Theory of Computing

Lecture 13

MAS 714

Hartmut Klauck

Linear Programming

- **Definition**
  - A Linear Program (LP) is an optimization problem over real variables $x_1,\ldots,x_n$
  - Maximizes/minimizes a linear function of x:
    max/min $C(x)=\sum_i c_i x_i$ (c_i are real coefficients)
  - Constraints: feasible solutions are those that satisfy a system of linear inqualities:
    - A is a real $m \times n$ matrix, b a vector
    - All x with $Ax \leq b$ are feasible
  - We are looking for a feasible x with maximum C(x)

Standard form

- Constraints using $\geq$, $\leq$ and = are possible
- Easy to reduce to the *standard form*:
  - max $c^T x$
  - Constraints:
  - $Ax \leq b$
  - $x \geq 0$
- c,b (column) vectors of lengths n,m
- x vector of n variables
- A is m by n matrix of coefficients

Duality

- Given an LP (A,b,c) in standard form
- We call this the *primal* LP
- The dual LP is defined as follow:
  - There are m variables y_i
  - Minimize $\sum_i b_i\, y_i$
  - Constraints:
    - $\sum_{i=1}^{m} A[i,j]\, y_i \geq c_j$
    - $y \geq 0$

Weak Duality

- **Claim:**
  - Let x be a feasible solution for the primal LP and y a feasible solution for the dual LP, then

$$\sum_{j=1}^{n} c_j x_j \leq \sum_{i=1}^{m} b_i y_i$$

Strong Duality

- Strong duality means that feasible solutions with the *same* value exist for the primal and the dual

- Compare Max Flow/Min Cut theorem

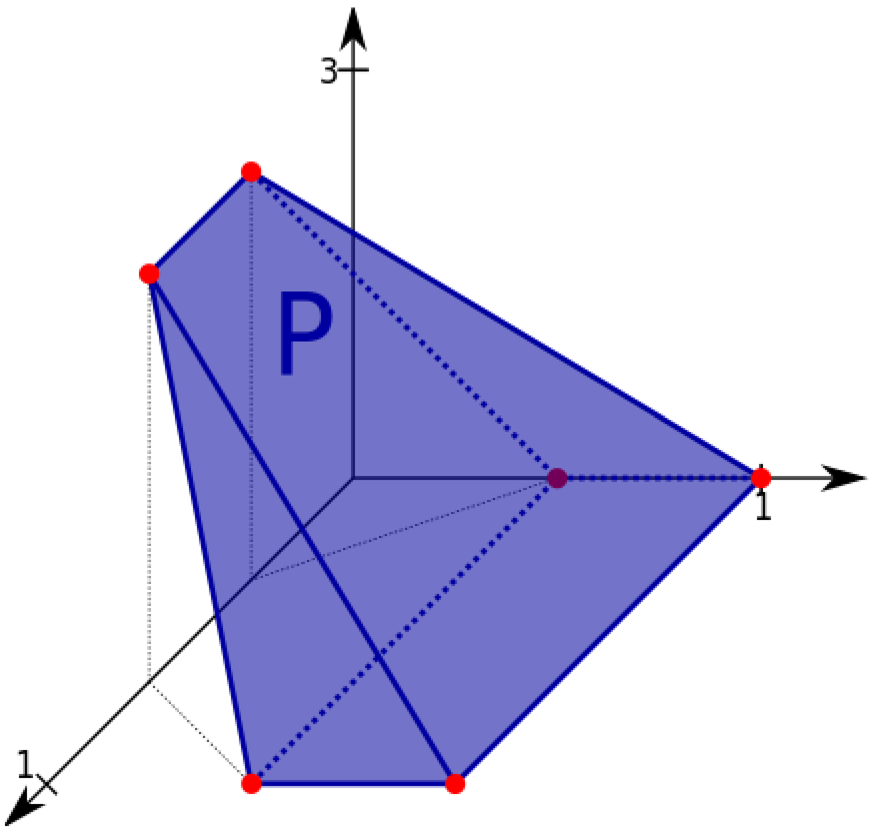
Linear Programming

- **Some facts:**
  - In practice LP‘s are usually solved with the Simplex Algorithm
  - Many math programs/libraries have LP solvers
  - Simplex is exponential in the worst case
  - There is a theoretical explanation why Simplex is fast in practice [smoothed analysis]
  - Worst case polynomial time algorithms:
    - Ellipsoid
    - Interior Point Methods

