

Program of the day:

- Surrogate relaxation
- Subgradient optimization for Lagrange multipliers
- A clue on solving LP problems without Simplex
- Applications: Manpower planning

1

Relaxation

In a branch-and-bound algorithm we find upper bounds by relaxing the problem

Relaxation (Wolsey sec. 2.1)

$$\begin{aligned} \max\{cx : x \in S\} & \quad (IP) \\ \max\{f(x) : x \in T\} & \quad (RP) \end{aligned}$$

RP is a relaxation of IP if

- $S \subseteq T$
- $f(x) \geq cx$ for all $x \in S$

Which constraints should be relaxed?

- Quality of bound (tightness of relaxation)
- Remaining problem can be solved efficiently
- Proper multipliers can be found efficiently
- Constraints difficult to formulate mathematically
- Constraints which are too expensive to write up

2

Overview

Different relaxations

- LP-relaxation
- Deleting constraint
- Lagrange relaxation
- Surrogate relaxation
- Semidefinite relaxation

Relaxations are often used in combination.

Hierarchy

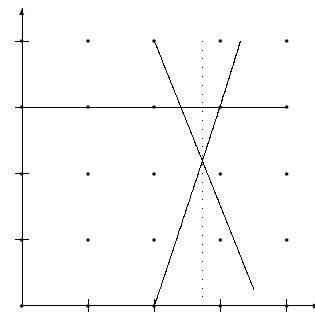
tighter

- Best surrogate relaxation
- Best Lagrangian relaxation
- LP relaxation

3

Surrogate relaxation, example

$$\begin{aligned} \text{maximize} & \quad 4x_1 + x_2 \\ \text{subject to} & \quad 3x_1 - x_2 \leq 6 \\ & \quad x_2 \leq 3 \\ & \quad 5x_1 + 2x_2 \leq 18 \\ & \quad x_1, x_2 \geq 0, \text{ integer} \end{aligned}$$



IP solution $(x_1, x_2) = (2, 3)$ with $z_{IP} = 11$
 LP solution $(x_1, x_2) = (\frac{30}{11}, \frac{24}{11})$ with $z_{LP} = \frac{144}{11} = 13.1$

First and third constraint complicating, surrogate relax using multipliers $\lambda_1 = 2$, and $\lambda_3 = 1$

$$\begin{aligned} \text{maximize} & \quad 4x_1 + x_2 \\ \text{subject to} & \quad x_2 \leq 3 \\ & \quad 11x_1 \leq 30 \\ & \quad x_1, x_2 \geq 0, \text{ integer} \end{aligned}$$

Solution $(x_1, x_2) = (2, 3)$ with $z_{SR} = 4 \cdot 2 + 3 = 11$
 Upper bound

4

Surrogate relaxation

Integer Programming Problem

$$\begin{aligned} & \text{maximize } cx \\ & \text{subject to } Ax \leq b \\ & \quad Dx \leq d \\ & \quad x_j \in \mathbb{Z}_+, \quad j = 1, \dots, n \end{aligned}$$

Surrogate relax $Dx \leq d$, using multipliers $\lambda \geq 0$, i.e. add together constraints using weights λ

$$\begin{aligned} & \text{maximize } z_{SR}(\lambda) = cx \\ & \text{subject to } Ax \leq b \\ & \quad \lambda Dx \leq \lambda d \\ & \quad x_j \in \mathbb{Z}_+, \quad j = 1, \dots, n \end{aligned}$$

Proposition 1 Optimal solution to relaxed problem gives upper bound on original problem

Proof show that relaxation

multiplier λ_i is “weighting” of constraint
 If λ_i large \Rightarrow constraint satisfied (weakening other constraints)
 If $\lambda_i = 0 \Rightarrow$ drop constraint

5

Surrogate relaxation

Surrogate relaxed problem as function of $\lambda \geq 0$

$$\begin{aligned} & \text{maximize } z_{SR}(\lambda) = cx \\ & \text{subject to } Ax \leq b \\ & \quad \lambda Dx \leq \lambda d \\ & \quad x_j \in \mathbb{Z}_+, \quad j = 1, \dots, n \end{aligned}$$

Surrogate Dual Problem

$$z_{SD} = \min_{\lambda \geq 0} z_{SR}(\lambda)$$

Natural questions:

- How do we find best λ ?
- How tight is relaxation?

