OR Methods in Channel Coding and Code Design

Marco Colombo

School of Mathematics
University of Edinburgh

Edinburgh, 24 April 2008

Outline

Problem statement

A codeword $c \in \mathcal{C}$ is sent over a noisy, memoryless channel.
A vector y is received.

Decoding problem:
Find the most likely codeword c given the vector y.

Decoding by Linear Programming

For binary LDPC codes, Feldman, Wainwright and Karger (2005) showed that the decoding problem can be posed as a linear programming problem.

Great news from the OR perspective!

- Analytical tools to understand the problem
- Numerical tools to solve it

An important bridge between Engineering–OR communities.

Maximum likelihood principle

Maximum likelihood codeword:

$$\hat{c} = \arg\max_{c \in \mathcal{C}} P(c|y)$$

Maximum likelihood principle

Maximum likelihood codeword:

$$\hat{c} = \arg\max_{c \in \mathcal{C}} P(c|y)$$

Consider a negative log-likelihood ratio as a cost function:

$$\gamma_i = \log\left(\frac{P(y_i|c_i = 0)}{P(y_i|c_i = 1)}\right)$$

Maximum likelihood principle

Maximum likelihood codeword:

$$\hat{c} = \arg\max_{c \in \mathcal{C}} P(c|y)$$

Consider a negative log-likelihood ratio as a cost function:

$$\gamma_i = \log\left(\frac{P(y_i|c_i = 0)}{P(y_i|c_i = 1)}\right)$$

The problem can be stated as:

$$\min \sum_{i=1}^{n} \gamma_i c_i \quad \text{s.t. } c \in \mathcal{C}$$

The codeword polytope

Convex hull of all possible codewords:

$$\mathcal{P} = \left\{ \sum_{c \in \mathcal{C}} \lambda_c c : \ \lambda_c \geq 0, \ \sum_{c \in \mathcal{C}} \lambda_c = 1 \right\}$$

The codeword polytope

Convex hull of all possible codewords:

$$\mathcal{P} = \left\{ \sum_{c \in \mathcal{C}} \lambda_c c : \ \lambda_c \geq 0, \ \sum_{c \in \mathcal{C}} \lambda_c = 1 \right\}$$

Fundamental theorem of linear programming:
For a linear programming problem with a feasible domain $\mathcal{P}$ containing at least one extreme point, the optimal objective value is either unbounded or is achievable at one extreme point of $\mathcal{P}$.

Cleverly consider only a subset of points.

Again, that combinatorial explosion...

Linear inequality description:
A full dimensional polyhedron $\mathcal{P}$ has a unique minimal representation by a finite set of linear inequalities:

$$Ax \leq b.$$

Again, that combinatorial explosion...

Linear inequality description:
A full dimensional polyhedron $\mathcal{P}$ has a unique minimal representation by a finite set of linear inequalities:

$$Ax \leq b.$$

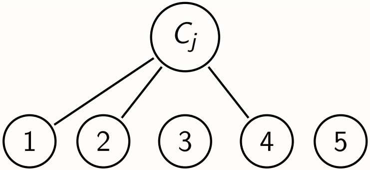
However...

- The number of inequalities is exponential in the code length
- Optimally decoding a LDPC code is NP-Complete.

Consider a relaxation.

Local codeword polytope

Consider one check node C_j:



Valid configurations for C_j:

$$S_j = \Big\{\emptyset, \{1,2\}, \{1,4\}, \{2,4\}\Big\}$$

Introduce a variable $w_{j,s}$ for each valid configuration:

$$\sum_{s \in S_j} w_{j,s} = 1, \qquad 0 \leq w_{j,s} \leq 1 \qquad (\star)$$

Linear programming relaxation

Introduce variables f_i for each code bit y_i:

$$f_i = \sum_{s \in S_j, s \ni i} w_{j,s}, \qquad 0 \leq f_i \leq 1 \qquad (\star\star)$$

Linear programming relaxation

Introduce variables f_i for each code bit y_i:

$$f_i = \sum_{s \in S_j, s \ni i} w_{j,s}, \qquad 0 \leq f_i \leq 1 \qquad (\star\star)$$

Define the polytope $\mathcal{Q} = \{(\star) + (\star\star)\}$:

$$\min \sum_{i=1}^{n} \gamma_i f_i \quad \text{s.t. } (f, w) \in \mathcal{Q}$$

OR perspectives on relaxation

The relaxation can produce fractional solutions.

Mathematically we can provide ways for recovering from failures:

- Strengthen the relaxation (cutting planes, combinatorial optimization)
- Approaches based on branching
- Nonconvex quadratic formulation

