Implementing the Alchemist problem with Julia

discussion posted 17 days ago by **jefftb**

Well, since this was such a nicely-posed problem, how could I resist trying to solve it? Here's my implementation in Julia, which yielded the solution of just making gold (20 units) for a max profit of 540. It looks like the critical constraint is the amount of pixie dust, which is true in real life too.

Is this the correct result, i.e. did I make any mistakes below? Any suggestions for improving the code or style (as I'm new to Julia/JuMP)? (Code runs on JuliaBox w/ Julia 0.4.7.)

```
# the alchemist problem, from MITx 15.053 week 4 lecture
# shows how to optimize with setup costs

using JuMP
m = Model()

# Variable:  binary values wi, with wi=1 for element i set up
@variable(m, w[1:3], Bin)

# Variable:  integer value xi, for amount of xi produced
```

Implementing the Alchemist problem with Julia 2

PS3 Final Problem Part E 12

```
@variable(m, 0 <= x[i=1:3] <= 30, Int)

# Constraint: setups
@constraint(m, setupconst[i=1:3], x[i] <= 30*w[i])

# Constraint: total production time
@constraint(m, 2*x[1] + 4*x[2] + 5*x[3] <= 100)
# Constraint: total pixie dust
@constraint(m, 10*x[1] + 5*x[2] + 2*x[3] <= 204)

# objective:  maximize profit
z =  (-500*w[1] + 52*x[1]) + (-400*w[2]+30*x[2]) + (-300*w[3] + 20*x[
@objective(m, Max, z)

solve(m)

println("Max objective: ", getvalue(z))
println(getvalue(w))
println(getvalue(x))
```

Related to: Week 4 / V. The Alchemist's Problem

This post is visible to everyone.

Add a Response

1 response

kqureshi1 Staff

17 days ago

Thanks

Support

Add a comment

Showing all responses

Add a response:

Preview

Submit

Support

Support