Not as nice properties as Lagrange relaxation since relaxed constraints are not linearized, and hence knowledge from LP cannot be used

6

Surrogate relaxation

Surrogate relaxation (as well as all other relaxations) has the following property

$$\begin{aligned} & \text{maximize } z_{SR}(\lambda) = cx \\ & \text{subject to } Ax \leq b \\ & \quad \lambda Dx \leq \lambda d \\ & \quad x_j \in \mathbb{Z}_+, \quad j = 1, \dots, n \end{aligned}$$

If solution to z_{SR} is feasible to original problem, then

$$\bar{z} = \underline{z} = z_{SR}$$

hence problem solved to optimality

Surrogate relaxation (example)

If we surrogate relax all constraints of a BIP, then we obtain a Knapsack Problem.

$$\begin{aligned} & \text{maximize } \sum_{j=1}^n p_j x_j \\ & \text{subject to } \sum_{j=1}^n w_j x_j \leq c \\ & \quad x_j \in \{0, 1\}, \quad j = 1, \dots, n, \end{aligned}$$

7

Tightness of relaxation

$$\begin{aligned} & \text{maximize } cx \\ & \text{subject to } Ax \leq b \\ & \quad Dx \leq d \\ & \quad x_j \in \mathbb{Z}_+, \quad j = 1, \dots, n \end{aligned}$$

$$\max \left\{ cx : x \in \text{conv}(Ax \leq b, Dx \leq d, x \in \mathbb{Z}_+) \right\}$$

Lagrange Relaxation, best multipliers $\lambda \geq 0$

$$\begin{aligned} & \text{maximize } z_{LR}(\lambda) = cx - \lambda(Dx - d) \\ & \text{subject to } Ax \leq b \\ & \quad x_j \in \mathbb{Z}_+, \quad j = 1, \dots, n \end{aligned}$$

$$\max \left\{ cx : Dx \leq d, x \in \text{conv}(Ax \leq b, x \in \mathbb{Z}_+) \right\}$$

Surrogate Relaxation, best multipliers $\lambda \geq 0$

$$\begin{aligned} & \text{maximize } z_{SR}(\lambda) = cx \\ & \text{subject to } Ax \leq b \\ & \quad \lambda Dx \leq \lambda d \\ & \quad x_j \in \mathbb{Z}_+, \quad j = 1, \dots, n \end{aligned}$$

$$\max \left\{ cx : x \in \text{conv}(Ax \leq b, \lambda Dx \leq \lambda d, x \in \mathbb{Z}_+) \right\}$$

Best surrogate relax. is tighter than best Lagrange relax.

8

Relaxation strategies

Which constraints should be relaxed

- "the complicating ones"
- remaining problem is polynomially solvable (e.g. min spanning tree, assignment problem, linear programming)
- remaining problem is totally unimodular (e.g. network problems)
- remaining problem is NP-hard but good techniques exist (e.g. knapsack)
- constraints which cannot be expressed in MIP terms (e.g. cutting)
- constraints which are too extensive to express (e.g. subtour elimination in TSP)

9

Subgradient optimization Lagrange multipliers

(Similar technique can be used for surrogate multipliers)

$$\begin{aligned} & \text{maximize } cx \\ & \text{subject to } Ax \leq b \\ & \quad Dx \leq d \\ & \quad x_j \in \mathbb{Z}_+, \quad j = 1, \dots, n \end{aligned}$$

