – antecedent conditions (such as the number of years since flow was last provided) and the expected conditions in the coming year.

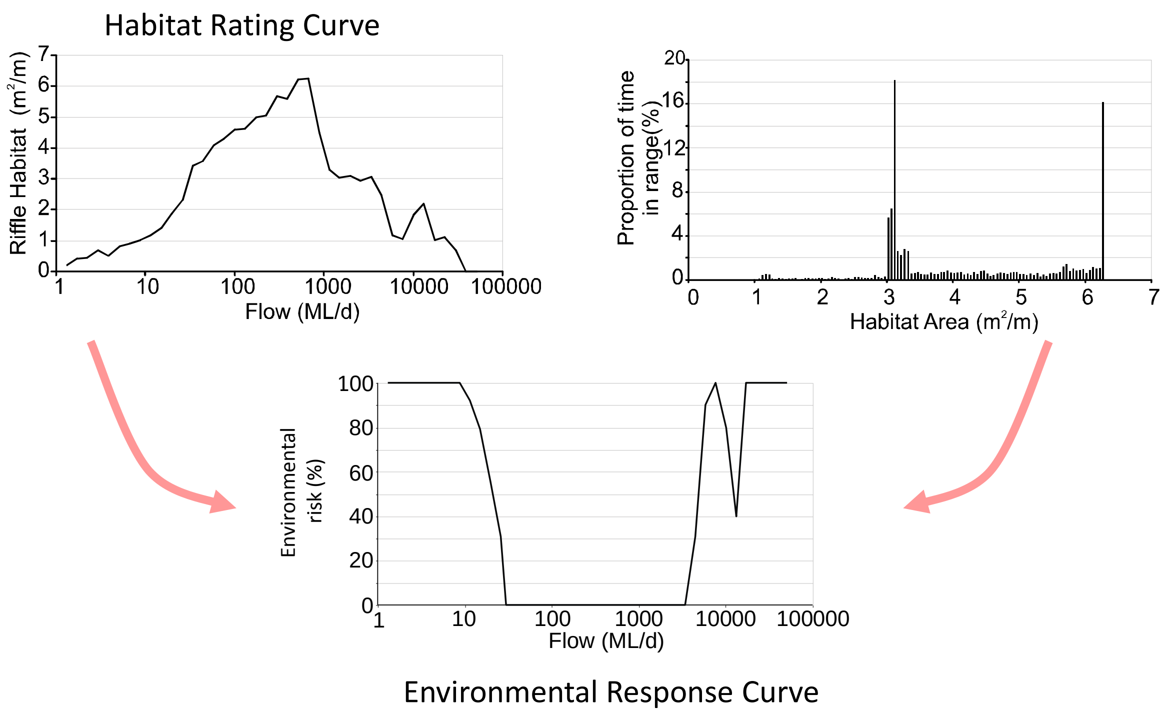

Figure 1: An illustration of the construction of environmental response curves (reproduced courtesy of Horne et al. [10].

From a modelling perspective, the most important point about environmental response curves is that they are nonconvex nonlinear functions of a single variable. In general, nonconvex nonlinear optimization is difficult. Fortunately, however, functions of a single variable can be well approximated by a piecewise linear function, (see for example Figure 2), which is amenable to integer programming modelling.

General piecewise linear functions are routinely modelled linearly with integer variables (see, for example, Chapter 9 of the undergraduate textbook [15]). Applied in this context, if the function ρ_q is defined by n_q linear pieces with endpoints $(a_0^q, b_0^q), (a_1^q, b_1^q), \ldots, (a_{n_q}^q, b_{n_q}^q)$, we define for each period $t \in \mathcal{T}_q$, n_q binary variables $g_{q,t,i}$, where $g_{q,t,i} = 1$ signifies that in period t, the ith piece of

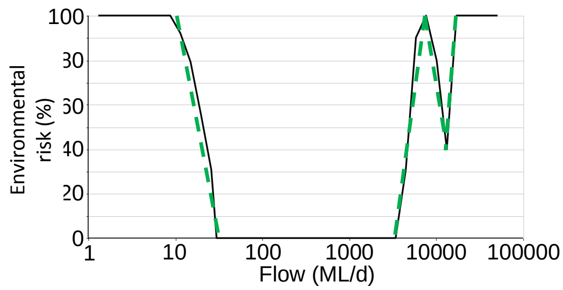

Figure 2: Approximating an environmental response curve with a piecewise linear function having seven linear pieces.

