Optimization of CIS

ECE 802

Tutorial 4 – Mixed Integer Programming

L. Tziovani and A. Astolfi

A wide class of practical problems can be modelled using integer variables and linear constraints. Sometimes such problems involve only integer variables hence it is a pure integer programming (PIP). More often, both continuous variables and integer variables are present, hence the problem is said to be a mixed integer programming (MIP) problem.

<u>Applications of integer programming</u>

1. There are situations in which it is only meaningful to consider integer quantities of certain goods, for example, cars, aeroplanes or houses. Therefore, the integer variables represent quantities that can only be integer (we cannot sell 1.5 car). In these cases we might use an integer programming (IP) instead of a linear programming (LP).
2. There are situations in which the integer variables are used to represent decisions. More specifically, in these situations the integer variables are restricted to take two values, 0 or 1. Such binary variables are used to represent 'yes or no' decisions. For example, in the unit commitment problem the binary variables are used to represent when a unit is ON (binary variable = 1) and when a unit is OFF (binary variable = 0). Finally, in applications using graphs the binary variables can be used to decide whether an edge in a graph should be included.
3. Sometimes functions are not continuous: in such a case the binary variables are used as indicator variables.
4. Sometimes it is necessary to represent logical connections between different decisions or states using linear constraints involving indicator variables (binary variables). For example, if either of products A or B (or both) are manufactured, then at least one of products C, D or E must also be manufactured.
5. Mixed Integer linear programming can be used to approximate non-convex functions using piece-wise linear functions.

<u>Advantages</u>

Integer programming can be used to non-linear and non-convex problems where the global optimal solution can be computed using a modern computer and an optimization solver. Most of these solvers use either the branch and bound method, or the cutting plane method to solve these problems.

<u>Disadvantages</u>

While integer programming can be used in a wide set of applications, the main drawback of this method is the computation time that is needed to compute an optimal solution. In the case of linear programming, a large problem with thousands variables and constraints can be solved in reasonable time, however in the case of integer programming there is a possibility to fail to find a solution in a reasonable time. This is due to the fact that integer programming problems are NP-hard problems, that are problems which cannot be solved in polynomial time. As the size of the problem increases, then the computation increases exponentially.

A problem with integral quantities. Consider the optimization problem:

$$\max f = x_1 + x_2,$$

subject to

$$-2x_1 + 2x_2 \geq 1,$$

$$-8x_1 + 10x_2 \leq 13,$$

$$x_1, x_2 \geq 0.$$

Solving the problem using linear programming yields the solution

$$x_1 = 4, x_2 = 4.5.$$

Solving the problem using integer programming yields the solution

$$x_1 = 1, x_2 = 2.$$

The solution using the linear programming is correct if the variables represent continuous quantities, for example representing certain goods (e.g., gallons of beer). However, the solution of the integer programming problem is correct if the variables represent integral quantities of certain goods (e.g., aeroplanes).

A problem with decision variables (binary variables). Suppose we wish to invest \$14,000. We have identified four investment opportunities. Investment 1 requires an investment of \$5,000 and has a present value (a time-discounted value) of \$8,000; Investment 2 requires \$7,000 and has a value of \$11,000; Investment 3 requires \$4,000 and has a value of \$6,000; and Investment 4 requires \$3,000 and has a value of \$4,000. How to invest to maximize our total present value? The integer programming formulation is

$$\max f = 8b_1 + 11b_2 + 6b_3 + 4b_4,$$

subject to

$$5b_1 + 7b_2 + 4b_3 + 3b_4 \leq 14.$$

$$b_j \in \{0, 1\}\, j = 1, ..4.$$

Solving the problem as an integer programming problem yields the solution

$$b_1 = 0, b_2 = b_3 = b_4 = 1, f = 21{,}000$$

Solving the problem as a linear programming problem (the variables can take any value between 0 and 1) yields the solution

$$b_1 = 1, b_2 = 1, b_3 = 0.5, b_4 = 0, f = 22{,}000$$

Unfortunately this solution is not integral. Rounding b_3 up to 1 gives an infeasible solution. Rounding x_3 down to 0 gives a feasible solution with a value of 19,000 which is not the optimal value. Therefore, this example shows that rounding does not necessarily gives an optimal value.

A problem with decision variables and logical restrictions. Consider the previous investment problem in which the following constraints are added:

1. we can only make two investments;

2. if Investment 2 is made, Investment 4 must also be made.

The problem formulation is

$$\max f = 8b_1 + 11b_2 + 6b_3 + 4b_4,$$

subject to

$$5b_1 + 7b_2 + 4b_3 + 3b_4 \leq 14,$$

$$b_1 + b_2 + b_3 + b_4 \leq 2,$$

$$b_2 - b_4 \leq 0,$$

$$b_j \in \{0, 1\}\, j = 1, ..4$$

<u>Modelling logical conditions</u>

Notation from Boolean algebra:

" $\vee$ " means "or" (this is inclusive, i.e. A or B or both).
"." means "and".
" $\sim$ " means "not" .
" $\rightarrow$ " means "implies" (or "if ...then").
" $\leftrightarrow$ " means "if and only if".

