Effective Coverage vs. Lives Saved: A Comparison

In this document, we will find the optimal set of nutritional interventions over space and time, first using each micronutrient’s effect on effective coverage separately, and then using all interventions together to see if they yield different optimal results, using a measure of the number of lives saved from each intervention.

## Summary of Results

import import\_ipynb import os import geopandas as gpd os.chdir(/home/lordflaron/Documents/minimod) import optimization\_work.demewoz\_lives\_saved.lives\_saved\_analysis as lives\_saved import optimization\_work.demewoz\_lives\_saved.folate\_effective\_coverage\_analysis as folate import optimization\_work.demewoz\_lives\_saved.zinc\_effective\_coverage\_analysis as zinc import optimization\_work.demewoz\_lives\_saved.vas\_effective\_coverage\_analysis as vas import pandas as pd

*# First only get the optimal values* lives\_saved\_opt = lives\_saved.models[lives\_saved][None].opt\_df.loc[**lambda** df: df[opt\_vals]>0].groupby([region,time]).sum()[[cumulative\_benefits, cumulative\_costs]].assign(bau\_cost\_per\_ben = lives\_saved.models[lives\_saved][None].cost\_per\_benefit/3) lives\_saved\_high\_opt = lives\_saved.models[lives\_saved\_high][None].opt\_df.loc[**lambda** df: df[opt\_vals]>0].groupby([region,time]).sum()[[cumulative\_benefits, cumulative\_costs]].assign(bau\_cost\_per\_ben = lives\_saved.models[lives\_saved\_high][None].cost\_per\_benefit/3) zinc\_opt = zinc.models[None].opt\_df.loc[**lambda** df: df[opt\_vals]>0].groupby([region,time]).sum()[[cumulative\_benefits, cumulative\_costs]].assign(bau\_cost\_per\_ben = zinc.models[None].cost\_per\_benefit/3) vas\_opt = vas.models[None].opt\_df.loc[**lambda** df: df[opt\_vals]>0].groupby([region,time]).sum()[[cumulative\_benefits, cumulative\_costs]].assign(bau\_cost\_per\_ben = vas.models[None].cost\_per\_benefit/3) folate\_opt = folate.models[None].opt\_df.loc[**lambda** df: df[opt\_vals]>0].groupby([region,time]).sum()[[cumulative\_benefits, cumulative\_costs]].assign(bau\_cost\_per\_ben = folate.models[None].cost\_per\_benefit/3)

*# Get names of optimal interventions* lives\_saved\_name = lives\_saved.models[lives\_saved][None].optimal\_interventions lives\_saved\_high\_name = lives\_saved.models[lives\_saved\_high][None].optimal\_interventions zinc\_name = zinc.models[None].optimal\_interventions vas\_name = vas.models[None].optimal\_interventions folate\_name = folate.models[None].optimal\_interventions

*# Get names of bau* lives\_saved\_bau = lives\_saved.models[lives\_saved][None].bau\_df.index.get\_level\_values(level=intervention).unique().tolist() lives\_saved\_high\_bau = lives\_saved.models[lives\_saved\_high][None].bau\_df.index.get\_level\_values(level=intervention).unique().tolist() zinc\_bau = zinc.models[None].bau\_df.index.get\_level\_values(level=intervention).unique().tolist() vas\_bau = vas.models[None].bau\_df.index.get\_level\_values(level=intervention).unique().tolist() folate\_bau = folate.models[None].bau\_df.index.get\_level\_values(level=intervention).unique().tolist()

opt\_dict = {Zinc\n(Children Eff. Cov.) : zinc\_opt, VA\n(Children Eff. Cov.) : vas\_opt, Folic Acid\n(WRA Eff. Cov.) : folate\_opt, All (Lives Saved) : lives\_saved\_opt, All (Lives Saved, Alt.) : lives\_saved\_high\_opt}

