Optimization under uncertainty: modeling and solution methods

Paolo Brandimarte
Dipartimento di Scienze Matematiche
Politecnico di Torino

e-mail: paolo.brandimarte@polito.it
URL: http://staff.polito.it/paolo.brandimarte

Lecture 4: Further modeling and approximation issues: stochastic mixed-integer programming and end effects
Consider the four binary variables $x_0, x_1, x_2, x_3 \in \{0, 1\}$ and the constraints:

$$3x_0 \leq x_1 + x_2 + x_3$$  \hspace{1cm} (1)

and

$$x_0 \leq x_1, \quad x_0 \leq x_2, \quad x_0 \leq x_3.$$  \hspace{1cm} (2)

Questions:

1. Is there any difference in the feasible sets of (1) and (2)?
2. If not, which of the two formulations would you prefer? Is it better to have one constraint or three?

The key issue is: What makes a branch and bound solution method fast or slow?
Motivation

- Stochastic LP problems are hard, and stochastic MILPs are harder.
- Model reformulation methods from deterministic integer programming may be used to improve solution efficiency.
- Tighter formulations may also improve performance of LP-based heuristics.
- In general, the more stages we have, the larger the tree. How can we reduce the planning horizon without suffering from end effects?
REFERENCES

REFERENCES


OUTLINE

1. STOCHASTIC MIXED-INTEGER LINEAR PROGRAMMING

2. DEALING WITH END EFFECTS IN MULTISTAGE PLANNING
Branch and bound methods for MILP
Automatic model strengthening in MILP
Model reformulation for efficiency
Heuristics for stochastic MILP

**Branch and bound methods**

Let us consider a MILP problem:

$$P(S) \quad \min \quad c^T x + d^T y$$

s.t. \hspace{1cm} \begin{align*}
Ax + Ey & \leq b \\
x & \in \mathbb{R}^{n_1}_+, \quad y & \in \mathbb{Z}^{n_2}_+
\end{align*}$$

where $S$ denotes its feasible set.

Due to its discrete character, the usual optimality conditions do not apply. Furthermore, the problem is not convex.

It is useful to consider its continuous (LP) relaxation:

$$P(\overline{S}) \quad \min \quad c^T x + d^T y$$

s.t. \hspace{1cm} \begin{align*}
Ax + Ey & \leq b \\
\begin{bmatrix} x \\ y \end{bmatrix} & \in \mathbb{R}^{n_1+n_2}_+
\end{align*}$$

with relaxed feasible set $\overline{S}$. 

P. Brandimarte – Dip. di Scienze Matematiche
Optimization Under Uncertainty
Branch and bound methods for MILP
Automatic model strengthening in MILP
Model reformulation for efficiency
Heuristics for stochastic MILP

Branch and bound methods

The LP-relaxation can be solved by the simplex method, but in general we do not get a feasible solution.

For instance, consider the knapsack problem:

\[
\text{max} \quad 10x_1 + 7x_2 + 25x_3 + 24x_4 \\
\text{s.t.} \quad 2x_1 + 1x_2 + 6x_3 + 5x_4 \leq 7 \\
x_j \in \{0, 1\}
\]

If we relax the integrality condition to \(x_j \in [0, 1]\), we obtain

\[
x_1 = 1, \quad x_2 = 1, \quad x_3 = 0, \quad x_4 = 0.8
\]
with objective value 36.2.

This objective value is an upper bound on the optimal value.

If we had a feasible solution with value 36, what could we conclude?

Roles are reversed for minimization problems.
In general, we can only find an optimal MILP solution by enumeration, which in the knapsack case can be accomplished by the following search tree:

Unfortunately, the tree size grows exponentially in the size of the problem, but we can considerably reduce the computational effort using upper and lower bounds.
Branch and bound methods for MILP
Automatic model strengthening in MILP
Model reformulation for efficiency
Heuristics for stochastic MILP

Branch and bound methods

If we have an integer solution \( \hat{x} \), its cost is an upper bound UB on the optimal value (minimization problem).

