

Basic linear programming in polymake

November 6, 2018

Michael Joswig (TU Berlin): Algorithmic Discrete Mathematics III, Winter 2018/19

1 Solve a first linear program

```
In [1]: $P = new Polytope(INEQUALITIES=>[[0,1,0,0],  
    [1,-1,0,0],[0,0,1,0],[1,0,-1,0],[0,0,0,1],[1,0,0,-1]]);
```

```
In [2]: $P->add("LP",new LinearProgram(LINEAR_OBJECTIVE=>[0,1,1,1]));
```

```
In [3]: print $P->LP->MINIMAL_VALUE;
```

```
Out [3]: 0
```

There is more information available.

```
In [4]: map { print $P->LP->$_, "\n"}  
    ("MINIMAL_VALUE", "MINIMAL_VERTEX", "MINIMAL_FACE");
```

```
Out [4]: 0  
    1 0 0 0  
    {5}
```

```
In [5]: print rows_labeled($P->VERTICES);
```

```
Out [5]: 0:1 1 0 0  
    1:1 1 1 0  
    2:1 1 1 1  
    3:1 1 0 1  
    4:1 0 0 1  
    5:1 0 0 0  
    6:1 0 1 0  
    7:1 0 1 1
```

```
In [6]: print $P->VERTICES->[5];
```

```
Out [6]: 1 0 0 0
```

```
In [7]: map { print $_, ":", $P->VERTICES->[$_], "\n" }
        (@{$P->LP->MINIMAL_FACE});
```

```
Out [7]: 5:1 0 0 0
```

2 Look at a slightly more interesting example

This time we maximize.

```
In [8]: $Q = new Polytope(POINTS=>
        $P->VERTICES->minor(~[2],All));
```

```
In [9]: $Q->add("LP",
        new LinearProgram(LINEAR_OBJECTIVE=>[0,1,1,1]));
```

```
In [10]: map { print $_, ":", $Q->VERTICES->[$_], "\n" }
         (@{$Q->LP->MAXIMAL_FACE});
```

```
Out [10]: 1:1 1 1 0
         2:1 1 0 1
         6:1 0 1 1
```

```
In [11]: $Q->VISUAL->MIN_MAX_FACE;
```

3 LP feasibility

```
In [12]: print $P->FEASIBLE;
```

```
Out [12]: true
```

```
In [13]: print $P->VALID_POINT;
```

```
Out [13]: 1 0 0 0
```

```
In [14]: print $P->REL_INT_POINT;
```

```
Out [14]: 1 1/2 1/2 1/2
```

```
In [ ]:
```