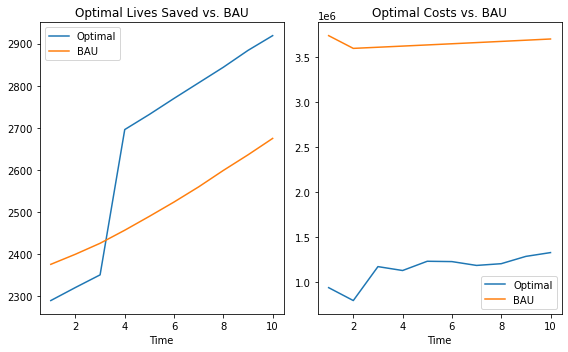
df\_all = pd.concat(opt\_dict.values(), axis=1)

df\_all.columns = pd.MultiIndex.from\_product([opt\_dict.keys(), [CB, CC, BAU\* Cost per Benefit]])

importing Jupyter notebook from /home/lordflaron/Documents/minimod/optimization\_work/demewoz\_lives\_saved/lives\_saved\_analysis.ipynb  
.pipe(observation\_adjustment,  
int1 = "cube",  
int2 = 0,  
time\_to\_replace = [1,2,3])  
.pipe(observation\_adjustment,  
int1 = "cubezcube",  
int2 = 0,  
time\_to\_replace = [1,2,3])  
.pipe(observation\_adjustment,  
int1 = "maxoilcube",  
int2 = "maxoil",  
time\_to\_replace = [1,2,3])  
.pipe(observation\_adjustment,  
int1 = "oilcube",  
int2 = "oil",  
time\_to\_replace = [1,2,3])  
.pipe(observation\_adjustment,  
int1 = "oilcubevas",  
int2 = "oilvas",  
time\_to\_replace = [1,2,3])  
.pipe(observation\_adjustment,  
int1 = "maxoilcubevas",  
int2 = "maxoilvas",  
time\_to\_replace = [1,2,3])  
.pipe(observation\_adjustment,  
int1 = "cubevas",  
int2 = "vas",  
time\_to\_replace = [1,2,3])  
.pipe(observation\_adjustment,  
int1 = "cubeclinic",  
int2 = "clinic",  
time\_to\_replace = [1,2,3])  
.pipe(observation\_adjustment,  
int1 = "maxoilcubeclinic",  
int2 = "maxoilclinic",  
time\_to\_replace = [1,2,3])  
.pipe(observation\_adjustment,  
int1 = "oilcubeclinic",  
int2 = "oilclinic",  
time\_to\_replace = [1,2,3])  
.pipe(observation\_adjustment,  
int1 = "cubezflour",  
int2 = "zflour",  
time\_to\_replace = [1,2,3])  
.pipe(observation\_adjustment,  
int1 = "maxoilcubezflour",  
int2 = "maxoilzflour",  
time\_to\_replace = [1,2,3])  
.pipe(observation\_adjustment,  
int1 = "oilcubezflour",  
int2 = "oilzflour",  
time\_to\_replace = [1,2,3])  
.pipe(observation\_adjustment,  
int1 = "oilcubevaszflour",  
int2 = "oilvaszflour",  
time\_to\_replace = [1,2,3])  
.pipe(observation\_adjustment,  
int1 = "cubevaszflour",  
int2 = "vaszflour",  
time\_to\_replace = [1,2,3])  
.pipe(observation\_adjustment,  
int1 = "maxoilcubevaszflour",  
int2 = "maxoilvaszflour",  
time\_to\_replace = [1,2,3])  
.pipe(observation\_adjustment,  
int1 = "oilcubecliniczflour",  
int2 = "oilcliniczflour",  
time\_to\_replace = [1,2,3])  
.pipe(observation\_adjustment,  
int1 = "maxoilcubecliniczflour",  
int2 = "maxoilcliniczflour",  
time\_to\_replace = [1,2,3])  
.pipe(observation\_adjustment,  
int1 = "cubezflourzcube",  
int2 = "zflour",  
time\_to\_replace = [1,2,3])  
.pipe(observation\_adjustment,  
int1 = "maxoilcubezflourzcube",  
int2 = "maxoilzflour",  
time\_to\_replace = [1,2,3])  
.pipe(observation\_adjustment,  
int1 = "oilcubezflourzcube",  
int2 = "oilzflour",  
time\_to\_replace = [1,2,3])  
.pipe(observation\_adjustment,  