If we compute a lower bound LB for a node (subproblem) in the tree, and \( LB \geq UB \), then we may safely prune that branch of the tree.

Branch and bound methods generate tree branches, and try to limit the search effort using bounds. In the general integer case, when we branch on an integer variable taking fractional value \( \bar{x}_j \) in the LP relaxation, two subproblems are generated, with additional constraints \( x_j \leq \lfloor \bar{x}_j \rfloor \) and \( x_j \geq \lfloor \bar{x}_j \rfloor + 1 \), respectively.

Clearly, efficiency is obtained if (in the case of minimization):

1. we obtain tight upper bounds, possibly by invoking high quality heuristics;
2. we obtain tight lower bounds.
The quality of the lower bound depends on the gap between the feasible region $S$ and its continuous relaxation $\overline{S}$.

If $\overline{S}$ is just the convex hull $[S]$, then we immediately find an integer solution.

The closer $\overline{S}$ to $[S]$, the better.

From this point of view, the disaggregated formulation (2) is better than (1), since by aggregating constraints we relax the feasible region for the LP problem.

In fact, any state-of-the-art commercial solver automatically reformulates (1) as (2).
Cover cuts

Many types of cuts can be automatically generated, cutting portions of the LP polyhedron not including integer solutions.

Let us consider feasible set of the previous knapsack problem:

\[ 2x_1 + 1x_2 + 6x_3 + 5x_4 \leq 7 \]

Clearly, items 1 and 3 cannot be both selected, as their total weight is 8. The same applies to items 1, 2, and 4.

Hence we may add the cuts

\[ x_3 + x_4 \leq 1 \]
\[ x_1 + x_2 + x_4 \leq 2 \]

They are obviously redundant in the discrete domain, but not redundant in the continuous relaxation.

Now, LP relaxation yields a stronger bound \( 34.66667 < 36.2 \).
THE ROLE OF THE BIG-\(M\)

In some cases, automatic reformulation is not possible.

Consider a fixed charge constraint \(x \leq M\delta\), where \(x \geq 0\), \(\delta \in \{0, 1\}\), and \(M\) is a suitably large constant.

The smaller the big-\(M\), the tighter the LP bound.
MULTI-ITEM LOT SIZING

Let us consider the lot-sizing problem (see Brandimarte, 2011):

\[
\begin{align*}
\min & \quad \sum_{i=1}^{N} \sum_{t=1}^{T} (h_i l_{it} + f_i \delta_{it}) \\
\text{s.t.} & \quad l_{it} = l_{i,t-1} + x_{it} - d_{it}, \quad i = 1, \ldots, N, \quad t = 1, \ldots, T, \\
& \quad \sum_{i=1}^{N} (r_{im} x_{it} + r'_{im} \delta_{it}) \leq R_{mt}, \quad m = 1, \ldots, M, \quad t = 1, \ldots, T, \\
& \quad x_{it} \leq M_{it} \delta_{it}, \quad i = 1, \ldots, N, \quad t = 1, \ldots, T, \\
& \quad x_{it}, l_{it}, \delta_{it} \geq 0; \quad \delta_{it} \in \{0, 1\}
\end{align*}
\]  

(3)

In concrete, we may set the big-M as the total demand of item $i$, over the time buckets from $t$ to $T$:

\[
x_{it} \leq \left( \sum_{\tau=t}^{T} d_{i,\tau} \right) \delta_{it}, \quad \forall i, t
\]

(4)
One way to reduce this big-$M$ is to disaggregate the production variable $x_{it}$ to a set of decision variables $y_{itp}$, representing the amount of item $i$ produced during time bucket $t$ to satisfy the demand during time bucket $p \geq t$.

This new variable represents a disaggregation of the original variable $x_{it}$, since

$$x_{it} = \sum_{p=t}^{T} y_{itp}$$

In some sense, when we perform a setup for item $i$ in time bucket $t$, we “open” a plant in time bucket $t$.

