Introduction to Optimization, DIKU 2007
Tuesday, March 20
David Pisinger

Program of the day:

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

Relaxation

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

Relaxation (Wolsey sec. 2.1)

$$\max\{cx : x \in S\} \qquad (IP)$$
$$\max\{f(x) : x \in T\} \qquad (RP)$$

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

Overview

Different relaxations

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

Relaxations are often used in combination.

Hierarchy

tighter ↑

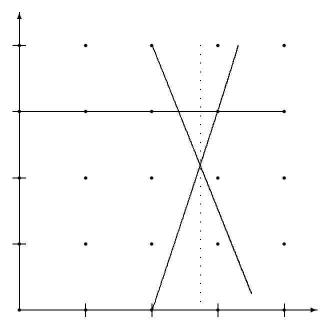
Best surrogate relaxation

Best Lagrangian relaxation

LP relaxation

Surrogate relaxation, example

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



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 $\lambda_3 = 1$

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

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

Surrogate relaxation

Integer Programming Problem

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

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

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

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

Surrogate relaxation

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

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

Surrogate Dual Problem

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

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

Surrogate relaxation

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

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

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.

$$\begin{array}{ll} \text{maximize} & \sum_{j=1}^{n} p_j x_j \\ \text{subject to} & \sum_{j=1}^{n} w_j x_j \le c \\ & x_j \in \{0,1\}, \quad j = 1, \ldots, n, \end{array}$$

Tightness of relaxation

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

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

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

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

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

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

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

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

Best surrogate relax. is tighter than best Lagrange relax.

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)

Subgradient optimization Lagrange multipliers

(Similar technique can be used for surrogate multipliers)

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

Lagrange Relaxation, multipliers $\lambda \geq 0$

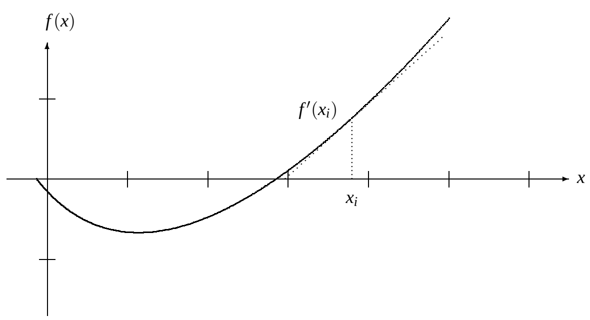
$$\begin{array}{ll} \text{maximize} & z_{LR}(\lambda) = cx - \lambda(Dx - d) \\ \text{subject to} & Ax \leq b \\ & x_j \in \mathbb{Z}_+, \quad j = 1, \ldots, n \end{array}$$

Lagrange Dual Problem

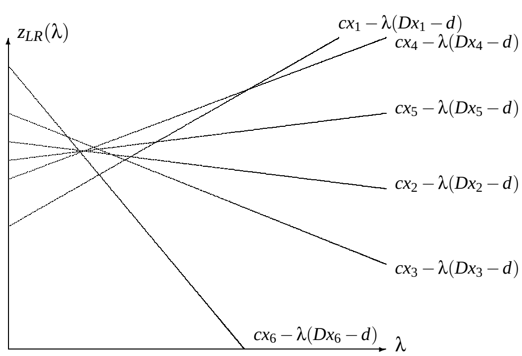
$$z_{LD} = \min_{\lambda \geq 0} z_{LR}(\lambda)$$

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

Subgradient optimization, motivation



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



Subgradient

Generalization of gradients to non-differentiable functions.

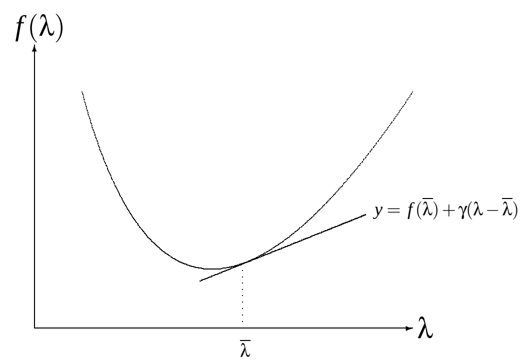
Definition 1 An m-vector γ is subgradient of $f(\lambda)$ at $\lambda = \bar{\lambda}$ if

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

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.