int1 = "oilcubevaszflourzcube",  
int2 = "oilvaszflour",  
time\_to\_replace = [1,2,3])  
.pipe(observation\_adjustment,  
int1 = "oilcubecliniczflourzcube",  
int2 = "oilcliniczflour",  
time\_to\_replace = [1,2,3])  
.pipe(observation\_adjustment,  
int1 = "maxoilcubecliniczflourzcube",  
int2 = "maxoilcliniczflour",  
time\_to\_replace = [1,2,3])  
.pipe(observation\_adjustment,  
int1 = "zcube",  
int2 = 0,  
time\_to\_replace = [1,2,3])  
.pipe(observation\_adjustment,  
int1 = "zflourzcube",  
int2 = "zflour",  
time\_to\_replace = [1,2,3])  
.pipe(observation\_adjustment,  
int1 = "oilvaszflourzcube",  
int2 = "oilvaszflour",  
time\_to\_replace = [1,2,3])  
.pipe(observation\_adjustment,  
int1 = "maxoilcubezflourzcube",  
int2 = "maxoilzflour",  
time\_to\_replace = [1,2,3])  
.pipe(observation\_adjustment,  
int1 = "oilzflourzcube",  
int2 = "oilzflour",  
time\_to\_replace = [1,2,3])  
.pipe(observation\_adjustment,  
int1 = "oilzflourfcubefflour",  
int2 = "oilzflourfflour",  
time\_to\_replace = [1,2,3])  
.pipe(observation\_adjustment,  
int1 = "maxoil",  
int2 = "oil",  
time\_to\_replace = [1,2])  
.pipe(observation\_adjustment,  
int1 = "maxoilcube",  
int2 = "oilcube",  
time\_to\_replace = [1,2])  
.pipe(observation\_adjustment,  
int1 = "maxoilvas",  
int2 = "oilvas",  
time\_to\_replace = [1,2])  
.pipe(observation\_adjustment,  
int1 = "maxoilcubevas",  
int2 = "oilcubevas",  
time\_to\_replace = [1,2])  
.pipe(observation\_adjustment,  
int1 = "maxoilclinic",  
int2 = "oilclinic",  
time\_to\_replace = [1,2])  
.pipe(observation\_adjustment,  
int1 = "maxoilcubeclinic",  
int2 = "oilcubeclinic",  
time\_to\_replace = [1,2])  
.pipe(observation\_adjustment,  
int1 = "maxoilzflour",  
int2 = "oilzflour",  
time\_to\_replace = [1,2])  
.pipe(observation\_adjustment,  
int1 = "maxoilcubezflour",  
int2 = "oilcubezflour",  
time\_to\_replace = [1,2])  
.pipe(observation\_adjustment,  
int1 = "maxoilvaszflour",  
int2 = "oilvaszflour",  
time\_to\_replace = [1,2])  
.pipe(observation\_adjustment,  
int1 = "maxoilcliniczflour",  
int2 = "oilcliniczflour",  
time\_to\_replace = [1,2])  
.pipe(observation\_adjustment,  
int1 = "maxoilcubevaszflour",  
int2 = "oilcubevaszflour",  
time\_to\_replace = [1,2])  
.pipe(observation\_adjustment,  
int1 = "maxoilcubecliniczflour",  
int2 = "oilcubecliniczflour",  
time\_to\_replace = [1,2])  
.pipe(observation\_adjustment,  
int1 = "maxoilzflourzcube",  
int2 = "oilzflourzcube",  
time\_to\_replace = [1,2])  
.pipe(observation\_adjustment,  
int1 = "maxoilcubezflourzcube",  
int2 = "oilcubezflourzcube",  
time\_to\_replace = [1,2])  
.pipe(observation\_adjustment,  
int1 = "maxoilvaszflourzcube",  