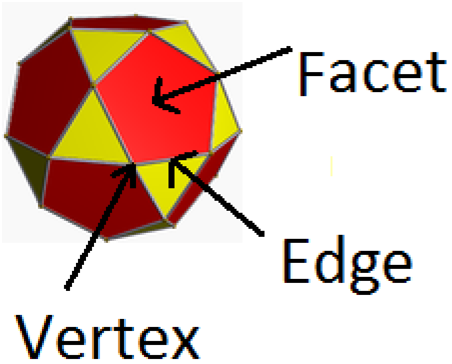
Geometry

- In n-dimensional space
- Hyperplane: n-1 dimensional affinee subspace, defined by a linear equation
- Halfspace: all points that satisfy a linear inequality
- Convex polytope: convex hull of a finite set of (affine independent) points $p_1,\ldots,p_m$
  - vertex of a polytope: one of the p_i but also an extreme point
  - edge: convex hull of 2 extreme points p_i, p_j
  - d-face: convex hull of d points
  - facet: (m-1)-face
- Simplex: n-dimensional polytope with n+1 vertices

A Convex Polytope



Vertices, Facets, Edges



Geometry

- Consider R^n
- Every inequality corresponds to a halfspace
- Feasible solutions lie in an intersection of m halfspaces
- This set is a convex polytope
  - Actually a convex polyhedron, since it is not necessarily a bounded set
  - We will just say polytope
- The objective function points in some direction

Types of LPs

- For an LP the objective function can be unbounded
  - Solution set/polytope must be unbounded
- There can be no solution
  - Solution set empty
- Or there is a nonempty set of optimal solutions
  - A face of the polytope
  - There is always an optimal vertex

Slack Form

- We can express constraints as equations as follows:
  - Take $\sum A[i,j]\, x_j \leq b_i$
  - Introduce a new variable x_{n+i} and set
  - $x_{n+i} = b_i - \sum A[i,j] x_j$ and $x_{n+i} \geq 0$
- x_{n+i} are called *slack variables*
- Doing this with all inequalities yields a system with n+m variables and m equations and the global nonnegativity constraint $x \geq 0$
- The variables on the left side of the equations (at any time) are called *basic variables*

Simplex Algorithm

- Use slack form
- Start at a vertex of the convex polytope (how?)

Simplex Algorithm

- Iteration:
  - follow an edge to a better vertex
- We reach a local maximum
- Due to convexity and linearity of the cost function this is also a global maximum

Simplex Algorithm

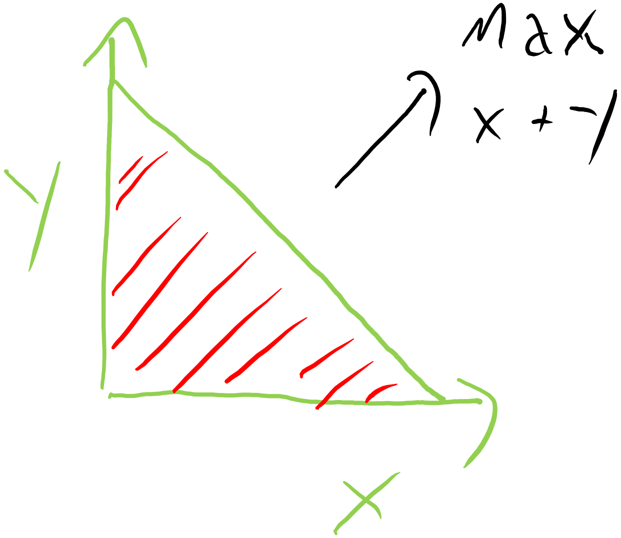
- Input: LP in slack form
- m basic variables and n nonbasic variables
  - basic variables do not appear in the objective function
- **Basic solution**:
  - Set all nonbasic variables to 0 and all basic variables via their equations
  - Assume (for now) this is a feasible solution
  - Then this is a vertex of the polytope
- Increase a nonbasic variable to improve the solution (until not possible anymore)
- Exchange the nonbasic variable with a basic variable
  - Rewrite the LP
- Continue until (local) optimum is reached

