COMS21103: Linear Programming - cont.

Dima Damen
Dima.Damen@bristol.ac.uk

Bristol University, Department of Computer Science
Bristol BS8 1UB, UK

December 8, 2015

Initial Solution of Simplex Algorithm

Eg.

$$\begin{array}{ll} \text{maximise} & 2x_1 - x_2 \\ \text{subject to} & 2x_1 - x_2 \leq 2 \\ & x_1 - 5x_2 \leq -4 \\ & x_1, x_2 \geq 0 \end{array}$$

Initial Solution of Simplex Algorithm

Eg.

$$\begin{array}{ll} \text{maximise} & 2x_1 - x_2 \\ \text{subject to} & 2x_1 - x_2 \leq 2 \\ & x_1 - 5x_2 \leq -4 \\ & x_1, x_2 \geq 0 \end{array}$$

Slack form:

$$\boxed{\begin{array}{ccccccc} z & = & & & 2x_1 & - & x_2 \\ x_3 & = & 2 & - & 2x_1 & + & x_2 \\ x_4 & = & -4 & - & x_1 & + & 5x_2 \end{array}}$$

Initial Solution of Simplex Algorithm

Eg.

$$\begin{array}{ll} \text{maximise} & 2x_1 - x_2 \\ \text{subject to} & 2x_1 - x_2 \leq 2 \\ & x_1 - 5x_2 \leq -4 \\ & x_1, x_2 \geq 0 \end{array}$$

Slack form:

$$\boxed{\begin{array}{ccccccc} z & = & & & 2x_1 & - & x_2 \\ x_3 & = & 2 & - & 2x_1 & + & x_2 \\ x_4 & = & -4 & - & x_1 & + & 5x_2 \end{array}}$$

Initial solution (x_1, x_2, x_3, x_4) equals (0, 0, 2, -4) is infeasible

Initial Solution of Simplex Algorithm

- The Simplex Algorithm assumes the initial solution is feasible,

Initial Solution of Simplex Algorithm

- The Simplex Algorithm assumes the initial solution is feasible,
- We need to convert the linear program to slack form where the basic solution would be feasible

Initial Solution of Simplex Algorithm

change the linear program L

$$\begin{array}{lll}
\text{maximise} & \sum_{j=1}^{n} c_j x_j & \\
\text{subject to} & \sum_{j=1}^{n} a_{ij} x_j \leq b_i & \text{for } i = 1, 2, ..m \\
 & x_j \geq 0 & \text{for } j = 1, 2, ..n
\end{array}$$

to an auxiliary linear program L_{aux}

$$\begin{array}{lll}
\text{maximise} & -x_0 & \\
\text{subject to} & \sum_{j=1}^{n} a_{ij} x_j - x_0 \leq b_i & \text{for } i = 1, 2, ..m \\
 & x_j \geq 0 & \text{for } j = 0, 1, ..n
\end{array}$$

Initial Solution of Simplex Algorithm

change the linear program L

$$\begin{array}{lll} \text{maximise} & \sum_{j=1}^{n} c_j x_j & \\ \text{subject to} & \sum_{j=1}^{n} a_{ij} x_j \leq b_i & \text{for } i = 1, 2, ..m \\ & x_j \geq 0 & \text{for } j = 1, 2, ..n \end{array}$$

to an auxiliary linear program L_{aux}

$$\begin{array}{lll} \text{maximise} & -x_0 & \\ \text{subject to} & \sum_{j=1}^{n} a_{ij} x_j - x_0 \leq b_i & \text{for } i = 1, 2, ..m \\ & x_j \geq 0 & \text{for } j = 0, 1, ..n \end{array}$$

- L is feasible if and only if the optimal objective value of L_{aux} is 0

Initial Solution of Simplex Algorithm - Ex

Basic solution for L is infeasible

$$\begin{array}{ll} \text{maximise} & 2x_1 - x_2 \\ \text{subject to} & 2x_1 - x_2 \leq 2 \\ & x_1 - 5x_2 \leq -4 \\ & x_1, x_2 \geq 0 \end{array}$$

Initial Solution of Simplex Algorithm - Ex

Basic solution for L is infeasible

$$\begin{array}{ll} \text{maximise} & 2x_1 - x_2 \\ \text{subject to} & 2x_1 - x_2 \leq 2 \\ & x_1 - 5x_2 \leq -4 \\ & x_1, x_2 \geq 0 \end{array}$$