Lagrange Relaxation, multipliers $\lambda \geq 0$

$$\begin{aligned} & \text{maximize } z_{LR}(\lambda) = cx - \lambda(Dx - d) \\ & \text{subject to } Ax \leq b \\ & \quad x_j \in \mathbb{Z}_+, \quad j = 1, \dots, n \end{aligned}$$

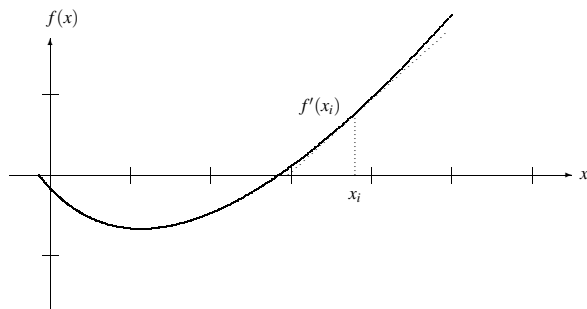
Lagrange Dual Problem

$$z_{LD} = \min_{\lambda \geq 0} z_{LR}(\lambda)$$

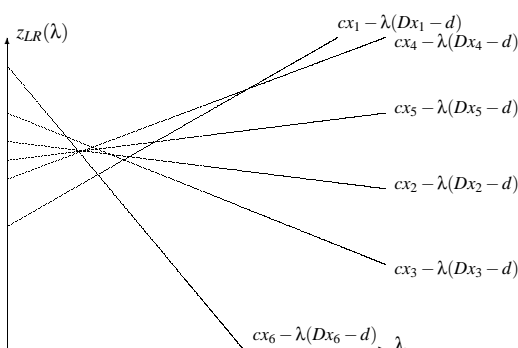
- We do not need best multipliers in B&B algorithm
- Subgradient optimization fast method
- Methods for minimizing a convex, possibly non-differentiable function over a convex and closed set
- Roots in nonlinear programming, Held and Karp (1971)

10

Subgradient optimization, motivation



Lagrange function $z_{LR}(\lambda)$ is piecewise linear and convex



11

Subgradient

Generalization of gradients to non-differentiable functions.

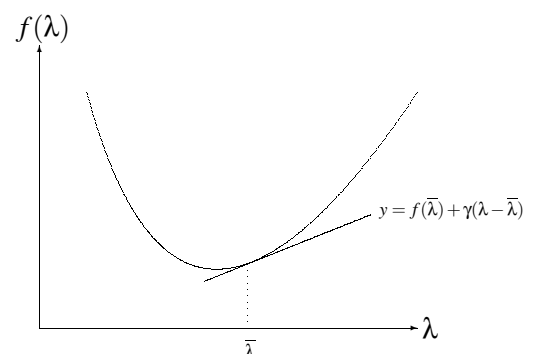
Definition 1 An m -vector γ is subgradient of $f(\lambda)$ at $\lambda = \bar{\lambda}$ if

$$f(\lambda) \geq f(\bar{\lambda}) + \gamma(\lambda - \bar{\lambda}) \quad (1)$$

The inequality says that the hyperplane

$$y = f(\bar{\lambda}) + \gamma(\lambda - \bar{\lambda})$$

is tangent to $y = f(\lambda)$ at $\lambda = \bar{\lambda}$ and supports $f(\lambda)$ from below.



12

Proposition 2 Given a choice of nonnegative multipliers $\bar{\lambda}$. If x' is an optimal solution to $z_{LR}(\bar{\lambda})$ then

$$\gamma = d - Dx'$$

is a subgradient of $z_{LR}(\lambda)$ at $\lambda = \bar{\lambda}$.