int2 = "oilvaszflourzcube",  
time\_to\_replace = [1,2])  
.pipe(observation\_adjustment,  
int1 = "maxoilcliniczflourzcube",  
int2 = "oilcliniczflourzcube",  
time\_to\_replace = [1,2])  
.pipe(observation\_adjustment,  
int1 = "maxoilcubevaszflourzcube",  
int2 = "oilcubevaszflourzcube",  
time\_to\_replace = [1,2])  
.pipe(observation\_adjustment,  
int1 = "maxoilcubecliniczflourzcube",  
int2 = "oilcubecliniczflourzcube",  
time\_to\_replace = [1,2])  
.pipe(observation\_adjustment,  
int1 = "zflourfflour33",  
int2 = 0,  
time\_to\_replace = slice(None))  
.pipe(observation\_adjustment,  
int1 = "fflour33",  
int2 = 0,  
time\_to\_replace = slice(None))  
Changed cube to 0  
Changed cubezcube to 0  
Changed maxoilcube to maxoil  
Changed oilcube to oil  
Changed oilcubevas to oilvas  
Changed maxoilcubevas to maxoilvas  
Changed cubevas to vas  
Changed cubeclinic to clinic  
Changed maxoilcubeclinic to maxoilclinic  
Changed oilcubeclinic to oilclinic  
Changed cubezflour to zflour  
Changed maxoilcubezflour to maxoilzflour  
Changed oilcubezflour to oilzflour  
Changed oilcubevaszflour to oilvaszflour  
Changed cubevaszflour to vaszflour  
Changed maxoilcubevaszflour to maxoilvaszflour  
Changed oilcubecliniczflour to oilcliniczflour  
Changed maxoilcubecliniczflour to maxoilcliniczflour  
Changed cubezflourzcube to zflour  
Changed maxoilcubezflourzcube to maxoilzflour  
Changed oilcubezflourzcube to oilzflour  
Changed oilcubevaszflourzcube to oilvaszflour  
Changed oilcubecliniczflourzcube to oilcliniczflour  
Changed maxoilcubecliniczflourzcube to maxoilcliniczflour  
Changed zcube to 0  
Changed zflourzcube to zflour  
Changed oilvaszflourzcube to oilvaszflour  
Changed maxoilcubezflourzcube to maxoilzflour  
Changed oilzflourzcube to oilzflour  
Changed oilzflourfcubefflour to oilzflourfflour  
Changed maxoil to oil  
Changed maxoilcube to oilcube  
Changed maxoilvas to oilvas  
Changed maxoilcubevas to oilcubevas  
Changed maxoilclinic to oilclinic  
Changed maxoilcubeclinic to oilcubeclinic  
Changed maxoilzflour to oilzflour  
Changed maxoilcubezflour to oilcubezflour  
Changed maxoilvaszflour to oilvaszflour  
Changed maxoilcliniczflour to oilcliniczflour  
Changed maxoilcubevaszflour to oilcubevaszflour  
Changed maxoilcubecliniczflour to oilcubecliniczflour  
Changed maxoilzflourzcube to oilzflourzcube  
Changed maxoilcubezflourzcube to oilcubezflourzcube  
Changed maxoilvaszflourzcube to oilvaszflourzcube  
Changed maxoilcliniczflourzcube to oilcliniczflourzcube  
Changed maxoilcubevaszflourzcube to oilcubevaszflourzcube  
Changed maxoilcubecliniczflourzcube to oilcubecliniczflourzcube  
Changed zflourfflour33 to 0  