Then, we transport items in time, incurring inventory holding cost.
Stochastic mixed-integer linear programming
Dealing with end effects in multistage planning
Branch and bound methods for MILP
Automatic model strengthening in MILP
Model reformulation for efficiency
Heuristics for stochastic MILP

**PLANT LOCATION REFORMULATION**

Supply periods

- $x_{i1}$
- $x_{i2}$
- $x_{i3}$
- $x_{i4}$

Demand periods

- $y_{i11}$
- $y_{i12}$
- $y_{i13}$
- $y_{i14}$

$y_{i33}$

$y_{i34}$

$d_{i1}$

$d_{i2}$

$d_{i3}$

$d_{i4}$

P. Brandimarte – Dip. di Scienze Matematiche
Optimization Under Uncertainty
The following model formulation is much tighter and efficient, because of a smaller big-$M$:

\[
\begin{align*}
\min & \quad \sum_{t=1}^{T} \sum_{i=1}^{N} \left( f_i \delta_{it} + \sum_{p=t+1}^{T} (p - t) h_{i} y_{itp} \right) \\
\text{s.t.} & \quad \sum_{i=1}^{N} \sum_{p=t}^{T} r_{im} y_{itp} + \sum_{i=1}^{N} r_{im}' \delta_{it} \leq R_{mt}, \quad \forall m, t \\
& \quad y_{itp} \leq d_{ip} \delta_{it}, \quad \forall i, t, p \geq t \\
& \quad \sum_{t=1}^{p} y_{itp} = d_{ip}, \quad \forall i, p \\
& \quad y_{itp} \geq 0, \quad \forall i, t, p \geq t \\
& \quad \delta_{it} \in \{0, 1\}, \quad \forall i, t 
\end{align*}
\]

P. Brandimarte – Dip. di Scienze Matematiche

Optimization Under Uncertainty
Some specific decomposition methods have been proposed for stochastic MILPs: see Chapter 7 of (Birge and Louveaux, 2011) or (Sen, 2005).

Still, large-scale Stochastic MILPs are beyond the reach of current technology and we must often settle for heuristics.

One possibility is using clever rounding of the continuous relaxation, or a restricted tree search.

Another possibility is taking advantage of problem structure.
THE IMPORTANCE OF A TIGHT FORMULATION

Rounding a continuous solution to integer values does not work, unless you have a tight formulation.

Consider the following pure integer problem (Williams, 1999):

\[
\begin{align*}
\text{max} & \quad x_1 + x_2 \\
\text{s.t.} & \quad 10x_1 - 8x_2 \leq 13 \\
& \quad 2x_1 - 2x_2 \geq 1 \\
& \quad x_1, x_2 \in \mathbb{Z}_+
\end{align*}
\]

If we relax the integrality requirement, i.e., we just require \( x_1, x_2 \geq 0 \), we can apply the simplex method and find

\[
x_1^* = 4.5, \quad x_2^* = 4
\]

with an optimal objective value 8.5.

If try rounding this solution up and down, what do you get?
The importance of a tight formulation

Unfortunately, the trivially rounded solutions are not feasible. In fact, the integer optimal solution is $x_1^* = 2$, $x_2^* = 1$ with optimal value 3.

The figure illustrates the big gap between $\bar{S}$ and $S$. 
The fix-and-relax heuristic has been proposed in (Dillenberger, 1994).

Consider an MILP where variables $x_j$ in set $V$ are binary, and the remaining variables are continuous.

Partition the set $V$ in subsets $V_i$, $i = 1, \ldots, k$.

