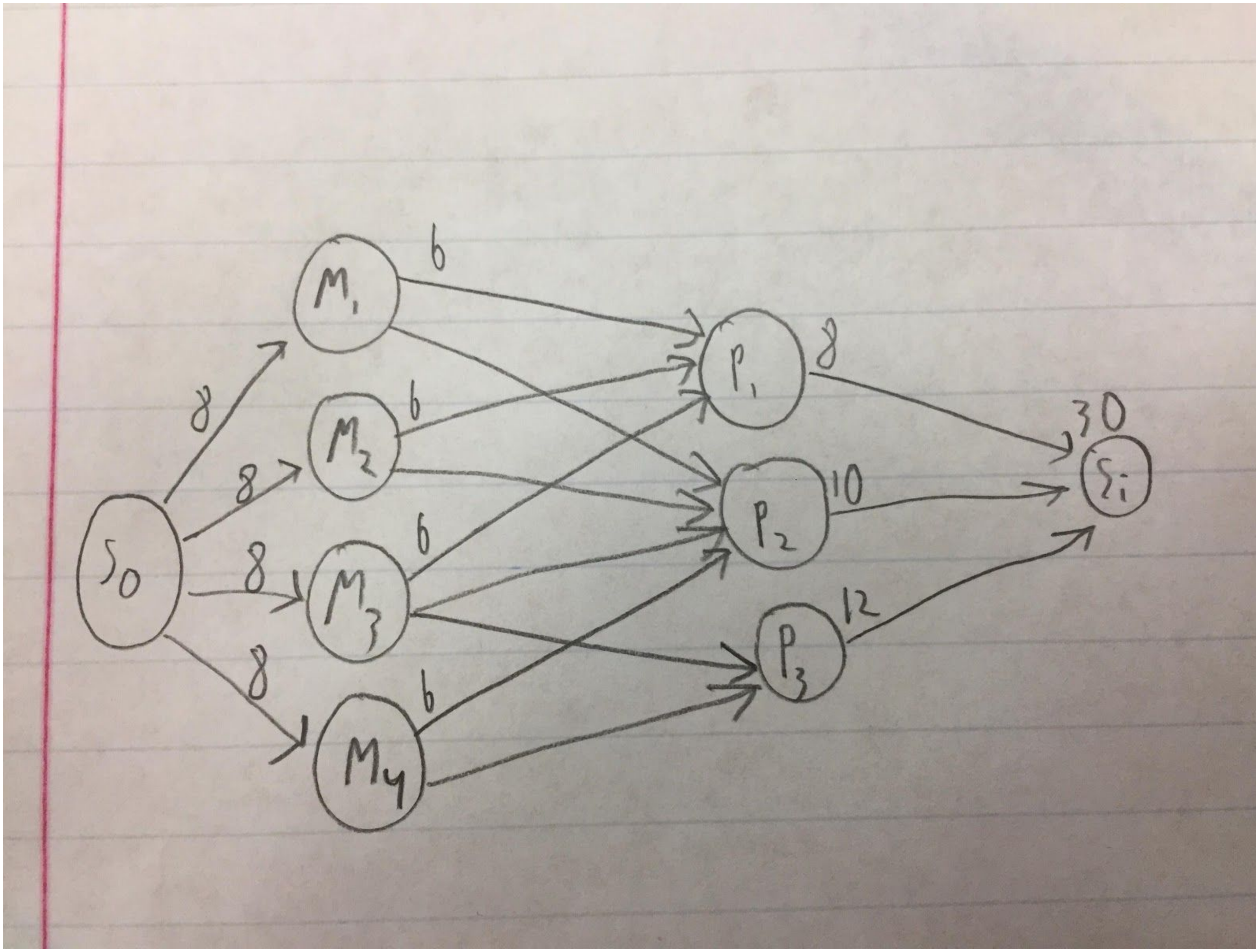
In [62]:

```
#Problem 1a: Cube
A = [1 0 0; -1 0 0; 0 1 0; 0 -1 0; 0 0 1; 0 0 -1]
b = [1;1;1;1;1;1]

#Problem 1b: Octahedron
A = [1 1 1; 1 -1 1;  -1 1 1; -1 -1 1;
   1 1 -1; 1 -1 -1; -1 1 -1; -1 -1 -1;]
b = [1,1,1,1,1,1,1,1]
```