the function applies, i.e. that $a^q_{i-1} \leq x_t + f_t \leq a^q_i$. To capture the logic that exactly one piece of the function must apply, we required that

$$\sum_{i=1}^{n_q} g_{q,t,i} = 1, \qquad \forall q \in \mathcal{E},\ \forall t \in \mathcal{T}_q. \tag{4}$$

In order evaluate the function, we introduce $n_q + 1$ new nonnegative continuous variables, $\lambda_{q,t,i}$ for $i = 0, 1, \ldots, n_q$, to indicate the proportion along the ith line segment of ρ_q the flow $x_t + f_t$ is. Thus we ask that

$$\sum_{i=0}^{n_q} \lambda_{q,t,i} = 1, \qquad \forall q \in \mathcal{E},\ \forall t \in \mathcal{T}_q, \tag{5}$$

and

$$x_t + f_t = \sum_{i=0}^{n_q} a^q_i \lambda_{q,t,i}, \qquad \forall q \in \mathcal{E},\ \forall t \in \mathcal{T}_q. \tag{6}$$

Then provided that at most two of the λ variables are nonzero, and these two are adjacent, the function $\rho_q(x_t + f_t)$ can be modelled linearly via:

$$\rho_q(x_t + f_t) = \sum_{i=0}^{n_q} b^q_i \lambda_{q,t,i}, \qquad \forall q \in \mathcal{E},\ \forall t \in \mathcal{T}_q.$$

To capture the adjacent nonzero requirements on λ, we may use the following set of constraints:

$$\begin{aligned} \lambda_{q,t,0} \leq\ & g_{q,t,1}, & \forall q \in \mathcal{E},\ \forall t \in \mathcal{T}_q, \\ \lambda_{q,t,i} \leq\ & g_{q,t,i-1} + g_{q,t,i}, \quad \forall q \in \mathcal{E},\ \forall t \in \mathcal{T}_q,\ \forall i = 1, \ldots, n_q - 1, & \text{and} \\ \lambda_{q,t,n_q} \leq\ & g_{q,t,n_q}, & \forall q \in \mathcal{E},\ \forall t \in \mathcal{T}_q. \end{aligned} \tag{7}$$

Now we have all but completed a linear model: it remains to consider the min function used for $q \in \mathcal{E} \setminus \mathcal{E}'$. We simply introduce a new continuous variable μ_q for each $q \in \mathcal{E} \setminus \mathcal{E}'$, add binary variables $v_{q,t}$ for each $t \in \mathcal{T}_q$ to indicate whether or not $\rho_q(x_t + f_t)$ is the smallest of these, and add constraints to ensure μ_q takes on this value. Thus we need

$$\sum_{t \in \mathcal{T}_q} v_{q,t} = 1, \qquad \text{and} \qquad \mu_q \geq \sum_{i=0}^{n_q} b_i^q \lambda_{q,t,i} - \left(\max_{i=1,\ldots,n_q} b_i^q \right) v_{q,t}, \forall t \in \mathcal{T}_q, \qquad \forall q \in \mathcal{E} \setminus \mathcal{E}'. \quad (8)$$

The minimization objective will naturally ensure μ_q takes on the minimal value, but redundant constraints to that effect may also be helpful. Putting it all together, the integer programming model is given by

$$\begin{array}{ll} \min & \sum_{q \in \mathcal{E}'} w_q \sum_{t \in \mathcal{T}_q} \sum_{i=0}^{n_q} b_i^q \lambda_{q,t,i} + \sum_{q \in \mathcal{E} \setminus \mathcal{E}'} w_q \mu_q \\ \text{s.t.} & (1), (3), (4), (5), (6), (7), \text{ and } (8). \end{array}$$

We note that this optimization model is solved first and the minimal risk found; then this objective function is constrained to equal the minimum risk, while the secondary objective,

$$\min \sum_{t=1}^{12} x_t$$

that of minimizing total environmental flow release, is minimized. In other words they have taken a lexicographic goal programming approach. An alternative would be to take a weighted goal programming approach, indeed this is the approach used in the second case, discussed below. Another alternative would be to seek multiple Pareto-optimal solutions to explore the trade-off between risk and total water release, however this would come at greater computational cost.