Changed fflour33 to 0  
Changed cube to 0  
Changed cubezcube to 0  
Changed maxoilcube to maxoil  
Changed oilcube to oil  
Changed oilcubevas to oilvas  
Changed maxoilcubevas to maxoilvas  
Changed cubevas to vas  
Changed cubeclinic to clinic  
Changed maxoilcubeclinic to maxoilclinic  
Changed oilcubeclinic to oilclinic  
Changed cubezflour to zflour  
Changed maxoilcubezflour to maxoilzflour  
Changed oilcubezflour to oilzflour  
Changed oilcubevaszflour to oilvaszflour  
Changed cubevaszflour to vaszflour  
Changed maxoilcubevaszflour to maxoilvaszflour  
Changed oilcubecliniczflour to oilcliniczflour  
Changed maxoilcubecliniczflour to maxoilcliniczflour  
Changed cubezflourzcube to zflour  
Changed maxoilcubezflourzcube to maxoilzflour  
Changed oilcubezflourzcube to oilzflour  
Changed oilcubevaszflourzcube to oilvaszflour  
Changed oilcubecliniczflourzcube to oilcliniczflour  
Changed maxoilcubecliniczflourzcube to maxoilcliniczflour  
Changed zcube to 0  
Changed zflourzcube to zflour  
Changed oilvaszflourzcube to oilvaszflour  
Changed maxoilcubezflourzcube to maxoilzflour  
Changed oilzflourzcube to oilzflour  
Changed oilzflourfcubefflour to oilzflourfflour  
Changed maxoil to oil  
Changed maxoilcube to oilcube  
Changed maxoilvas to oilvas  
Changed maxoilcubevas to oilcubevas  
Changed maxoilclinic to oilclinic  
Changed maxoilcubeclinic to oilcubeclinic  
Changed maxoilzflour to oilzflour  
Changed maxoilcubezflour to oilcubezflour  
Changed maxoilvaszflour to oilvaszflour  
Changed maxoilcliniczflour to oilcliniczflour  
Changed maxoilcubevaszflour to oilcubevaszflour  
Changed maxoilcubecliniczflour to oilcubecliniczflour  
Changed maxoilzflourzcube to oilzflourzcube  
Changed maxoilcubezflourzcube to oilcubezflourzcube  
Changed maxoilvaszflourzcube to oilvaszflourzcube  
Changed maxoilcliniczflourzcube to oilcliniczflourzcube  
Changed maxoilcubevaszflourzcube to oilcubevaszflourzcube  
Changed maxoilcubecliniczflourzcube to oilcubecliniczflourzcube  
Changed zflourfflour33 to 0  
Changed fflour33 to 0  
Changed cubezflourzcubefcubefflour to zflourfflour  
Changed cubezflourfcubefflour to zflourfflour  
Changed oilzflourzcubefcubefflour to oilzflourfflour  
Changed fcube to 0  
Changed fcube to 0  
Changed oilfcube to oil  
Changed oilcubeclinicfcube to oilclinic  
Changed maxoilzflourfcubefflour to maxoilzflourfflour  
Changed maxoilcliniczflourfcubefflour to maxoilcliniczflourfflour  
Changed maxoilzflourfcubefflour to oilzflourfcubefflour  
Changed maxoilcliniczflourfcubefflour to oilcliniczflourfcubefflour  
Running lives\_saved with None  
[Note]: Processing Data...  
[Note]: Creating Base Model with constraints  
  