Change to L_{aux}:

$$\begin{array}{ll} \text{maximise} & -x_0 \\ \text{subject to} & 2x_1 - x_2 - x_0 \leq 2 \\ & x_1 - 5x_2 - x_0 \leq -4 \\ & x_0, x_1, x_2 \geq 0 \end{array}$$

Initial Solution of Simplex Algorithm - Ex

L_{aux}:

$$\begin{array}{ll} \text{maximise} & -x_0 \\ \text{subject to} & 2x_1 - x_2 - x_0 \leq 2 \\ & x_1 - 5x_2 - x_0 \leq -4 \\ & x_0, x_1, x_2 \geq 0 \end{array}$$

Initial Solution of Simplex Algorithm - Ex

L_{aux}:

$$
\begin{array}{ll}
\text{maximise} & -x_0 \\
\text{subject to} & 2x_1 - x_2 - x_0 \leq 2 \\
& x_1 - 5x_2 - x_0 \leq -4 \\
& x_0, x_1, x_2 \geq 0
\end{array}
$$

Slack form:

$$
\boxed{
\begin{array}{ccccccccc}
z & = & & & & & & - & x_0 \\
x_3 & = & 2 & - & 2x_1 & + & x_2 & + & x_0 \\
x_4 & = & -4 & - & x_1 & + & 5x_2 & + & x_0
\end{array}
}
$$

Initial Solution of Simplex Algorithm - Ex

L_{aux}:

$$
\begin{array}{ll}
\text{maximise} & -x_0 \\
\text{subject to} & 2x_1 - x_2 - x_0 \leq 2 \\
& x_1 - 5x_2 - x_0 \leq -4 \\
& x_0, x_1, x_2 \geq 0
\end{array}
$$

Slack form:

$$
\boxed{
\begin{array}{ccccccccc}
z & = & & & & & & - & x_0 \\
x_3 & = & 2 & - & 2x_1 & + & x_2 & + & x_0 \\
x_4 & = & -4 & - & x_1 & + & 5x_2 & + & x_0
\end{array}
}
$$

Initial solution $(x_0, x_1, x_2, x_3, x_4)$ equals (0, 0, 0, 2, -4) is **still** infeasible

Initial Solution of Simplex Algorithm - Ex

$$
\boxed{
\begin{array}{ccccccccc}
z & = & & & & & & - & x_0 \\
x_3 & = & 2 & - & 2x_1 & + & x_2 & + & x_0 \\
x_4 & = & -4 & - & x_1 & + & 5x_2 & + & x_0
\end{array}
}
$$

Switch x_0 with x_4 (constraint causing solution to be infeasbile)

Initial Solution of Simplex Algorithm - Ex

$$
\begin{array}{ccccccccc}
z & = & & & & & & - & x_0 \\
x_3 & = & 2 & - & 2x_1 & + & x_2 & + & x_0 \\
x_4 & = & -4 & - & x_1 & + & 5x_2 & + & x_0
\end{array}
$$

Switch x_0 with x_4 (constraint causing solution to be infeasbile)

$$
\begin{array}{ccccccccc}
x_0 & = & 4 & + & x_1 & - & 5x_2 & + & x_4
\end{array}
$$

Initial Solution of Simplex Algorithm - Ex

$$
\begin{array}{lllllllll}
z & = & & & & & & - & x_0 \\
x_3 & = & 2 & - & 2x_1 & + & x_2 & + & x_0 \\
x_4 & = & -4 & - & x_1 & + & 5x_2 & + & x_0
\end{array}
$$

Switch x_0 with x_4 (constraint causing solution to be infeasbile)

$$
\begin{array}{lllllllll}
z & = & -4 & - & x_1 & + & 5x_2 & - & x_4 \\
x_3 & = & 6 & - & x_1 & - & 4x_2 & + & x_4 \\
x_0 & = & 4 & + & x_1 & - & 5x_2 & + & x_4
\end{array}
$$

Basic solution $(x_0, x_1, x_2, x_3, x_4)$ is now feasible (4, 0, 0, 6, 0) for L_{aux} - but not yet for L

Initial Solution of Simplex Algorithm - Ex

Solve the auxiliary linear program L_{aux}

$$
\boxed{
\begin{array}{ccccccccc}
z & = & -4 & - & x_1 & + & 5x_2 & - & x_4 \\
x_3 & = & 6 & - & x_1 & - & 4x_2 & + & x_4 \\
x_0 & = & 4 & + & x_1 & - & 5x_2 & + & x_4
\end{array}
}
$$