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

$$\gamma = d - Dx'$$

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

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

$$\max_{Ax \le b} (cx - \lambda(Dx - d)) \;\ge\; \gamma(\lambda - \overline{\lambda}) + \max_{Ax \le b}\left(cx - \overline{\lambda}(Dx - d)\right)$$

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

$$\begin{aligned}\max_{Ax \le b} (cx - \lambda(Dx - d)) &\ge (d - Dx')(\lambda - \overline{\lambda}) + (cx' - \overline{\lambda}(Dx' - d)) \\ &= cx' - \lambda(Dx' - d)\end{aligned}$$

□

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

Intuition

Lagrange Relaxation

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

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

Held and Karp

Initially

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

compute the new multipliers by recursion

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

where γ is subgradient.

The step size θ is defined by

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

where μ is an appropriate constant.
E.g. $\mu = 1$ and halved if upper bound not decreased in 20 iterations

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

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

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

Example: Manpower planning

$$\begin{array}{ll}
\text{minimize} & 5x_1+4x_2+6x_3+3x_4+7x_5+2x_6+5x_7+3x_8+2x_9+3x_{10} \\
\text{subject to} & x_1 + x_3 + x_5 + x_8 \geq 1 \\
& x_2 + x_3 + x_4 + x_5 + x_7 + x_8 + x_{10} \geq 1 \\
& x_4 + x_6 \geq 1 \\
& x_1 + x_2 + x_3 + x_6 + x_8 + x_{10} \geq 1 \\
& x_1 + x_2 + x_7 + x_8 + x_9 \geq 1 \\
& x_2 + x_3 + x_{10} \geq 1 \\
& x_5 \geq 1 \\
& x_1 + x_2 + x_4 + x_6 + x_8 + x_{10} \geq 1 \\
& x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8,x_9,x_{10} \in \{0,1\}
\end{array}$$

Optimal solution is

$$x_2 = x_5 = x_6 = 1$$

objective value 13

Notice

Minimization problem
Lagrangian dual is a maximization problem
Subgradient $\gamma = Ax' - b$ where $b = (1,\ldots,1)$

Example: Manpower planning

Lagrange relaxing all constraints using $\lambda \geq 0$

$$\begin{array}{ll}
\text{minimize} & \sum_{j=1}^{n} c_j x_j - \sum_{i=1}^{m} \lambda_i \left(\sum_{j=1}^{n} a_{ij} x_j - 1 \right) \\
\text{subject to} & x_j \in \{0,1\}
\end{array}$$

which can be reduced to

$$\begin{array}{ll}
\text{minimize} & \sum_{j=1}^{n} \left(c_j - \sum_{i=1}^{m} \lambda_i a_{ij} \right) x_j + \sum_{i=1}^{m} \lambda_i \\
\text{subject to} & x_j \in \{0,1\}
\end{array}$$

Latter problem is easily solved by inspection

- $c_j - \sum_{i=1}^{m} \lambda_i a_{ij} < 0$ then $x_j = 1$
- $c_j - \sum_{i=1}^{m} \lambda_i a_{ij} > 0$ then $x_j = 0$
- $c_j - \sum_{i=1}^{m} \lambda_i a_{ij} = 0$ then $x_j = 1$ or 0

Remaining constraints define the convex hull, hence best choice of λ corresponds to dual variables
Many manpower problems are so large that they cannot be solved by LP solvers