 MiniMod Nutrition Intervention Tool  
 Optimization Method: MIN  
 Version: 0.0.6dev  
 Solver: CBC,  
 Show Output: True  
   
   
[Note]: Optimizing...  
[Note]: Optimal Solution Found  
Running lives\_saved\_high with None  
[Note]: Processing Data...  
[Note]: Creating Base Model with constraints  
  
 MiniMod Nutrition Intervention Tool  
 Optimization Method: MIN  
 Version: 0.0.6dev  
 Solver: CBC,  
 Show Output: True  
   
   
[Note]: Optimizing...  
[Note]: Optimal Solution Found  
+----------------------------+----------------------------+  
| MiniMod Solver Results | |  
| Method: | MIN |  
| Solver: | CBC |  
| Optimization Status: | OptimizationStatus.OPTIMAL |  
| Number of Solutions Found: | 1 |  
+----------------------------+----------------------------+  
+-----------------------------+--------+  
| No. of Variables: | 6750 |  
| No. of Integer Variables: | 6750 |  
| No. of Constraints | 1159 |  
| No. of Non-zeros in Constr. | 231504 |  
+-----------------------------+--------+  
Interventions Chosen:  
+-------------------+-----------------+  
| Minimum Benefit | 22019.1 |  
| Objective Bounds | 1.15113e+07 |  
| Total Cost | 1.15113e+07 |  
| Total Lives Saved | 23220.5 |  
+-------------------+-----------------+  
+------------------+--------+  
| Cost per Benefit | 495.74 |  
+------------------+--------+  
+-----------------------------------+--+  
| Total Cost and Benefits over Time | |  
+-----------------------------------+--+  
| time | opt\_vals | opt\_benefit | opt\_costs |  
|-------:|-----------:|--------------:|-----------------:|  
| 1 | 3 | 2290 | 939944 |  
| 2 | 3 | 2321 | 796726 |  
| 3 | 3 | 2351 | 1.17346e+06 |  
| 4 | 3 | 2696 | 1.13025e+06 |  
| 5 | 3 | 2732 | 1.23289e+06 |  
| 6 | 3 | 2770 | 1.22924e+06 |  
| 7 | 3 | 2807 | 1.18669e+06 |  
| 8 | 3 | 2844 | 1.20595e+06 |  
| 9 | 3 | 2884 | 1.28702e+06 |  
| 10 | 3 | 2919 | 1.32918e+06 |  
  
Optimal Interventions  
  
+------------------------------------+-----+-----+-----+-----+-----+-----+-----+-----+-----+------+  
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |  
|------------------------------------+-----+-----+-----+-----+-----+-----+-----+-----+-----+------|  
| ('oilzflourfcubefflour', 'Cities') | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  
| ('oilzflourfcubefflour', 'North') | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  
| ('oilzflourfcubefflour', 'South') | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  
+------------------------------------+-----+-----+-----+-----+-----+-----+-----+-----+-----+------+  
importing Jupyter notebook from /home/lordflaron/Documents/minimod/optimization\_work/demewoz\_lives\_saved/folate\_effective\_coverage\_analysis.ipynb  
Changed fcube to 0  
Changed fflour to fflour33  
[Note]: Processing Data...  
[Note]: Creating Base Model with constraints  
  
 MiniMod Nutrition Intervention Tool  
 Optimization Method: MIN  
 Version: 0.0.6dev  
 Solver: CBC,  
 Show Output: True  
   
