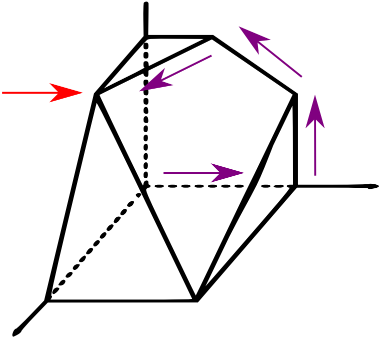
Algolab 2016
LP/QP: Some facts

Antonis Thomas

November 2, 2016

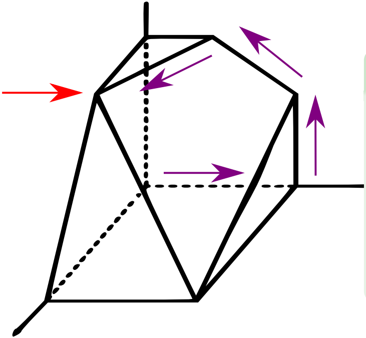
Solving Linear Programs

- The constraints define a convex polyhedron $P = \{x \mid Ax \leq b\}$
- Simplex-type algorithms walk along the edges of P to vertices with better values of the objective function until an optimum is reached.
- Decide on next edge: *pivot rule* or *pricing strategy*



Solving Linear Programs

- The constraints define a convex polyhedron $P = \{x \mid Ax \leq b\}$
- Simplex-type algorithms walk along the edges of P to vertices with better values of the objective function until an optimum is reached.
- Decide on next edge: *pivot rule* or *pricing strategy*

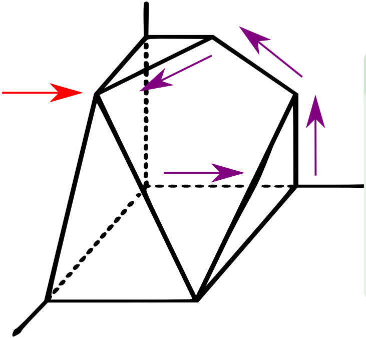


Complexity

- exponential!

Solving Linear Programs

- The constraints define a convex polyhedron $P = \{x \mid Ax \leq b\}$
- Simplex-type algorithms walk along the edges of P to vertices with better values of the objective function until an optimum is reached.
- Decide on next edge: *pivot rule* or *pricing strategy*

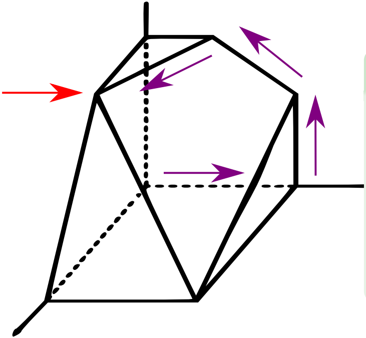


Complexity

- exponential!
- $O(m)$ if n is fixed (exponential in n) (m # inequalities, n # variables)

Solving Linear Programs

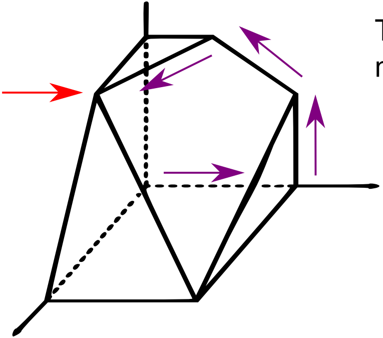
- The constraints define a convex polyhedron $P = \{x \mid Ax \leq b\}$
- Simplex-type algorithms walk along the edges of P to vertices with better values of the objective function until an optimum is reached.
- Decide on next edge: *pivot rule* or *pricing strategy*



Complexity

- exponential!
- $O(m)$ if n is fixed (exponential in n) (m # inequalities, n # variables)
- CGAL: for geometric applications, dimension typically $n \leq 50$ or ≤ 100

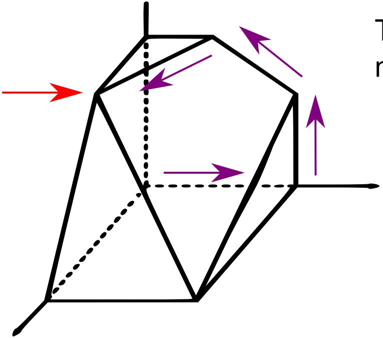
Solving Linear Programs with CGAL



The default pivot rule/pricing strategy might fail:

- it always picks the same improving edge every time it visits the same vertex
- suppose the algorithm reaches a certain vertex for the second time $\Rightarrow$ infinite cycle

Solving Linear Programs with CGAL



The default pivot rule/pricing strategy might fail:

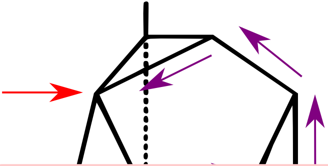
- it always picks the same improving edge every time it visits the same vertex
- suppose the algorithm reaches a certain vertex for the second time $\Rightarrow$ infinite cycle

Change the pivot rule

- Bland's pivot rule avoids cycling (but it is slower. . .)

```
CGAL::Quadratic_program_options options;
options.set_pricing_strategy(CGAL::QP_BLAND);
Solution s = CGAL::SOLVER(program, ET(), options);
```

Solving Linear Programs with CGAL



BUG - Don't Forget!

There is a bug in the assignment `operator=` for `Solution` objects…

The default pivot rule/pricing strategy might fail:

- it always picks the same improving edge every time it visits the same vertex
- suppose the algorithm reaches a certain vertex for the second time ⇒ infinite cycle

Change the pivot rule

- Bland's pivot rule avoids cycling (but it is slower…)

```
CGAL::Quadratic_program_options options;
options.set_pricing_strategy(CGAL::QP_BLAND);
Solution s = CGAL::SOLVER(program, ET(), options);
```