Initial Solution of Simplex Algorithm - Ex

Solve the auxiliary linear program L_{aux}

$$
\boxed{\begin{array}{lllllllll}
z & = & -4 & - & x_1 & + & 5x_2 & - & x_4 \\
x_3 & = & 6 & - & x_1 & - & 4x_2 & + & x_4 \\
x_0 & = & 4 & + & x_1 & - & 5x_2 & + & x_4
\end{array}}
$$

Switch x_2 with x_0

Initial Solution of Simplex Algorithm - Ex

Solve the auxiliary linear program L_{aux}

$$\begin{array}{ccccccccc} z & = & -4 & - & x_1 & + & 5x_2 & - & x_4 \\ x_3 & = & 6 & - & x_1 & - & 4x_2 & + & x_4 \\ x_0 & = & 4 & + & x_1 & - & 5x_2 & + & x_4 \end{array}$$

Switch x_2 with x_0

$$\begin{array}{ccccccccc} x_2 & = & \frac{4}{5} & - & \frac{x_0}{5} & + & \frac{x_1}{5} & + & \frac{x_4}{5} \end{array}$$

Initial Solution of Simplex Algorithm - Ex

Solve the auxiliary linear program L_{aux}

$$\begin{array}{ccccccccc} z & = & -4 & - & x_1 & + & 5x_2 & - & x_4 \\ x_3 & = & 6 & - & x_1 & - & 4x_2 & + & x_4 \\ x_0 & = & 4 & + & x_1 & - & 5x_2 & + & x_4 \end{array}$$

Switch x_2 with x_0

$$\begin{array}{ccccccccc} z & = & & - & x_0 & & & & \\ x_3 & = & \frac{14}{5} & + & \frac{4x_0}{5} & - & \frac{9x_1}{5} & + & \frac{x_4}{5} \\ x_2 & = & \frac{4}{5} & - & \frac{x_0}{5} & + & \frac{x_1}{5} & + & \frac{x_4}{5} \end{array}$$

Initial Solution of Simplex Algorithm - Ex

Solve the auxiliary linear program L_{aux}

$$\begin{array}{lllllllll} z & = & -4 & - & x_1 & + & 5x_2 & - & x_4 \\ x_3 & = & 6 & - & x_1 & - & 4x_2 & + & x_4 \\ x_0 & = & 4 & + & x_1 & - & 5x_2 & + & x_4 \end{array}$$

Switch x_2 with x_0

$$\begin{array}{lllllllll} z & = & & - & x_0 & & & & \\ x_3 & = & \frac{14}{5} & + & \frac{4x_0}{5} & - & \frac{9x_1}{5} & + & \frac{x_4}{5} \\ x_2 & = & \frac{4}{5} & - & \frac{x_0}{5} & + & \frac{x_1}{5} & + & \frac{x_4}{5} \end{array}$$

Final solution to L_{aux} is $(0, 0, \frac{4}{5}, \frac{14}{5})$, and objective function is 0

Initial Solution of Simplex Algorithm - Ex

Final solution to L_{aux} is $(0, 0, \frac{4}{5}, \frac{14}{5})$, and objective function is 0

Initial Solution of Simplex Algorithm - Ex

Final solution to L_{aux} is $(0, 0, \frac{4}{5}, \frac{14}{5})$, and objective function is 0

$(x_1, x_2) = (0, \frac{4}{5})$ is also a feasible solution for L

$$\begin{array}{ll} \text{maximise} & 2x_1 - x_2 \\ \text{subject to} & 2x_1 - x_2 \leq 2 \\ & x_1 - 5x_2 \leq -4 \\ & x_1, x_2 \geq 0 \end{array}$$

Initial Solution of Simplex Algorithm - Ex

Final solution to L_{aux} is $(0, 0, \frac{4}{5}, \frac{14}{5})$, and objective function is 0

$(x_1, x_2) = (0, \frac{4}{5})$ is also a feasible solution for L

$$\begin{array}{ll} \text{maximise} & 2x_1 - x_2 \\ \text{subject to} & 2x_1 - x_2 \leq 2 \\ & x_1 - 5x_2 \leq -4 \\ & x_1, x_2 \geq 0 \end{array}$$

As we found a solution for L_{aux} with objective 0, we also know that the initial linear program L is feasible, and we have found a vertex on the convex hull of feasible explanations.