Solve a sequence of $k$ problems, where problem $r = 1, \ldots, k$ is subject to the following requirements:

\begin{align*}
  x_j &= \hat{x}_j, \quad \forall j \in V_i, \ i = 1, \ldots, r - 1 \quad \text{(for } r > 1) \\
  x_j &\in \{0, 1\}, \quad \forall j \in V_r \\
  x_j &\in [0, 1], \quad \forall j \in V_i, \ i = r + 1, \ldots, k \quad \text{(for } r < k)
\end{align*}

In practice, we perform a restricted branch and bound search, which requires a tight formulation anyway.
Plant location reformulation: stochastic case

The following is a disaggregated model for multistage lot-sizing under demand uncertainty and lost sales (see Brandimarte, 2006):

\[
\begin{align*}
\min & \sum_{n \in \mathcal{N}} p^n \left[ \sum_{i} \left( f_i \delta_i[n] + h_i l_i[n] + g_i z_i[n] \right) \right] + \sum_{n \in \mathcal{N} \setminus \mathcal{T}} p^n \left[ \sum_{i} \sum_{\tau > T(n)} h_i(\tau - T(n)) y_{i,\tau}^n \right] \\
\text{s.t.} & \quad l_i[a(n)] + \sum_{t < T(n)} y_{i,\tau}^{\Omega(n,t)} + y_{i,\tau(n)} = d_i[n] + l_i[n] - z_i[n] \quad \forall i, n \\
& \quad y_{i,\tau}^n \leq \left( \max_{j \in \Sigma(n, \tau)} d_j[j] \right) \delta_i[n] \quad \forall i, n, \tau > T(n) \\
& \quad y_{i,\tau(n)} \leq d_i[n] \delta_i[n] \quad \forall i, n \\
& \quad \sum_{i} \sum_{\tau \geq T(n)} r_i y_{i,\tau}^n + \sum_{i} r'_i \delta_i[n] \leq R \quad \forall n \\
& \quad y_{i,\tau}^n, l_i[n], z_i[n] \geq 0; \quad \delta_i[n] \in \{0, 1\}
\end{align*}
\]
Taking advantage of the previous reformulation, a fix-and-relax strategy can be pursued.

A natural partition of setup variables is based on the time index, resulting in a sort of forward time-sweep strategy which enforces integrality of setup variables one time-layer at a time.

1. Set $i = 1$.

2. Solve SCLSP by branch and bound with fixed setup variables for $t < i$ and relaxed setup variables for $t > i$ (possibly within a relative suboptimality gap).

3. Set $i = i + 1$ and repeat step 2 until the end of the planning horizon is reached.
PROGRESSIVE HEDGING FOR SINGLE-ITEM STOCHASTIC LOT SIZING

The uncapacitated, single-item lot-sizing problem is, in the deterministic case a very easy problem to solve.

It is easy to show that in the optimal solution we must have $l_{t-1}x_t = 0$.

This Wagner-Whitin property leads to a very efficient solution procedure by dynamic programming.

Unfortunately, this property does not extend to the stochastic case.

Nevertheless, the single item problem, is simple enough to be solved by dynamic programming (or heuristics, if needed), even with an augmented objective function.
PROGRESSIVE HEDGING FOR SINGLE-ITEM STOCHASTIC LOT-SIZING

In (Haugen et al., 2001), progressive hedging is used as a heuristic for single-item stochastic lot-sizing.

Dualization of non-anticipativity constraints, and the introduction of a regularization term (augmented Lagrangian penalty) yields a set of individual scenario subproblems that are coordinated by dual variables.

It is possible to solve individual problems approximately, since in any case convergence is problematic here.

An implementable solution is progressively built. It is also possible to keep the binary variables, and reoptimize with respect to continuous variables.

The approach has been also proposed for other network flow problems.
CONSUMPTION/SAVING PROBLEM

Consider a typically stylized problem in Economics:

\[
\max \quad E_t \left[ \sum_{\tau=t}^{T} \beta^{\tau-t} u(C_{\tau}) \right] \\
\text{s.t.} \quad S_{\tau} = X_{\tau} - C_{\tau} \\
\quad X_{\tau} = R_{\tau} S_{\tau-1} + Y_{\tau}
\]

where

- $\beta$ is a discount factor and $u(\cdot)$ is a concave utility function;
- $X_{\tau}$ is cash on hand at time $\tau$;
- $C_{\tau}$ are $S_{\tau}$ consumption and saving at time $\tau$, respectively;
- $R_{\tau}$ is the return from period $\tau - 1$ to $\tau$, and $Y_{\tau}$ is the noncapital income.