3 Case 2: Reducing Habitat Fragmentation in Forestry

This case is based on the recent paper of Öhman and Wikström [12]. Here a forest area that is subject to harvesting is partitioned into regions called *stands*. The forest management must decide a *treatment schedule* for each stand over the planning horizon: for each period, should the stand be thinned, harvested, or neither? A stand is characterized as "old" if it met certain criteria, e.g. has not been harvested for at least 100 years and has not been thinned for at least 25 years. The time horizon is sufficiently long that some stands could become old during the horizon. As in Case 1, the forest management had two objectives: the first is to maximize the NPV of the harvest; the second is to minimize fragmentation of old forest. The latter is addressed by minimizing the perimeter of contiguous stands of old forest, subject to the requirement that a minimum area of old forest is maintained. This case illustrates key integer programming features: the use of very large-scale models of a type often addressed via column generation techniques, arising in the harvest scheduling aspect of the problem, and the modelling of logical relationships, arising in the old forest fragmentation reduction aspect.

Due to the complex, nonlinear nature of forest growth in response to various silvicultural activities, the harvest aspect of the problem is embedded in a subproblem. At the master problem level, for each stand s, we imagine constructing the set K_s of all possible treatment schedules for

s. For each $k \in K_s$, the NPV is calculated, denoted by $v_{s,k}$. For a given treatment schedule, it is also not difficult to calculate the harvest volume from the stand in each period, denoted by $h_{s,k,t}$, and the area of old forest in the stand, denoted by $g_{s,k,t}$, (this is set either to the area of the stand if it meets the criteria for old forest in the period under the given treatment schedule, and zero otherwise), for each period $t = 1, \ldots, T$, where T is the number of planning periods. The master problem decisions are binary variables denoted by $x_{s,k}$, set to 1 if treatment schedule $k \in K_s$ is selected for stand $s \in S$, where S is the set of all stands. The model seeks to select a treatment schedule for each stand, which is easily modelled via

$$\sum_{k \in K_s} x_{s,k} = 1, \qquad \forall s \in S \tag{9}$$

while ensuring that harvest volumes do not fluctuate too greatly from one period to another. The absolute change in harvest volume from period t to $t+1$ can be written as

$$\sum_{s \in S} \sum_{k \in K_s} (h_{s,k,t+1} - h_{s,k,t}) x_{s,k},$$

however forest management wishes the *relative* change to be in the interval $[\alpha, \beta]$. This can be expressed linearly as

$$\alpha \sum_{s \in S} \sum_{k \in K_s} h_{s,k,t} x_{s,k} \leq \sum_{s \in S} \sum_{k \in K_s} (h_{s,k,t+1} - h_{s,k,t}) x_{s,k} \leq \beta \sum_{s \in S} \sum_{k \in K_s} h_{s,k,t} x_{s,k}, \qquad \forall t = 1, \ldots, T. \tag{10}$$

The total area of old forest must be maintained above some lower threshold value, m, so we require

$$\sum_{s \in S} \sum_{k \in K_s} g_{s,k,t} x_{s,k} \geq m, \qquad \forall t = 1, \ldots, T. \tag{11}$$

The harvest NPV maximization problem is thus

$$\begin{array}{ll} \max & \displaystyle\sum_{s \in S} \sum_{k \in K_s} v_{s,k} x_{s,j} \\ \text{s.t.} & (9),\ (10)\ \text{ and } (11) \text{ all hold.} \end{array}$$

Harvest models of this type have been used extensively in forestry for some time now, as references in [12] show; see also the survey paper by Rönnqvist [13]. Surprisingly, models of this type are quite well-solved, using a technique known as *column generation*. Even though the model has potentially of order $|S|3^T$ variables, (for each stand, there may be of order 3^T treatment schedules, since one of three actions is possible for the stand in each of the T periods), an optimization subproblem can be used to find "useful" treatment schedules on the fly; the whole set K_s does not have to be generated.