Initial Solution of Simplex Algorithm - Ex

Now rewrite the objective function to be $2x_1 - x_2$ for L_{aux}:

$$
\boxed{\begin{array}{llllllllll}
z & = & & - & x_0 & & & & & \\
x_3 & = & \frac{14}{5} & + & \frac{4x_0}{5} & - & \frac{9x_1}{5} & + & \frac{x_4}{5} \\
x_2 & = & \frac{4}{5} & - & \frac{x_0}{5} & + & \frac{x_1}{5} & + & \frac{x_4}{5}
\end{array}}
$$

Initial Solution of Simplex Algorithm - Ex

Now rewrite the objective function to be $2x_1 - x_2$ for L_{aux}:

$$
\begin{array}{|cccccccccc|}
\hline
z & = & & - & x_0 & & & & & \\
x_3 & = & \frac{14}{5} & + & \frac{4x_0}{5} & - & \frac{9x_1}{5} & + & \frac{x_4}{5} & \\
x_2 & = & \frac{4}{5} & - & \frac{x_0}{5} & + & \frac{x_1}{5} & + & \frac{x_4}{5} & \\
\hline
\end{array}
$$

objective function $2x_1 - x_2 = 2x_1 - (\frac{4}{5} - \frac{x_0}{5} + \frac{x_1}{5} + \frac{x_4}{5})$

Initial Solution of Simplex Algorithm - Ex

Now rewrite the objective function to be $2x_1 - x_2$ for L_{aux}:

$$
\boxed{
\begin{array}{llllllllll}
z & = & & - & x_0 & & & & \\
x_3 & = & \frac{14}{5} & + & \frac{4x_0}{5} & - & \frac{9x_1}{5} & + & \frac{x_4}{5} \\
x_2 & = & \frac{4}{5} & - & \frac{x_0}{5} & + & \frac{x_1}{5} & + & \frac{x_4}{5}
\end{array}
}
$$

objective function $2x_1 - x_2 = 2x_1 - (\frac{4}{5} - \frac{x_0}{5} + \frac{x_1}{5} + \frac{x_4}{5})$

Set $x_0 = 0$ and simplify to be $-\frac{4}{5} + \frac{9x_1}{5} - \frac{x_4}{5}$

Initial Solution of Simplex Algorithm - Ex

Now rewrite the objective function to be $2x_1 - x_2$ for L_{aux}:

$$\begin{array}{cccccccccc} z & = & & - & x_0 & & & & \\ x_3 & = & \frac{14}{5} & + & \frac{4x_0}{5} & - & \frac{9x_1}{5} & + & \frac{x_4}{5} \\ x_2 & = & \frac{4}{5} & - & \frac{x_0}{5} & + & \frac{x_1}{5} & + & \frac{x_4}{5} \end{array}$$

objective function $2x_1 - x_2 = 2x_1 - (\frac{4}{5} - \frac{x_0}{5} + \frac{x_1}{5} + \frac{x_4}{5})$

Set $x_0 = 0$ and simplify to be $-\frac{4}{5} + \frac{9x_1}{5} - \frac{x_4}{5}$

The Slack form will accordingly be

$$\begin{array}{cccccccc} z & = & -\frac{4}{5} & + & \frac{9x_1}{5} & - & \frac{x_4}{5} \\ x_3 & = & \frac{14}{5} & - & \frac{9x_1}{5} & + & \frac{x_4}{5} \\ x_2 & = & \frac{4}{5} & + & \frac{x_1}{5} & + & \frac{x_4}{5} \end{array}$$

Initial Solution of Simplex Algorithm - Ex

Now rewrite the objective function to be $2x_1 - x_2$ for L_{aux}:

$$\boxed{\begin{array}{llllllll} z & = & & - & x_0 & & & \\ x_3 & = & \frac{14}{5} & + & \frac{4x_0}{5} & - & \frac{9x_1}{5} & + \frac{x_4}{5} \\ x_2 & = & \frac{4}{5} & - & \frac{x_0}{5} & + & \frac{x_1}{5} & + \frac{x_4}{5} \end{array}}$$

objective function $2x_1 - x_2 = 2x_1 - (\frac{4}{5} - \frac{x_0}{5} + \frac{x_1}{5} + \frac{x_4}{5})$

Set $x_0 = 0$ and simplify to be $-\frac{4}{5} + \frac{9x_1}{5} - \frac{x_4}{5}$

The Slack form will accordingly be

$$\boxed{\begin{array}{llllll} z & = & -\frac{4}{5} & + \frac{9x_1}{5} & - & \frac{x_4}{5} \\ x_3 & = & \frac{14}{5} & - \frac{9x_1}{5} & + & \frac{x_4}{5} \\ x_2 & = & \frac{4}{5} & + \frac{x_1}{5} & + & \frac{x_4}{5} \end{array}}$$

Initial feasible solution is $(x_1, x_2, x_3, x_4) = (0, \frac{4}{5}, \frac{14}{5}, 0)$

The Initialise-Simplex Algorithm

(N',B',A',b',c',v') = INITIALISE-SIMPLEX (A,b,c);

- returns modified slack form

The Initialise-Simplex Algorithm

(N',B',A',b',c',v') = INITIALISE-SIMPLEX (A,b,c);

Let k be the index of minimum b_i;

if $b_k \geq 0$ **then**
    return ({1,2,..,n},{n+1,..,n+m},A,b,c,0)

end

- returns modified slack form
- is initial solution feasible?