Simplex Algorithm

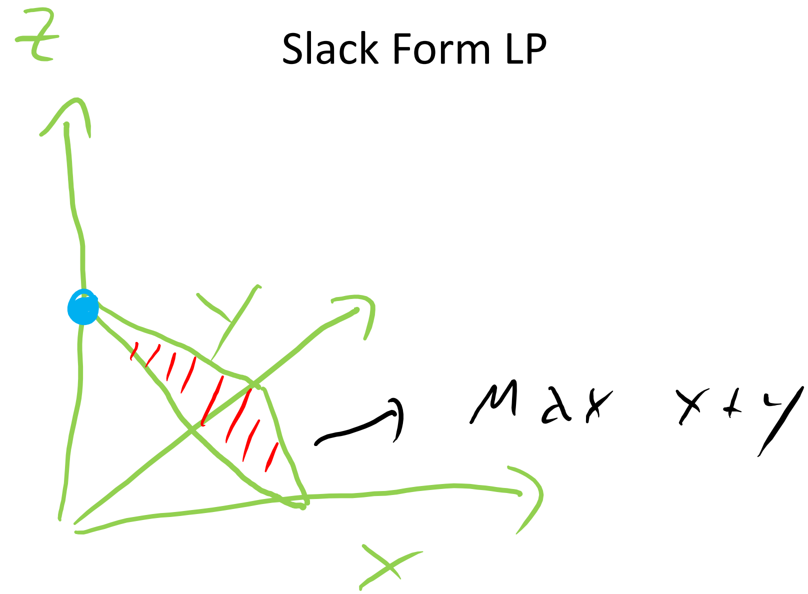
- Example:
- Max x+y such that x+y$\leq$ 1 and x,y$\geq$ 0
- Slack form:
  - Max w
    - w=x+y
    - z=1-x-y
    - x,y,z $\geq$ 0

Example

Standard LP



Slack Form LP



Basic solution

Example

- Initial basic solution: x=y=0, z=1
  - value is 0
- Choose x and increase x until constraint violated
  - Now x=1, y=0, z=0
  - Rewrite LP:
    - Max w
    - w=y+x = y+1-y-z = 1-z
    - x=1-z-y
    - x,y,z$\geq$ 0
  - New basic solution: y=z=0, x=1

At this point z,y are nonbasic, but we cannot increase z,y and increase the objective function. Simplex stops

Change of the slack form

- Choose a nonbasic variable x_e with $c(e)>0$. Increasing x_e increases the objective function
- Increasing x_e changes the value of basic variables
- Increase x_e until a basic variable is 0
  - If this is never the case the objective function is unbounded
- Change the LP to make x_e a basic variable
  - This is called pivoting
- Iterate

- Geometry: move along edges of the polytope

Operation pivot

- Input:
  - slack form LP:
    - N: Indices of nonbasic variables
    - B: Indices of basic variables
    - A,b: Matrix and vector of the LP
    - v: constant in the objective function
    - c: coefficients of the objective function
    - e: Index of some nonbasic variable
    - l: Index of a basic variable
- Output: all these
  - new slack form LP with x_e as basic variable and x_l as nonbasic variable

Operation pivot

1. b‘(e)=b(l)/A(l,e)
2. For all j $\in$ N-{e}: A‘(e,j)=A(l,j)/A(l,e)
3. A‘(e,l)=1/A(l,e)

4. For all i $\in$ B-{l}:
    1. b‘(i)=b(i)-A(i,e)b‘(e)
    2. For all j $\in$ N-{e}:
        1. A‘(i,j)=A(i,j)-A(i,e)A‘(e,j)
    3. A‘(i,l)=-A(i,e)A‘(e,l)

5. v‘=v+c(e)b‘(e)
6. For all j $\in$ N-{e}:
    1. c‘(j)=c(j)-c(e)A‘(e,j)
7. c‘(l)=-c(e)A‘(e,l)
8. Update B,N

Simplex Algorithm

- Inputs A,b,c

1. Bring LP into slack form, basic variables B, nonbasic variables N
2. While there is $j \in N$ with $c(j)>0$:
   1. Choose e with $c(e)>0$
   2. For all $i \in B$:
      1. If $A(i,e)>0$: $\Delta(i)=b(i)/A(i,e)$
      2. Else $\Delta(i)=\infty$
   3. Choose the minimal Δ_l
   4. $\Delta_l=\infty$: „objective function unbounded“
   5. Else: Pivot
3. For i=1 to n:
   1. If $i \in B$: Set $x_i=b(i)$
   2. else $x_i=0$
4. Output $x_1,\ldots,x_n$

Issues

- Need to find initial feasible basic solution
- How to choose the basic and nonbasic variables for pivot step
  - We choose the basic variable that first becomes 0 when we increase the nonbasic variable

Then:

- Correctness:
  – Computed solution is optimal
- Running time:
  – Termination?
- Find initial feasible basic solution
  – How?