In [63]:

#Problem3



In this diagram, so is the source node, M1-M4 are the individual months, P1-P3 are the three projects, and Si is a sink for output. Since the maximum flow of the network is 30, all projects can be completed on time.

```
#Problem 4a
using NamedArrays

# import Stigler's data set
raw = readcsv("stigler.csv")
(m,n) = size(raw)

n_nutrients = 2:n      # columns containing nutrients
n_foods = 3:m          # rows containing food names

nutrients = raw[1,n_nutrients][:]   # the list of nutrients (convert to 1-D array)
foods = raw[n_foods,1][:]           # the list of foods (convert to 1-D array)

# lower[i] is the minimum daily requirement of nutrient i.
lower = Dict( zip(nutrients,raw[2,n_nutrients]) )

# data[f,i] is the amount of nutrient i contained in food f.
data = NamedArray( raw[n_foods,n_nutrients], (foods,nutrients), ("foods","nutrients") );

#println(data)

using JuMP;

using Clp;

m = Model(solver = ClpSolver())
@variable(m, x[foods] >= 0)
@constraint(m, a[j in nutrients],
    sum(x[i]*data[i,j] for i in foods) >= lower[j])
@objective(m, Min, sum(x[i] for i in foods))


#print(m);

status = solve(m);

println("The minimized cost diet costs: ", getobjectivevalue(m)*365);
println("This optimal diet is 27 cents cheaper than Stigler's")
println("The diet is made up of:")
for i in foods
    if getvalue(x[i]) > 0
        println(getvalue(x[i]), "units of ", i)
    end
end

#Problem 4b

NonVeg = ["Lard", "Sirloin Steak", "Round Steak", "Rib Roast", "Chuck Roast", "Liver (Beef)",
"Leg of Lamb", "Lamb Chops (Rib)", "Pork Chops", "Pork Loin Roast", "Bacon", "Ham, smoked",
"Salt Pork", "Roasting Chicken", "Veal Cutlets", "Salmon, Pink (can)"]

mVegie = Model(solver = ClpSolver())
```

```julia
@variable(mVegie, costVeg[foods] >= 0)
@constraint(mVegie, nutrientsVeg[j in nutrients],
    sum(costVeg[i]*data[i,j] for i in foods) >= lower[j])
@constraint(mVegie, y[i in NonVeg], costVeg[i] == 0)
@objective(mVegie, Min, sum(costVeg[i] for i in foods))

solve(mVegie)

println()
println("Min cost of Vegie diet is : \$", getobjectivevalue(mVegie)*365)
println("The vegitarian diet consists of:")
for i in foods
    if getvalue(costVeg[i]) > 0
        println(getvalue(costVeg[i]), " units of ", i)
    end
end
```

```
The minimized cost diet costs: 39.66173154546625
This optimal diet is 27 cents cheaper than Stigler's
The diet is made up of:
0.02951906167648827units of Wheat Flour (Enriched)
0.0018925572907052643units of Liver (Beef)
0.011214435246144865units of Cabbage
0.005007660466725203units of Spinach
0.061028563526693246units of Navy Beans, Dried

Min cost of Vegie diet is : $39.79866435040896
The vegitarian diet consists of:
0.03545558140888771 units of Wheat Flour (Enriched)
0.008591461668763544 units of Evaporated Milk (can)
0.011249517312443502 units of Cabbage
0.005112832613199646 units of Spinach
0.04862804357316852 units of Navy Beans, Dried
```