The Initialise-Simplex Algorithm

(N',B',A',b',c',v') = INITIALISE-SIMPLEX (A,b,c);

Let k be the index of minimum b_i;

if $b_k \geq 0$ **then**
    return ({1,2,..,n},{n+1,..,n+m},A,b,c,0)

end

form L_{aux}

- returns modified slack form
- is initial solution feasible?
- add $-x_0$ to each constraint and set objective to $-x_0$

The Initialise-Simplex Algorithm

(N',B',A',b',c',v') = INITIALISE-SIMPLEX (A,b,c);

Let k be the index of minimum b_i;

if $b_k \geq 0$ **then**
    return ({1,2,..,n},{n+1,..,n+m},A,b,c,0)

end

form L_{aux}

$l = n + k$

- returns modified slack form
- is initial solution feasible?
- add $-x_0$ to each constraint and set objective to $-x_0$
- decide on basic variable with minimum b_i

The Initialise-Simplex Algorithm

(N',B',A',b',c',v') = INITIALISE-SIMPLEX (A,b,c);

Let k be the index of minimum b_i;

if $b_k \geq 0$ **then**
  return ({1,2,..,n},{n+1,..,n+m},A,b,c,0)

end

form L_{aux}

$l = n + k$

(N,B,A,b,c,v) = PIVOT(N,B,A,c,v,l,0)

- returns modified slack form
- is initial solution feasible?
- add $-x_0$ to each constraint and set objective to $-x_0$
- decide on basic variable with minimum b_i
- switch the roles of x_0 and x_l. Basic solution is now feasible

The Initialise-Simplex Algorithm

(N',B',A',b',c',v') = INITIALISE-SIMPLEX (A,b,c);

Let k be the index of minimum b_i;

if $b_k \geq 0$ **then**
    return ({1,2,..,n},{n+1,..,n+m},A,b,c,0)

end

form L_{aux}

$l = n + k$

(N,B,A,b,c,v) = PIVOT(N,B,A,c,v,l,0)

Solve SIMPLEX for L_{aux}

- returns modified slack form
- is initial solution feasible?
- add $-x_0$ to each constraint and set objective to $-x_0$
- decide on basic variable with minimum b_i
- switch the roles of x_0 and x_l. Basic solution is now feasible
- iterate lines 2-12 of SIMPLEX

The Initialise-Simplex Algorithm

(N',B',A',b',c',v') = INITIALISE-SIMPLEX (A,b,c);
Let k be the index of minimum b_i;
if $b_k \geq 0$ **then**
    return ({1,2,..,n},{n+1,..,n+m},A,b,c,0)
end
form L_{aux}
$l = n + k$
(N,B,A,b,c,v) = PIVOT(N,B,A,c,v,l,0)
Solve SIMPLEX for L_{aux}
if *optimal solution of* L_{aux} *sets* x_0 *to 0* **then**

else
    return "infeasible"
end

- returns modified slack form
- is initial solution feasible?
- add $-x_0$ to each constraint and set objective to $-x_0$
- decide on basic variable with minimum b_i
- switch the roles of x_0 and x_l. Basic solution is now feasible
- iterate lines 2-12 of SIMPLEX
- L is feasible

Dima Damen
Dima.Damen@bristol.ac.uk

The Initialise-Simplex Algorithm

(N',B',A',b',c',v') = INITIALISE-SIMPLEX (A,b,c);
Let k be the index of minimum b_i;
if $b_k \geq 0$ **then**
    return ({1,2,..,n},{n+1,..,n+m},A,b,c,0)
end
form L_{aux}
$l = n + k$
(N,B,A,b,c,v) = PIVOT(N,B,A,c,v,l,0)
Solve SIMPLEX for L_{aux}
if *optimal solution of* L_{aux} *sets* x_0 *to 0* **then**
    if x_0 *is basic* **then**
        perform one PIVOT to make it nonbasic
    end
    remove x_0 from constraints and restore L
else
    return "infeasible"
end

- returns modified slack form
- is initial solution feasible?
- add $-x_0$ to each constraint and set objective to $-x_0$
- decide on basic variable with minimum b_i
- switch the roles of x_0 and x_l. Basic solution is now feasible
- iterate lines 2-12 of SIMPLEX
- L is feasible
- Restore original objective function

The Initialise-Simplex Algorithm

Lemma

Given a linear program (A,b,c), suppose that the call to INITIALISE-SIMPLEX returns a slack form for which the basic solution is feasible, then if SIMPLEX returns a solution, it is a feasible solution to the linear program. If it returns "unbounded", the linear program is unbounded.