Correctness (I)

- Claim:
  - Simplex computes a feasible solution
    - Or reports that the objective function is unbounded (correctly)
- Proof idea: all slack forms computed are equivalent

Termination

- The objective function never decreases
- It might not increase
- This is called a *degenerate* iteration.
- It can happen that Simplex get‘s into a loop

Termination

- Claim:
  - If Simplex uses more than $\binom{n+m}{m}$ Iterations, then it is in a loop
- Proof:
  - For a given LP and set of basic variables the slack form is unique
  - There are n+m variables und m basic variables
  - There are $\binom{n+m}{m}$ choices of basic variables
  - More iterations means loop

- So we can at least stop
- Loops are rare in practice

Preventing loops

- Loops happen, if the objective function does not increase
  - Solution: perturb the input slightly
- Bland‘s rule:
  - Always choose the variables with the smallest index (among candidates for x_e,x_l)
- Under Bland‘s Rule there are no loops

Running Time

- The running time is bounded by the number of vertices of the polytope
- Polytope has at most n+m facets and lies in an n-dimensional subspace
- Vertex: 0-dimensional, intersection of n facets
- Number of vertices is at most $\binom{n+m}{n}$

Correctness (II)

- We have to show that the solution obtained in the end is optimal
- This is proved using the dual

Strong Duality and Optimality

- Claim:
  - If $\sum c_j x_j = \sum b_i y_i$, and x is feasible for the primal and y is feasible for the dual, then both are optimal
- Proof:
  - By weak duality no solution x' of the primal can have larger value than the optimum of the dual.

Strong Duality

- Primal and Dual LP have feasible solutions x,y that have the same objective value

- **Theorem:**
  - Assume Simplex produces solution $x_1,\ldots,x_n$
  - B and N are the basic and nonbasic variables of the last slack form
  - c' coefficients of the objective function of the last slack form
  - define y by: $y_i = -c'_{n+i}$ if $n+i \in N$
    $= 0$ otherwise
  - Then x is optimal for the primal and y for the dual
  - x and y have the same objective value

Proof

- We show that x and y are feasible and have the same value
- x is feasible (found by Simplex)
- Last slack form:
  - objective function max $v' + \sum_N c'_j x_j$
- Upon termation: $c'_j \leq 0$ for all j in N
  Set $c'_j=0$ for all other j in B
- Objective function on x:
- I.e., x has cost v'
- Reason:
  - For all j in N: x(j)=0 in the last slack form
  - All slack forms have the same objective value

$$\underbrace{v' + \sum_N c'_j x_j}_{= v'} = \sum c_j x_j$$

Proof

$$\forall z_1, \ldots, z_n: \quad \sum c_j z_j = v' + \sum_{j=1}^{n+m} c_j' z_j$$

$$= v' + \sum_{j=1}^{n} c_j' z_j + \sum_{i=1}^{m} (-y_j) z_{n+i}$$

$$= v' + \sum_{j=1}^{n} c_j' z_j + \sum_{i=1}^{m} (-y_i)\left(b_i - \sum_{j=1}^{n} A[i,j] z_j\right)$$

$$\Rightarrow \sum_{j=1}^{n} c_j z_j = \left(v' - \sum_{i=1}^{m} b_i y_i\right) + \sum \left(c_j' + \sum_{i=1}^{m} A[i,j] y_i\right) z_j$$

Proof

- Above z(n+1),..., z(n+m) are x(n+1),..., x(n+m)
- Since this holds for all z(1),..., z(n), we have

$$v' - \sum_{i=1}^{m} b_i \cdot y_i = 0$$

$$c_j' + \sum_{i=1}^{m} A[i, j]\, y_i = c_j$$

- Accordingly (for all j=1…n) we can set y_i such that y is feasible, and both solutions have value v'

The initial basic solution

- Given an LP in Standard form A,b,c
- The corresponding slack form might not have a feasible basic solutionn

- We use a new LP:
  - Max $-x_0$
  - Constraints
    - $\sum_j A[i,j]\, x_j - x_0 \leq b_i$
    - $x \geq 0$
- Claim: This LP is always feasible and has maximum 0 iff A,b,c is feasible. Furthermore, the basic solution of its slack form becomes feasible after one call to pivot.

- We omit the proof

The initial basic solution

- Input : A,b,c
- Find slack form and test basic solution
- If this is not feasible , use new LP and its slack form
- Find optimal solution of the new LP with Simplex
- If the optimum is 0, and x_0 is not basic: remove x_0 and receive feasible basic solution and slack form for original LP (can now start Simplex)
- Otherwise report „not feasible“

Conclusion

- Algorithm Design Techniques:
  - Divide and Conquer
  - Greedy
  - Dynamic Programming
- Problems:
  - Sequences: Search, Sort, LCS
  - Graphs:
    - traversing graphs (BFS,DFS,Euler tours)
    - Spanning trees, Shortest paths, Max Flow
  - Linear Programming
- Data-structures:
  - Lists, stacks, queues, priority queues, union find