Proof We wish to prove (1) which in our version is:

$$\max_{Ax \leq b} (cx - \lambda(Dx - d)) \geq \gamma(\lambda - \bar{\lambda}) + \max_{Ax \leq b} (cx - \bar{\lambda}(Dx - d))$$

where x' is an opt. solution to the right-most subproblem
Inserting γ we get:

$$\begin{aligned} \max_{Ax \leq b} (cx - \lambda(Dx - d)) &\geq (d - Dx')(\lambda - \bar{\lambda}) + (cx' - \bar{\lambda}(Dx' - d)) \\ &= cx' - \lambda(Dx' - d) \end{aligned}$$

□

Proposition 3 Optimality condition. If the function $f(\lambda)$ is convex and a subgradient $\gamma = 0$ exists in $\lambda = \bar{\lambda}$ then $\bar{\lambda}$ is optimal.

13

Intuition

Lagrange Relaxation

$$\begin{aligned} &\text{maximize } z_{LR}(\lambda) = cx - \lambda(Dx - d) \\ &\text{subject to } Ax \leq b \\ &\quad x_j \in \mathbb{Z}_+, \quad j = 1, \dots, n \end{aligned}$$

Gradient in x' is

$$\gamma = d - Dx'$$

Subgradient iteration

Recursion

$$\lambda^{(k+1)} = \max \left\{ \lambda^{(k)} - \theta \gamma^{(k)}, 0 \right\}$$

Where $\theta > 0$ is step-size.

If $\gamma > 0$ and θ is sufficiently small $z_{LR}(\lambda)$ will decrease.

- Small θ slow convergence.
- Large θ unstable

14

Held and Karp

Initially

$$\lambda^{(0)} = \{0, \dots, 0\}$$

compute the new multipliers by recursion

$$\lambda_i^{(k+1)} := \begin{cases} \lambda_i^{(k)} & \text{if } |\gamma_i| \leq \epsilon \\ \max(\lambda_i^{(k)} - \theta \gamma_i, 0) & \text{if } |\gamma_i| > \epsilon \end{cases}$$

where γ is subgradient.

The step size θ is defined by

$$\theta = \mu \frac{\bar{z} - z}{\sum_i \gamma_i^2}$$

where μ is an appropriate constant.

E.g. $\mu = 1$ and halved if upper bound not decreased in 20 iterations

15

Example: Manpower planning

One of the most successful applications of OR

Cover a number of job functions using least possible resources

Constraints difficult to formulate

- Air-crew scheduling
- Hospital-crew scheduling
- Supermarket-crew scheduling

Model is so general that it can handle

- Assignment of teachers to classes/rooms
- Planning of transportation

Set-covering model

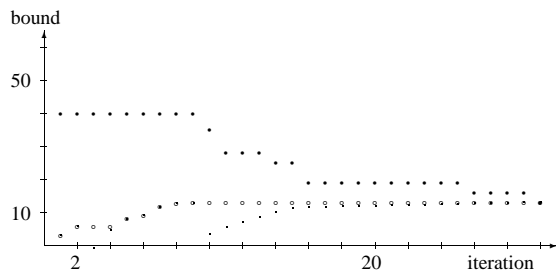
$$\begin{aligned} &\text{minimize } \sum_{j=1}^n c_j x_j \\ &\text{subject to } \sum_{j=1}^n a_{ij} x_j \geq 1 \\ &\quad x_j \in \{0, 1\} \end{aligned}$$

where $a_{ij} = 1$ iff job i is covered by job-schedule j , and c_j is the cost of job-schedule j .

16

Lagrange Relaxation

Development of \bar{z} and \underline{z} .



After 30 iterations we have $\underline{z} = \bar{z} = 13$

$$\lambda = (1.85, 0, 2.77, 0, 1.13, 2.99, 6.11, 0)$$

Dual variables

$$y = (0, 0, 2, 0, 1, 3, 7, 0)$$

Lagrange relaxation and LP

For an LP-problem where we Lagrange relax all constraints

- Dual variables are best choice of Lagrange multipliers
- Lagrange relaxation and LP “relaxation” give same bound

Gives a clue to solve LP-problems without Simplex

- Iterative algorithms
- Polynomial algorithms