Now to protect habitat, it is considered helpful to retain contiguous regions of old forest that are "compact", not "stretched out". This can be achieved by minimizing the perimeter of old growth areas (while of course retaining the area above the required threshold). Let p_s denote the perimeter of stand s. Then the sum of perimeters of old forest is given by $\sum_{s \in S} \sum_{k \in K_s} p_s \mathbf{I}[g_{s,k,t}] x_{s,k}$, where

$\mathbf{I}[\cdot]$ denotes the indicator function, set to 1 if the argument is nonzero, and zero otherwise. To find the actual perimeter of contiguous areas, we simply subtract off, for each pair of contiguous stands, the length of their common boundary. So if $q_{r,s}$ is the length of the common boundary between stands r and s, (set to be zero if they have no common boundary), we simply subtract $\sum_{s\in S}\sum_{k\in K_s}\sum_{r\in S}\sum_{k'\in K_r} q_{s,r}\mathbf{I}[g_{s,k,t}g_{r,k',t}]x_{s,k}x_{r,k'}$ off the total perimeter. This is unfortunately nonlinear in x. It could be linearized by standard means, but this would lead to an intractably large number of variables, and alter the structure of the subproblem so as to become rather difficult. Instead, Öhman and Wikström [12] cleverly introduce binary variables $z_{r,s,t}$ which are set to 1 if both stands r and s have been assigned to schedules that make them old growth in period t. This is modelled via

$$\sum_{k\in K_s}\mathbf{I}[g_{s,k,t}]x_{s,k} \geq z_{r,s,t}, \quad \text{and} \qquad \sum_{k\in K_r}\mathbf{I}[g_{r,k,t}]x_{r,k} \geq z_{r,s,t}, \quad \forall r,s\in S, r<s, \forall t=1,\ldots,T. \tag{12}$$

Now to minimize the perimeter, we simply minimize

$$\min\sum_{t=1}^{T}\left(\sum_{s\in S}\sum_{k\in K_s} p_s\mathbf{I}[g_{s,k,t}]x_{s,k} - \sum_{r\in S}\sum_{s\in S, r<s} q_{r,s}z_{r,s,t}\right).$$

Öhman and Wikström [12] combine the two objectives into a single objective, by taking a weighted sum of the two (a weighted goal programming approach). Again, trade-offs could be explored by instead generating Pareto-optimal solutions.

4 Solving Integer Programs

In recent years, there has been dramatic improvement in both tools available to model integer programs, and technology available to solve them. For an overview of available tools, we refer the interested reader to the survey of Fourer [8], which can be viewed online. This represents the software offered by around 29 vendors. Although the survey is titled "Linear programming", in fact the majority of systems described handle integer variables. All commercial software discussed in the remainder of this paper is available from vendors listed in [8], which gives contact details. We note there is also rapid growth in "freeware" for both linear and integer programming; see, for example, the COIN-OR website [5], or that of SCIP [14].

Modelling languages, such as AMPL, AIMMS, GAMS, Mosel, and the like, enable very fast implementation, so new models and ideas can be trialled quickly. To illustrate, in AMPL, constraint (1), can be written

```
subject to StoreConstraint{t in 1..12}:
      sum {s in 1..t} x[s] <= H + sum {s in 1..t} h[s];
```

while in Mosel, the same constraint might appear as

```
forall (t in 1..12) StoreConstraint(t):=
      sum(s in 1..t) x(s) <= H + sum(s in 1..t) h(s)
```

Such modelling languages often come with quite a bit of programming power. The most recently developed is Mosel, which in the author's opinion, offers the most flexible programming capability.

Progress in integer programming solution has in part been driven by tremendous advances in linear programming solver technology. As discussed by Bixby [3], from 1987 to 2000, algorithmic speed-ups in linear programming have improved solution times by three orders of magnitude. During the same period, of course computers also got faster, also by an estimated three orders of magnitude. In combination, this means that a problem that might have taken ten years to solve in 1997 could be solved in the year 2000 in 30 seconds. Since integer programming relies on solving linear programs, there has been a flow-on effect for integer programming solution, in addition to the benefit derived from improvements to integer programming solution algorithms themselves.

Integer programming progress was reported slightly later by Bixby et al. [4]. At that time, it was clear that improvements in integer programming solvers were having more impact on harder problems, with the harder problems showing greater speedup factors with more recent solver versions. More recent studies, such as that of Laundy et al. [11], have focussed on a new testbed, MIPLIB 2003 [1], developed in 2003 in response to the need for harder benchmark data. Compelling graphics on the [1] and [14] websites attest to the progress that has been made through solver technology on these difficult problems. What is not clearly shown in these results, however, is the increasing ease with which "average" real-world problems are being solved. To illustrate, we note that in the computational study of Öhman and Wikström [12], they used AIMMS to model the problem and ILOG Cplex as the solver. Their problem had about 95,000 variables and 64,000 constraints, and it was solved to within 0.01% of optimality. The five problem variants they tackled solved in between 10 to 82 minutes of CPU time, with an average of about 41 minutes – not an unreasonable time to plan more than 100 years of forest growth!