Relation between logical conditions and indicator variables:

We use X_i for the proposition $b_i = 1$, where the variable b_i is a 0-1 indicator variable. The following propositions and constraints can easily be seen to be equivalent:

$X_1 \vee X_2$ is equivalent to $b_1 + b_2 \geq 1$
$X_1 . X_2$ is equivalent to $b_1 = 1, b_2 = 1$
$\sim X_1$ is equivalent to $b_1 = 0$ $(or\ \ 1 - b_1 = 1)$
$X_1 \rightarrow X_2$ is equivalent to $b_1 - b_2 \leq 0$
$X_1 \leftrightarrow X_2$ is equivalent to $b_1 - b_2 = 0$

Example: If either of the products A or B (or both) are manufactured, then at least one of the products C, D or E must be manufactured. Let X_i stand for the proposition 'Product i is manufactured' (i is A, B, C, D or E). We wish to impose the logical condition

$$(X_A \vee X_B) \rightarrow (X_C \vee X_D \vee X_E)$$

Indicator variables are introduced to perform the following functions: $b_i = 1$ if and only if product i is manufactured; $\hat{b} = 1$ if the proposition $(X_A \vee X_B)$ holds. With this notation the proposition $(X_A \vee X_B)$ can be represented by the inequality

$$b_1 + b_2 \geq 1.$$

The proposition $(X_C \vee X_D \vee X_E)$ can be represented by the inequality

$$b_1 + b_2 + b_3 \geq 1.$$

Firstly, we impose the following condition:

$$b_1 + b_2 \geq 1 \rightarrow \hat{b} = 1,$$

which is equivalent to

$$b_1 + b_2 - 2\hat{b} \leq 0.$$

Secondly, we impose the condition

$$\hat{b} = 1 \rightarrow b_1 + b_2 + b_3 \geq 1,$$

which is equivalent to

$$\hat{b} - b_1 - b_2 - b_3 \leq 0.$$

<u>Modelling non-linear constraints</u>. Consider a constraint that includes an absolute value function. Assume that the function is linear. Then the absolute value can be formulated with two linear expressions, that is

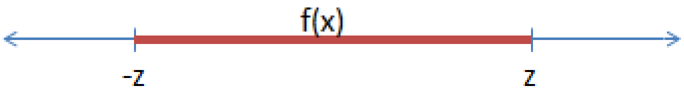
$$|f(x)| \leq z$$

can be reformulated as

$$f(x) \leq z,$$

$$-f(x) \leq z \ \ (or\ f(x) \geq -z)$$

The figure below shows the geometric representation, where the absolute value function as well as the two linear functions described above, demonstrating that the two formulations are equivalent. This constraint is convex and such a problem can be formulated as a linear programming problem.



<u>Modelling non-linear and non-convex constraints</u>. Consider again a constraint that includes an absolute value function. Assume that the function is linear. Then the absolute value can be formulated with two linear expressions, that is

$$|f(x)| \geq z$$

can be reformulated as

$$f(x) \geq z$$

$$-f(x) \geq z \ \ (or\ f(x) \leq -z)$$

The geometric representation is given in the figure below.

In this case a discontinuity is present. As a result, the two constraints cannot be satisfied together if we formulate the problem as linear programming. However, this non-convex constraint can be formulated using integer programming and considering the constraints

$$f(x) + Mb \geq z,$$

$$-f(x) + M(1-b) \geq z,$$

where M is a sufficiently enough constant and b is a binary variable. If $b = 0$ then the first constraint is active and the second constraint is always fulfilled. If $b = 1$ then the second constraint is active and the first constraint is always fulfilled.

<u>Solving integer programming problems using the Gurobi solver</u>: The Gurobi Optimizer is a commercial optimization solver for linear programming, quadratic programming, quadratically constrained programming, mixed integer linear programming, mixed-integer quadratic programming, and mixed-integer quadratically constrained programming. The academic license is free and Gurobi can be used in Matlab. For more information read the Gurobi optimizer guide (pages 6-10 and 80-83), at

https://www.gurobi.com/wp-content/plugins/hd_documentations/content/pdf/quickstart_windows_8.1.pdf

Example: Consider the problem

$$\min f = 2x_1 + 3x_2 + 5b,$$

subject to

$$x_1 + x_2 + 3b \geq 5,$$

where x_1 is a positive continuous variable; x_2 is a positive integer variable and b is a binary variable.

```
%% ***************************
clear all;
clc;
%Variables: x1, x2, b
%objective function
f=[2 3 5 ];
%bounds
lb=[0 0 0];
ub=[inf inf 1];
%Equality constraints (There are no equality constraints)
Aeq=[];
beq=[];
%Inequality constraints
A=[-1 -1 -3];
b=-5;
%Declare the integer variables. Declare their position in the variables set
 integer_variables=[2,3];
```

```
%call the Gurobi solver
model.obj =f; %objective function
%converts a full matrix into sparse.
A_sparse=sparse(A);
Aeq_sparse=sparse(Aeq);
%constraints
 model.A = [A_sparse; Aeq_sparse];
 model.rhs = [b; beq];
 model.sense = [repmat('<', size(A,1), 1); repmat('=', size(Aeq,1), 1)];
 model.lb = lb;
 model.ub = ub;
 n = size(A, 2);
 model.vtype = repmat('C', n, 1);
 model.vtype(integer_variables) = 'I';
  %call the solver
 result = gurobi(model);
 solution=result.x;
```

Solution: x=[2, 0], b=1.

Homework: Formulate the following problems as linear programming problems or as mixed integer linear programming problems.

a)

$$\min f = |x_1| + 3x_2$$

subject to

$$ax_1 + bx_2 \geq c.$$

b)

$$\max f = |x_1| + 3x_2$$

subject to

$$ax_1 + bx_2 \leq c.$$