Fundamental Theorem of Linear Programming

Theorem

Any linear program L, given in standard form, either

1. *has an optimal solution with a finite objective value,*
2. *is infeasible, or*
3. *is unbounded.*

If L is infeasible, SIMPELX returns "infeasble". If L is unbounded, SIMPLEX returns "unbounded". Otherwise SIMPLEX returns an optimal solution with a finite objective value.

The Simplex Algorithm - Time Analysis

The Simplex Algorithm - Time Analysis

- Klee and Minty [1972] and Avis and Chvatal [1978] found examples where the SIMPELX algorithm needs 2^n iterations on L with n variables and $2n$ constraints.

The Simplex Algorithm - Time Analysis

- Klee and Minty [1972] and Avis and Chvatal [1978] found examples where the SIMPELX algorithm needs 2^n iterations on L with n variables and $2n$ constraints.
- Simplex algorithm is not a polynomial-time algorithm.

The Simplex Algorithm - Time Analysis

- Klee and Minty [1972] and Avis and Chvatal [1978] found examples where the SIMPELX algorithm needs 2^n iterations on L with n variables and $2n$ constraints.
- Simplex algorithm is not a polynomial-time algorithm.
- Not known whether there is a pivot rule that leads to polynomial time.

The Simplex Algorithm - Time Analysis

- Klee and Minty [1972] and Avis and Chvatal [1978] found examples where the SIMPELX algorithm needs 2^n iterations on L with n variables and $2n$ constraints.
- Simplex algorithm is not a polynomial-time algorithm.
- Not known whether there is a pivot rule that leads to polynomial time.
- Borgwardt [1982] showed that average running time can be bounded by a polynomial.

The Simplex Algorithm - Time Analysis

- Klee and Minty [1972] and Avis and Chvatal [1978] found examples where the SIMPELX algorithm needs 2^n iterations on L with n variables and $2n$ constraints.
- Simplex algorithm is not a polynomial-time algorithm.
- Not known whether there is a pivot rule that leads to polynomial time.
- Borgwardt [1982] showed that average running time can be bounded by a polynomial.
- The Simplex algorithm is very efficient in practice.

The Simplex Algorithm - Time Analysis

- Klee and Minty [1972] and Avis and Chvatal [1978] found examples where the SIMPELX algorithm needs 2^n iterations on L with n variables and $2n$ constraints.
- Simplex algorithm is not a polynomial-time algorithm.
- Not known whether there is a pivot rule that leads to polynomial time.
- Borgwardt [1982] showed that average running time can be bounded by a polynomial.
- The Simplex algorithm is very efficient in practice.
- The Ellipsoid algorithm (Iudin and Nemirovskii [1976] and Shor [1977]) is proven to be a polynomial-time algorithm.
- The Ellipsoid algorithm is though too inefficient in practice.

Linear Programs - Duality

Definition

Given a linear program L (also known as the **primal** L),

$$\begin{array}{lll} \text{maximise} & \sum_{j=1}^{n} c_j x_j & \\ \text{subject to} & \sum_{j=1}^{n} a_{ij} x_j \leq b_i & \text{for } i = 1, 2, ..m \\ & x_j \geq 0 & \text{for } j = 1, 2, ..n \end{array}$$

we define the **dual** L to be the linear program,

$$\begin{array}{lll} \text{minimise} & \sum_{j=1}^{m} b_j y_j & \\ \text{subject to} & \sum_{j=1}^{n} a_{ji} y_j = c_i & \text{for } i = 1, 2, ..n \\ & y_j \geq 0 & \text{for } j = 1, 2, ..m \end{array}$$

Linear Programs - Duality Ex

$$\begin{array}{ll} \text{maximise} & 18x_1 + 12.5x_2 \\ \text{subject to} & x_1 + x_2 \leq 20 \\ & x_1 \leq 12 \\ & x_2 \leq 16 \\ & x_1, x_2 \geq 0 \end{array}$$

is equivalent to:

$$\begin{array}{ll} \text{minimise} & 20y_1 + 12y_2 + 16y_3 \\ \text{subject to} & y_1 + y_2 = 18 \\ & y_1 + y_3 = 12.5 \\ & y_1, y_2, y_3 \geq 0 \end{array}$$

Linear Programs - Duality

Lemma

The dual of the dual of a linear program L is (equivalent to) the primal L.