5 Conclusions

We conclude that integer programming offers a powerful modelling paradigm for environmental problems, and is accompanied by solution technology that make it highly accessible, and eminently practical. Of course, challenges remain. There are still classes of relatively small problems that are difficult to solve. Often problem classes require "tuning" of the many parameters governing the performance of integer programming software. However Atamtürk and Savelsbergh [2] provide a recent study on this topic, and we note that very recently one vendor, of the Xpress-MP software suite, offers an automated tuning capability.

One particular challenge for the adoption of integer programming in environmental modelling may be the issue of multiple objectives, and indeed the whole notion of optimization. How can one trade off environmental benefit against net present value for a commercial operation? How can one trade off one form of environmental benefit against another? Sensitivity analysis may offer some relief, but whereas sensitivity analysis is fully developed for linear programming, it is less well developed for integer programming, and systems typically do not offer much in this regard at present. The two cases discussed here took a goal programming approach, but perhaps techniques that facilitate exploration of the Pareto-optimal frontier are warranted; a more complete investigation of multi-objective optimization is required.

Furthermore, environmental modelling depends critically on factors that are uncertain, and difficult to predict, such as future weather patterns, or indeed the precise impact of environmental conditions on particular species. Thus it is important that models consider stochastic effects.

Whilst there has been some development in multi-objective and stochastic integer programming, it is relatively early days for these areas, and much remains to be done. But probably the most significant challenge to the uptake of what integer programming has to offer is the need for separate communities of researchers to interact, and learn to speak to one another. Only in this way will true progress be made.

References

[1] Achterberg, T., T. Koch and A. Martin, "MIPLIB 2003", *Operations Research Letters* **34**, 2006, 361–372, http://miplib.zib.de/

[2] Atamtürk, A. and M.W.P. Savelsbergh, "Integer programming software systems", *Annals of Operations Research* **140**, 2005, 67-124.

[3] Bixby, R.E., "Solving real-world linear programs: a decade and more of progress", *Operations Research* **50**, 2002, 3-15.

[4] Bixby, R. E., M. Fenelon, Z. Gu, E. Rothberg and R. Wunderling, "Mixedinteger programming: A progress report", in: *The Sharpest Cut: The Impact of Manfred Padberg and His Work*, M. Grötschel (ed.), MPS-SIAM Series on Optimization 4, 2004.

[5] http://www.coin-or.org/

[6] Cottingham, P., M. Stewardson, D. Crook, T. Hillman, J. Roberts and I. Rutherfurd, *Environmental flow recommendations for the Goulburn River below Lake Eildon*. Technical Report 01/2003, CRC Freshwater Ecology and CRC Catchment Hydrology, 2003.

[7] Cottingham, P., M. Stewardson, D. Crook, T. Hillman, R. Oliver, J. Roberts and I. Rutherfurd, *Evaluation of summer inter-valley water transfers from the Goulburn River*. Prepared for the Goulburn-Broken Catchment Management Authority, Melbourne, 2007.

[8] Fourer, R., 'Linear programming", *OR/MS Today*, June 2007, http://lionhrtpub.com/orms/surveys/LP/LP-survey.html.

[9] Horne, A., O. Smith, T.A. MchMahon and J. Freebairn, "Optimisation of environmental release decisions: building an annual conceptual model", *Working Paper*, 2009.

[10] Horne, A., M. Stewardson, J. Freebairn and T.A. MchMahon, "Using an economic framework to inform management of environmental entitlements", *Submitted paper*, 2008.

[11] Laundy, R., M. Perregaard, G. Tavares, H. Tipic and A. Vazacopoulos, "Solving hard mixed integer programming problems with Xpress-MP: A MIPLIB 2003 case study", *Rutcor Research Report*, RRR 2-2007, 2007.

[12] Öhman, K. and P. Wikström, "Incorporating aspects of habitat fragmentation into long-term forest planning using mixed integer programming", *Forest Ecology and Management*, **255**, 2008, 440-446.

[13] M. Rönnqvist, "Optimization in forestry", *Math. Programming, Ser. B*, **97**, 2003, 267–284.

[14] http://scip.zib.de/

[15] W. Winston, *Operations Research: Applications and Algorithms*, (3rd ed.), Duxbury Press, 1994.