Example: Manpower planning

$$\begin{array}{ll}
\text{minimize} & 5x_1+4x_2+6x_3+3x_4+7x_5+2x_6+5x_7+3x_8+2x_9+3x_{10} \\
\text{subject to} & x_1 + x_3 + x_5 + x_8 \geq 1 \\
& x_2 + x_3 + x_4 + x_5 + x_7 + x_8 + x_{10} \geq 1 \\
& x_4 + x_6 \geq 1 \\
& x_1 + x_2 + x_3 + x_6 + x_8 + x_{10} \geq 1 \\
& x_1 + x_2 + x_7 + x_8 + x_9 \geq 1 \\
& x_2 + x_3 + x_{10} \geq 1 \\
& x_5 \geq 1 \\
& x_1 + x_2 + x_4 + x_6 + x_8 + x_{10} \geq 1 \\
& x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8,x_9,x_{10} \in \{0,1\}
\end{array}$$

Lagrange relaxed

$$\begin{array}{ll}
\text{minimize} & (5-\lambda_1-\lambda_4-\lambda_5-\lambda_8)x_1 \\
& +(4-\lambda_2-\lambda_4-\lambda_5-\lambda_6-\lambda_8)x_2 \\
& +(6-\lambda_1-\lambda_2-\lambda_4-\lambda_6)x_3 \\
& +(3-\lambda_2-\lambda_3-\lambda_8)x_4 \\
& +(7-\lambda_1-\lambda_2-\lambda_7)x_5 \\
& +(2-\lambda_3-\lambda_4-\lambda_8)x_6 \\
& +(5-\lambda_2-\lambda_5)x_7 \\
& +(3-\lambda_1-\lambda_2-\lambda_4-\lambda_5-\lambda_8)x_8 \\
& +(2-\lambda_5)x_9 \\
& +(3-\lambda_2-\lambda_4-\lambda_6-\lambda_8)x_{10} \\
& +\lambda_1+\lambda_2+\lambda_3+\lambda_4+\lambda_5+\lambda_6+\lambda_7+\lambda_8 \\
\text{subject to} & x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8,x_9,x_{10} \in \{0,1\}
\end{array}$$

Start with $\lambda^{(1)} = (1,1,1,1,1,1,1,1)$
$\underline{z} = 0$, $\bar{z} = 40$ (sum of all costs)

Example: Manpower planning

Iteration 1

$$\begin{array}{ll}
\text{minimize} & x_1 - x_2 + 2x_3 + 0x_4 + 4x_5 - x_6 + 3x_7 - 2x_8 + x_9 - x_{10} + 8 \\
\text{subject to} & x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8,x_9,x_{10} \in \{0,1\}
\end{array}$$

Optimal solution $x' = (0,1,0,1,0,1,0,1,0,1)$, infeasible
$\underline{z} = 3$, $\bar{z} = 40$

$$\gamma^{(1)} = Ax' - b = (0,3,1,3,1,1,-1,4),$$

$\theta = 0.19$, $\lambda^{(2)} = (1.00, 0.42, 0.81, 0.42, 0.81, 0.81, 1.19, 0.22)$

Iteration 2

$$\begin{array}{ll}
\text{minimize} & 2.56x_1 + 1.34x_2 + 3.36x_3 + 1.56x_4 + 4.39x_5 + \\
& 0.56x_6 + 3.78x_7 + 0.14x_8 + 1.19x_9 + 1.14x_{10} + 5.66 \\
\text{subject to} & x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8,x_9,x_{10} \in \{0,1\}
\end{array}$$

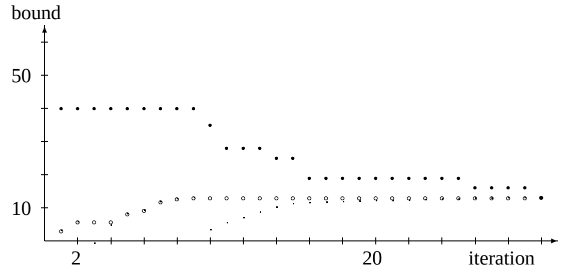
Optimal solution $x' = (0,0,0,0,0,0,0,0,0,0)$, infeasible
$\underline{z} = 5.66$, $\bar{z} = 40$

$$\gamma^{(1)} = Ax' - b = (-1,-1,-1,-1,-1,-1,-1,-1),$$

$\theta = 0.86$, $\lambda^{(3)} = (1.86, 1.27, 1.66, 1.27, 1.66, 1.66, 2.05, 1.08)$

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