Advanced linear programming

Lagrangean relaxation (ILP duality)

Marjan van den Akker

Universiteit Utrecht

[Faculty of Science
Information and Computing Sciences]

Lagrangean relaxation

- can be viewed as ILP duality
- gives a bound at least as good as the linear programming relaxation
- Today:
  - Strength of Lagrangean dual
  - Knapsack example

Lagrangean relaxation

`Nasty' constraints

$$Z_{IP}: \min\ cx$$
$$Ax \geq b$$
$$Dx \geq d$$
$$x \text{ integer}$$

Lagrangean subproblem

$$p \geq 0:$$
$$Z(p): \min cx + p(b - Ax)$$
$$Dx \geq d$$
$$x \text{ integer}$$

You may violate the constraints but this has cost

We do a minimization problem!

$$Z(p) \leq Z_{IP}$$

$$Z_{LP} = \min\{cx \mid Ax \geq b, Dx \geq d\}$$

Lagrangean relaxation (2)

Z_D : max $Z(p)$

$p \geq 0$

Lagrangean dual

$X = \{x \mid Dx \geq d,\ x \text{ integer}\}$

Theorem: Z_D is equal to

Corollary: $Z_{LP} \leq Z_D \leq Z_{IP}$

min cx

$Ax \geq b$

$x \in \text{CH}(X)$ (i.e. convex hull of X)

Corollary:

if $CH(x) = \{x \mid Dx \geq d\}$ (integrality redundant in subproblem), then $Z_D = Z_{LP}$

In general Z_D can be solved by the subgradient method.

Knapsack problem

$$\max \sum_{j=1}^{n} c_j x_j$$

$$s.t. \sum_{j=1}^{n} a_j x_j \le B$$

$$x_j \in \{0,1\}$$

We now do a maximization problem!

$$Z_{IP} \le Z_D \le Z_{LP}$$

$$\text{actually}, Z_D = Z_{LP}$$

We can check this.

Let $\lambda \ge 0$:

$$L(\lambda) = \max \sum_{j=1}^{n} c_j x_j + \lambda \left(B - \sum_{j=1}^{n} a_j x_j\right)$$

$$x_j \in \{0,1\}$$

$$Z_{IP} \le L(\lambda)$$

Lagrangean dual (best bound):

$$Z_D = \min_{\lambda \ge 0} L(\lambda)$$

Wrap up

- ILP models many combinatorial optimization problems (Ch 10)
- ILP solved by algorithms with LP-relaxations as subproblems
- If constraint matrix TUM: LP-relaxation solves problem
- In general: use branch-and-bound with LP-relaxation as bound (Ch 11.2)
- LP-relaxation can be strengthened by cutting planes
- Results in branch-and-cut:
  - Framework algorithm, many features have to be included
  - This algorithm in used by well-known solvers like CPLEX and Gurobi, GLPK.

Wrap up

Dealing with very large models:

- Decomposition on LP:
  - Column generation = Dantzig Wollfe decomposition = cutting planes in dual (Ch 6.1, 6.2, 6.3, 6.4)
    - *Reduces the number of constraints at the cost of a very large number of variables, but you only add the necessary ones*
  - Benders decomposition = Dantzig Wolfe decomposition in dual (Ch 6.5)
    - *Reduces the number of variables at the cost of a very large number of constraints, but you only add the necessary ones*
- Decomposition on ILP:
  - Lagrangean relaxation (Ch 11.4)