Cutting planes

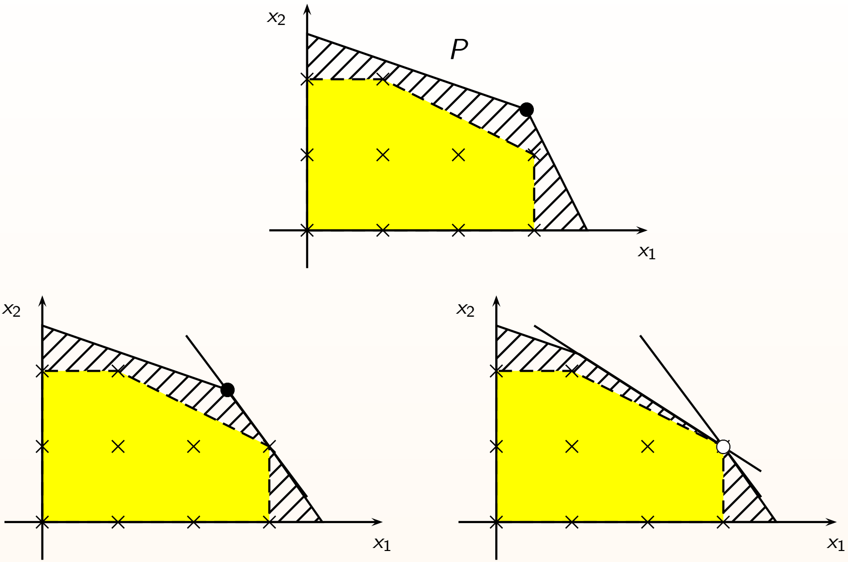
A cutting plane is a constraint with these two properties:

- Any feasible integer point will satisfy the cut.
- The optimal solution of the current linear programming relaxation will violate the cut.

This can be embedded in an iterative algorithm:

- Solve the LP relaxation of the integer problem.
- If the optimal solution is integer, it solves the IP as well.
- Generate a cutting plane and add it to the constraints.

Example of cutting planes



Branch-and-cut

Divide the problem into smaller problems until these yield an integral solution.

Disjunctive cutting planes are added to the problem. Each node in the branching tree generates two sons:

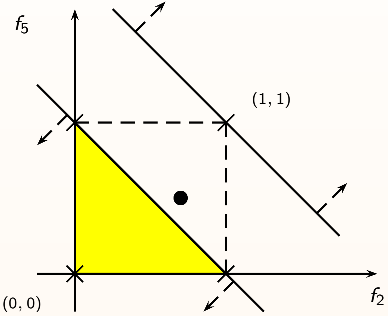
$$\mathcal{Q} \cap h_1, \qquad h_1 = a^T x \leq b - 1$$

$$\mathcal{Q} \cap h_2, \qquad h_2 = a^T x \geq b$$

Example for BIP

Suppose the current solution contains (among others) these two fractional variables:

$$f_2 = 0.7 \qquad f_5 = 0.5$$

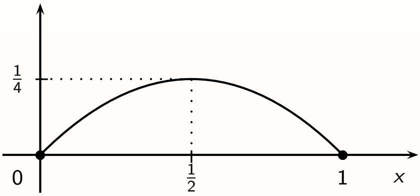


Introduce two cuts:

$f_2 + f_5 \leq 1$ and $f_2 + f_5 \geq 2$.

Nonconvex QP approach

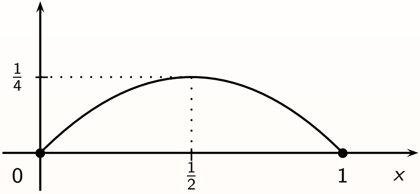
Express a continuous variable $0 \leq x \leq 1$ such that $x(1-x) = 0$.



In the interval $[0, 1]$ the function is non-negative, and attains its minimum at the extreme points, which have integer coordinates.

Nonconvex QP approach

Express a continuous variable $0 \leq x \leq 1$ such that $x(1-x) = 0$.



In the interval $[0, 1]$ the function is non-negative, and attains its minimum at the extreme points, which have integer coordinates.

Issues:

- This setup is nonconvex: not possible to use the simplex.
- Interior point methods struggle if the nonconvexity is large.

Exploiting the nonconvex QP approach

An integral solution satisfies:

$$f_i(1 - f_i) = 0, \qquad i = 1, \ldots, n$$

Consider the following quantity:

$$\sum_i f_i(1 - f_i) \geq 0$$

- For a fractional solution, the term is positive.
- For an integral solution, the term in zero.

Exploiting the nonconvex QP approach

An integral solution satisfies:

$$f_i(1 - f_i) = 0, \qquad i = 1, \ldots, n$$

Consider the following quantity:

$$\sum_i f_i(1 - f_i) \geq 0$$

- For a fractional solution, the term is positive.
- For an integral solution, the term in zero.

This term can be used as a penalty term in the objective:

$$\min \sum_i \lambda_i f_i + \alpha \sum_i f_i(1 - f_i)$$

The 3 S's in linear programming

The practical efficiency for a solver is affected by these considerations regarding the constraint matrix:

- Size
- Sparsity
- Structure

The 3 S's in linear programming

The practical efficiency for a solver is affected by these considerations regarding the constraint matrix:

- Size
- Sparsity
- Structure

These can affect the choice of solution methods:

- Simplex method
- Interior point methods
- Specialized algorithms

Checklist on solution methods

Simplex method:

- Suited for linear programming
- Explore the vertices of the polytope
- Polynomial complexity in practice
- Easy to warmstart
- Difficult to parallelize

Checklist on solution methods

Simplex method:

- Suited for linear programming
- Explore the vertices of the polytope
- Polynomial complexity in practice
- Easy to warmstart
- Difficult to parallelize

Interior point methods:

- Suited for linear, quadratic, nonlinear programming
- Move in the interior of the polytope
- Polynomial complexity in practice and theory
- Difficult to warmstart
- Parallel implementation and structure exploitation

Conclusions

An important bridge between Engineering and OR:

- For OR: interesting application with peculiar requirements
- For Engineering: exploring the available tools and techniques
- More work to be done!