Linear Programs - Duality

Lemma

If a primal linear program L is unbounded then its dual L is infeasible.
If a primal linear program L has an optimum solution, then its dual also has an optimum solution.

The Simplex Algorithm

Back to the Convex Hull...

The Simplex Algorithm

Back to the Convex Hull...

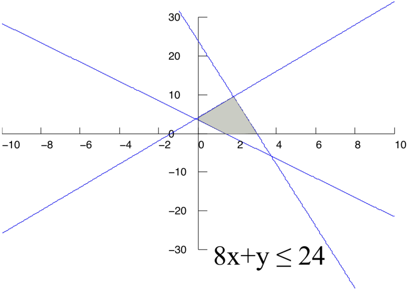
- The solution at each iteration of the simplex algorithm represents a vertex in the space of feasible solutions.

The Simplex Algorithm

Back to the Convex Hull...

- For the linear program

$$
\begin{array}{ll}
\textbf{maximise} & -x + y \\
\textit{subject to} & -5x - 2y \leq -7 \\
 & -3x + y \leq 4 \\
 & 8x + y \leq 24
\end{array}
$$



The Simplex Algorithm

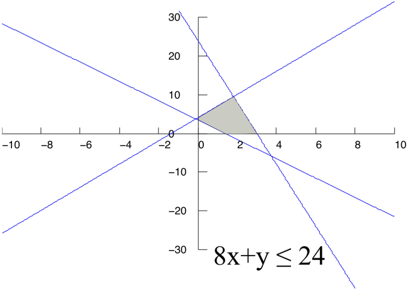
Back to the Convex Hull...

- For the linear program

maximise	$-x + y$
subject to	$-5x - 2y \leq -7$
	$-3x + y \leq 4$
	$8x + y \leq 24$

And its slack form

z	=		-	x	+	y
x_2	=	-7	+	$5x$	+	$2y$
x_3	=	4	+	$3x$	-	y
x_4	=	24	-	$8x$	-	y



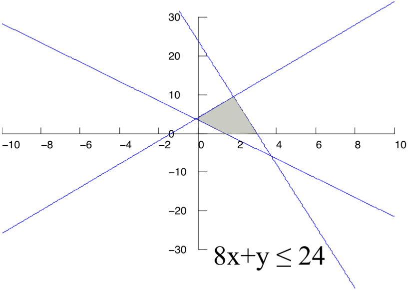
The Simplex Algorithm

Back to the Convex Hull...

- For the linear program — And its slack form

$$\begin{array}{ll} \textbf{maximise} & -x+y \\ \textit{subject to} & -5x-2y \leq -7 \\ & -3x+y \leq 4 \\ & 8x+y \leq 24 \end{array}$$

$$\begin{array}{cccccccc} z & = & & - & x & + & y \\ x_2 & = & -7 & + & 5x & + & 2y \\ x_3 & = & 4 & + & 3x & - & y \\ x_4 & = & 24 & - & 8x & - & y \end{array}$$



Initial solution $(x, y, x_2, x_3, x_4) = (0, 0, -7, 4, 24)$ is not feasible.

The Simplex Algorithm

Using Initialise-Simplex, the slack form can be re-written to be:

$$
\boxed{
\begin{array}{lllllll}
z & = & \frac{7}{2} & - & \frac{7x}{2} & + & \frac{x_2}{2} \\
y & = & \frac{7}{2} & - & \frac{5x}{2} & + & \frac{x_2}{2} \\
x_3 & = & \frac{1}{2} & + & \frac{11x}{2} & - & \frac{x_2}{2} \\
x_4 & = & \frac{41}{2} & - & \frac{11x}{2} & - & \frac{x_2}{2}
\end{array}
}
$$

The Simplex Algorithm

Using Initialise-Simplex, the slack form can be re-written to be:

$$\begin{array}{cccccccc} z & = & \frac{7}{2} & - & \frac{7x}{2} & + & \frac{x_2}{2} \\ y & = & \frac{7}{2} & - & \frac{5x}{2} & + & \frac{x_2}{2} \\ x_3 & = & \frac{1}{2} & + & \frac{11x}{2} & - & \frac{x_2}{2} \\ x_4 & = & \frac{41}{2} & - & \frac{11x}{2} & - & \frac{x_2}{2} \end{array}$$

