Tragedy of the Commons Model

Here is a simple model based on Hardin's original narrative of the Tragedy of the Commons.

We consider a community of X cattle ranchers, each grazing a herd of h cattle on the common pasture. Let q be the quality of the pasture, such that the utility of grazing h cattle is hq. Since the pasture is degraded by grazing, we suppose that the quality is of the form

$q = q_0 - dhX,$

where q_0 is the quality of a pristine pasture and d is the degradation per head of cattle.

As Hardin outlines, the utility to a single rancher who adds one more head to the common pasture is

$(h + 1)(q_0 - d(hX + 1))$

so that the rancher's utility gain is

$(h + 1)(q_0 - d(hX + 1)) - hq = (h + 1)(q - d) - hq = q - d(h + 1).$

Assuming d is small enough that $q > 0$, it follows that this utility difference is positive.

This is sufficient to drive Hardin's conclusion, that (as long as they act to maximize this utility function) ranchers will increase their herds even though it leads to degradation of the pasture for all.

This quantitative model also works in a biological context, essentially verbatim: consider a population of X grams of bacteria, say, whose metabolism produces a toxic byproduct h that indirectly reduces their fecundity:

$\frac{dX}{dt} = (hq - m)X,$

where $q = q_0 - dhX$.

If variant populations X' produce variant amounts of byproduct h', then under adaptive dynamics assumptions h will increase, and the population will create an increasingly toxic habitat for itself.

```
# produces: toc.sobj toc.sage.out.tex
from sage.all import *
```

```
from sage.misc.latex import _latex_file_

import dynamicalsystems

ltx = dynamicalsystems.latex_output( 'toc.sage.out.tex' )

class TragedyOfCommonsModel(dynamicalsystems.PopulationDynamicsSystem):
    def __init__(
        self,
        x_indices=None,
        X = dynamicalsystems.indexer('X'),
        h = dynamicalsystems.indexer('h'),
        m = dynamicalsystems.indexer('m'),
        bindings=dynamicalsystems.Bindings()):
        self._indexers = { 'm':m, 'h':h }
        super(TragedyOfCommonsModel,self).__init__(
            [], x_indices, X, bindings=bindings
        )
    def flow(self):
        X = self._population_indexer
        h = self._indexers['h']
        m = self._indexers['m']
        return {
            X[i] : (h[i]*(SR.symbol('q_0') - SR.symbol('d')*sum(h[j]*X[j] for j in self._popul
            for i in self._population_indices
        }

# now that I've defined the general resource-competition model, let's
# create a 1-resource, 1-population instantiation to work with
toc = TragedyOfCommonsModel(
    x_indices=[0],
    X = dynamicalsystems.indexer( lambda i: i and 'X_%s'%i or 'X' ),
    h = dynamicalsystems.indexer( lambda i: i and 'h_%s'%i or 'h' ),
    m = dynamicalsystems.indexer( lambda i: 1 )
);

ltx.write( 'The ToC model:' )
ltx.write_block( toc )

toc_adap = dynamicalsystems.AdaptiveDynamicsModel(
    toc,
    [ toc._indexers['h'] ]
)

ltx.write( 'Whence the adaptive dynamics of $h$ is' )
ltx.write_environment( 'align*', [ r'\\\n  '.join( r'\frac{d%s}{dt} &\propto %s' % (latex(v),

ltx.close()

save_session('toc')
```

The model equations, according to Sage: `[WorkingWiki encountered errors:`
`Error: <ww-make-failed: toc.sage.out.html.tex, /Selection_Gradients/toc.sage.out.html.tex.`
`]`

And the big question is, of course, does evolution of the quadratic term here have a positive direct effect and negative indirect one?

I suspect that actually both are negative, for the same reason as in the MacLev model, because this actually has the same form as the MacLev model.

The gradient of selection on h is q, and increasing even only the patient h intensifies the competition term $a_{ij} = -dh_i h_j$.

The main part of the selection gradient is $\frac{\partial r}{\partial h} = q_0$.