Groups

groups.google.com/forum

AMPL Modeling Language ›

problems with sign when calculating reduced cost manually

2 posts by 2 authors

me (Qiaolun Zhang change)

22 May (19 hours ago)

Hello everyone!

I'm solving the virtual network embedding problem with column generation. The problem is converted to a problem related the multi-commodity flow problem.

I'm a little confused with the calculating of the reduced cost.

When we calculate the reduced cost of minimization for example:

minimize cTx

subject to Ax >= b;

When some constraints are in the form of x1 – x2 – x3 + 3x4 ≤ 1, we will convert it to the following constraint:

-x1 + x2 + x3 - 3x4 >= -1

The reduced cost of a variable is the transpose of coefficient times the dual variable.

It seems to me that in cplex(or ampl), there is no such conversion. So when we calculate the reduced cost, we don't need to do the following two steps:

1. times -1 in both sides of the smaller or equal constraint
2. cost minus the transpose of the transformed coefficient times the dual variable

What we only need to do is the following:

> cost minus the transpose of the original coefficient times the dual variable

Is it correct?

I also attached a model file and the run file to check the reduced cost of q. The following is the code to calculate the reduced cost of the file manually.

- c[s,t,p] is the cost of the primal variable.
- The coefficients are just the transpose of the primal problem.

```
for {(s,t) in El_u, p in 1..n_path[s,t]} {


   printf "(%s, %s, %s, %s)\n", s,t,p,c[s,t,p] -sum{(i,j) in Ep_u: (i,j) in path_link[s,t,p]}
linkCapacity[i,j].dual  -  sum{(s1,i) in AE: (s1,i) in path_link[s,t,p]}
flowNodeMapping[s,t,s1,i].dual - flowPath[s,t].dual;;


}
```

Thank you so much!

Best regards,

Zhang Qiaolun

Attachments (3)

data3.dat

2 KB Download

master1.mod

3 KB Download

calculate_rc.run

1 KB Download

Click here to Reply

AMPL Google Group

22 May (9 hours ago)

AMPL converts each constraint to this form:

lower-bound <= sum-of-terms-involving-variables <= upper-bound

Where there is a single comparison operator (the usual case), the sum-of-terms-involving-variables equals the sum of terms on the left minus the sum of terms on the right. For a single <= constraint the lower-bound is -Infinity, for a single >= constraint the upper-bound is +Infinity, and for an = constraint the lower-bound equals the upper-bound.

AMPL's interface to a solver conveys this information in the form that that solver requires. AMPL does not have any control over how the solver might transform the problem after receiving it, but in general the solvers return dual values that are consistent with this form. Thus you can use the formula "variable-cost minus dual values times variable-coefficients" as you propose.

You can check your computation of the reduced cost by comparing it to q[s,t,p].rc, which is the reduced cost that AMPL computes using the dual values returned by the solver.

--
Robert Fourer
am...@googlegroups.com
{#HS:1172851569-77389#}
On Fri, May 22, 2020 at 5:43 AM UTC, AMPL Modeling Language <am...@googlegroups.com> wrote:

> Hello everyone!

I'm solving the virtual network embedding problem with column generation. The problem is converted to a problem related the multi-commodity flow problem.

I'm a little confused with the calculating of the reduced cost.

When we calculate the reduced cost of minimization for example:

minimize cTx

subject to Ax >= b;

When some constraints are in the form of x1 - x2 - x3 + 3x4 ≤ 1, we will convert it to the following constraint:

-x1 + x2 + x3 - 3x4 >= -1

The reduced cost of a variable is the transpose of coefficient times the dual variable.

It seems to me that in cplex(or ampl), there is no such conversion. So when we calculate the reduced cost, we don't need to do the following two steps:

1. times -1 in both sides of the smaller or equal constraint
2. cost minus the transpose of the transformed coefficient times the dual variable

What we only need to do is the following:

> cost minus the transpose of the original coefficient times the dual variable

Is it correct?

I also attached a model file and the run file to check the reduced cost of q. The following is the code to calculate the reduced cost of the file manually.

- c[s,t,p] is the cost of the primal variable.
- The coefficients are just the transpose of the primal problem.

```
for {(s,t) in El_u, p in 1..n_path[s,t]} {

 printf "(%s, %s, %s, %s)\n", s,t,p,c[s,t,p] -sum{(i,j) in Ep_u: (i,j) in path_link[s,t,p]} linkCapacity[i,j].dual - sum{(s1,i) in AE: (s1,i) in path_link[s,t,p]} flowNodeMapping[s,t,s1,i].dual - flowPath[s,t].dual;;

}
```

Thank you so much!

Best regards,

Zhang Qiaolun

--
You received this message because you are subscribed to the Google Groups "AMPL Modeling Language" group.
To unsubscribe from this group and stop receiving emails from it, send an email to am...@googlegroups.com.
To view this discussion on the web visit https://groups.google.com/d/msgid/ampl/4238631b-8a06-4e29-8801-a7ab3c6d72dd%40googlegroups.com.