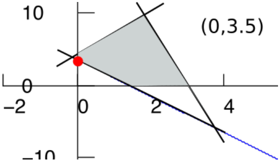
- Initial solution is $(0, \frac{7}{2}, 0, \frac{1}{2}, \frac{41}{2})$ is feasible

The Simplex Algorithm

Using Initialise-Simplex, the slack form can be re-written to be:

$$
\boxed{
\begin{array}{cccccccc}
z & = & \frac{7}{2} & - & \frac{7x}{2} & + & \frac{x_2}{2} \\
y & = & \frac{7}{2} & - & \frac{5x}{2} & + & \frac{x_2}{2} \\
x_3 & = & \frac{1}{2} & + & \frac{11x}{2} & - & \frac{x_2}{2} \\
x_4 & = & \frac{41}{2} & - & \frac{11x}{2} & - & \frac{x_2}{2}
\end{array}
}
$$

- Initial solution is $(0, \frac{7}{2}, 0, \frac{1}{2}, \frac{41}{2})$ is feasible



- A vertex

The Simplex Algorithm

After one iteration of Simplex algorithm

$$
\boxed{
\begin{array}{llllll}
z & = & 4 & + & 2x & - & x_3 \\
y & = & 4 & + & 3x & - & x_3 \\
x_2 & = & 1 & + & 11x & - & 2x_3 \\
x_4 & = & 20 & - & 11x & + & x_3
\end{array}
}
$$

The Simplex Algorithm

After one iteration of Simplex algorithm

$$\boxed{\begin{array}{lllllll} z & = & 4 & + & 2x & - & x_3 \\ y & = & 4 & + & 3x & - & x_3 \\ x_2 & = & 1 & + & 11x & - & 2x_3 \\ x_4 & = & 20 & - & 11x & + & x_3 \end{array}}$$

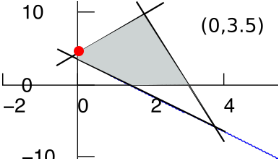
- Current solution is $(0, 4, 1, 0, 20)$ is feasible

The Simplex Algorithm

After one iteration of Simplex algorithm

$$
\begin{array}{lllllll}
z & = & 4 & + & 2x & - & x_3 \\
y & = & 4 & + & 3x & - & x_3 \\
x_2 & = & 1 & + & 11x & - & 2x_3 \\
x_4 & = & 20 & - & 11x & + & x_3
\end{array}
$$

- Current solution is $(0, 4, 1, 0, 20)$ is feasible



The Simplex Algorithm

After the second iteration of Simplex algorithm

$$
\begin{array}{llllll}
z & = & 7.636 & - & 0.818x_3 & - & 0.183x_4 \\
y & = & 9.455 & - & 0.727x_3 & - & 0.273x_4 \\
x_2 & = & 21 & - & x_3 & - & x_4 \\
x & = & 1.818 & + & 0.091x_3 & - & 0.091x_4
\end{array}
$$

The Simplex Algorithm

After the second iteration of Simplex algorithm

$$\begin{array}{lllllll}
z & = & 7.636 & - & 0.818x_3 & - & 0.183x_4 \\
y & = & 9.455 & - & 0.727x_3 & - & 0.273x_4 \\
x_2 & = & 21 & - & x_3 & - & x_4 \\
x & = & 1.818 & + & 0.091x_3 & - & 0.091x_4
\end{array}$$

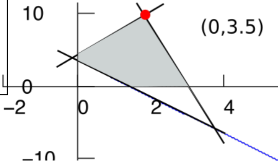
- Final solution is $(1.818, 9.455, 21, 0, 0)$

The Simplex Algorithm

After the second iteration of Simplex algorithm

$$
\begin{array}{lllllll}
z & = & 7.636 & - & 0.818x_3 & - & 0.183x_4 \\
y & = & 9.455 & - & 0.727x_3 & - & 0.273x_4 \\
x_2 & = & 21 & - & x_3 & - & x_4 \\
x & = & 1.818 & + & 0.091x_3 & - & 0.091x_4
\end{array}
$$

- Final solution is $(1.818, 9.455, 21, 0, 0)$



Further Reading

- Introduction to Algorithms
  T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein.
  MIT Press/McGraw-Hill, ISBN: 0-262-03293-7.
  - Chapter 27 – Linear Programming
- Combinatorial Optimization, Theory and Algorithms
  B. Korte and J. Vygen.
  Springer, 4th edition.
  - Chapter 3 – Linear Programming