What is the saving decision at time $T$?
Sometimes, the previous model is adjusted by including a function $B(X_T)$ of terminal wealth, called utility of bequest, in the objective.

This may be somewhat arbitrary, and infinite time horizons are often used to overcome the issue (they can be tackled by dynamic programming; see Lecture 8).

However, this may not be the best solution for operational problems where we have time-varying data.

When dealing with dynamic decisions over time, the terminal state and decision may influence the early decisions, unless the planning horizon is very long.

In multistage stochastic programming, a long planning horizon is associated with a possibly huge scenario tree.
To further illustrate terminal state issues, we recall that deterministic, single-item, infinite capacity lot-sizing problems can be solved quite easily by taking advantage of Wagner-Whitin (WW) property.

Is the procedure actually used? No! Why???

In practice, rolling horizon simulations show that the “optimal” solution is outperformed by heuristics.

This is due to the fact that the ending inventory in the optimal solution is zero, and when a new demand bucket enters the planning horizon, a drastic change in the ordering pattern can occur.
Conceputally, we may be dealing with an infinite horizon problem, but we have to settle for a finite planning horizon for lack of information and computational limits.

We need a way to approximate the infinite time horizon problem with a finite one, in such a way that when applying the policy according to a rolling horizon, we obtain good performance.

In (Grinold, 1983) four general strategies to cope with end effects are considered:

1. truncation
2. salvage value
3. primal equilibrium
4. dual equilibrium
In (Fisher et al., 2001), the following modified objective function is proposed for single-item lot sizing (deterministic case):

$$\sum_{t=1}^{T} [K\delta_t + hl_t] + \left\{ K - \frac{h}{2D}(q^* - l_T)^2 \right\}$$

where $q^*$ is the optimal order quantity for an infinite horizon problem with constant demand $D$.

The ending inventory valuation term can be interpreted as the future setup cost avoided by leaving a terminal inventory $l_t$.

This strategy may be interpreted as a salvage value approach.
An example of primal equilibrium

We consider a fisheries management problem borrowed from (Kall, Wallace, 1994).

Let \( z_t \) the biomass of fish stock available (state variable) and \( x_t \in [0, 1] \) be the fraction of stock caught (control variable).

A possible state transition function is:

\[
z_{t+1} = z_t - x_t z_t + \rho z_t \left( 1 - \frac{z_t}{K} \right)
\]

where \( \rho \) is a growth rate and \( K \) is the carrying capacity of the environment.

The objective is to maximize

\[
\sum_{t=0}^{\infty} \beta^t z_t x_t
\]
AN EXAMPLE OF PRIMAL EQUILIBRIUM

If \( \rho \) is random, we may consider scenarios \( \xi_t^s \) for that parameter and make the problem stochastic.

To keep problem size under control, we need to limit the planning horizon \( T \). However, we do not want \( X_T = 1 \). So, we look for a suitable terminal value function \( Q(Z_{T+1}) \).

Let us consider an average growth rate \( \bar{\xi} \) and assume that, after \( T \), we catch a fraction such that the population is stable:

\[
x_t = \bar{\xi} \left( 1 - \frac{Z_{T+1}}{K} \right), \quad t \geq T + 1
\]

Then, using the basic property of the geometric series we find:

\[
Q(Z_{T+1}) = \sum_{t=T+1}^{\infty} \beta^{t-T-1} x_t z_t = \bar{\xi} Z_{T+1} \frac{(1 - Z_{T+1}/K)}{1 - \beta}
\]

By adding this term to the finite horizon cost, we obtain a nonlinear stochastic programming model that can be tackled, e.g., by progressive hedging.
Sometimes, we may also consider adding constraints on terminal state variables. For instance, we may use actuarial approaches to set a minimal target wealth ensuring solvency of a pension fund.

Infinite time horizon dynamic programming may also be used to find a suitable value function (see Lecture 8).

See also

- (King, Wallace, 2012) for an example of dual equilibrium in production planning,
- (Cariño et al., 1998) for a discussion of end effects in financial planning.