   
[Note]: Optimizing...  
[Note]: Optimal Solution Found  
+----------------------------+----------------------------+  
| MiniMod Solver Results | |  
| Method: | MIN |  
| Solver: | CBC |  
| Optimization Status: | OptimizationStatus.OPTIMAL |  
| Number of Solutions Found: | 1 |  
+----------------------------+----------------------------+  
+-----------------------------+------+  
| No. of Variables: | 120 |  
| No. of Integer Variables: | 120 |  
| No. of Constraints | 313 |  
| No. of Non-zeros in Constr. | 1623 |  
+-----------------------------+------+  
Interventions Chosen:  
+----------------------------------------+-------------+  
| Minimum Benefit | 1.45004e+07 |  
| Objective Bounds | 1.55095e+06 |  
| Total Cost | 1.55095e+06 |  
| Total WRA Effectively Covered (Folate) | 1.45004e+07 |  
+----------------------------------------+-------------+  
+------------------+----------+  
| Cost per Benefit | 0.106959 |  
+------------------+----------+  
+-----------------------------------+--+  
| Total Cost and Benefits over Time | |  
+-----------------------------------+--+  
| time | opt\_vals | opt\_benefit | opt\_costs |  
|-------:|-----------:|--------------:|------------:|  
| 1 | 3 | 0 | 286412 |  
| 2 | 3 | 0 | 134017 |  
| 3 | 3 | 1.92897e+06 | 135594 |  
| 4 | 3 | 1.98616e+06 | 137190 |  
| 5 | 3 | 2.04362e+06 | 138805 |  
| 6 | 3 | 2.10248e+06 | 140439 |  
| 7 | 3 | 2.16447e+06 | 142093 |  
| 8 | 3 | 2.22715e+06 | 143766 |  
| 9 | 3 | 2.29016e+06 | 145460 |  
| 10 | 3 | 2.35346e+06 | 147172 |  
  
Optimal Interventions  
  
+------------------------+-----+-----+-----+-----+-----+-----+-----+-----+-----+------+  
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |  
|------------------------+-----+-----+-----+-----+-----+-----+-----+-----+-----+------|  
| ('fflour33', 'Cities') | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  
| ('fflour33', 'North') | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  
| ('fflour33', 'South') | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  
+------------------------+-----+-----+-----+-----+-----+-----+-----+-----+-----+------+  
Running with None  
[Note]: Processing Data...  
[Note]: Creating Base Model with constraints  
  
 MiniMod Nutrition Intervention Tool  
 Optimization Method: MIN  
 Version: 0.0.6dev  
 Solver: CBC,  
 Show Output: True  
   
   
[Note]: Optimizing...  
[Note]: Optimal Solution Found  
Running with time  
[Note]: Processing Data...  
[Note]: Creating Base Model with constraints  
  
 MiniMod Nutrition Intervention Tool  
 Optimization Method: MIN  
 Version: 0.0.6dev  
 Solver: CBC,  
 Show Output: True  
   
   
[Note]: Optimizing...  
[Note]: Optimal Solution Found  
Running with space  
[Note]: Processing Data...  
[Note]: Creating Base Model with constraints  
  
 MiniMod Nutrition Intervention Tool  
 Optimization Method: MIN  
 Version: 0.0.6dev  
 Solver: CBC,  
 Show Output: True  
   
   
[Note]: Optimizing...  
[Note]: Optimal Solution Found  
Running with both  
[Note]: Processing Data...  
[Note]: Creating Base Model with constraints  
  
 MiniMod Nutrition Intervention Tool  
 Optimization Method: MIN  
 Version: 0.0.6dev  
 Solver: CBC,  
 Show Output: True  
   
   
[Note]: Optimizing...  
[Note]: Optimal Solution Found  
+----------------------------+----------------------------+  
| MiniMod Solver Results | |  
| Method: | MIN |  
| Solver: | CBC |  
| Optimization Status: | OptimizationStatus.OPTIMAL |  
| Number of Solutions Found: | 1 |  
+----------------------------+----------------------------+  
+-----------------------------+------+  
| No. of Variables: | 120 |  
| No. of Integer Variables: | 120 |  
| No. of Constraints | 313 |  
| No. of Non-zeros in Constr. | 1623 |  
+-----------------------------+------+  
Interventions Chosen:  
+----------------------------------------+-------------+  
| Minimum Benefit | 1.45004e+07 |  
| Objective Bounds | 1.55095e+06 |  
| Total Cost | 1.55095e+06 |  
| Total WRA Effectively Covered (Folate) | 1.45004e+07 |  
+----------------------------------------+-------------+  
+------------------+----------+  
| Cost per Benefit | 0.106959 |  
+------------------+----------+  
+-----------------------------------+--+  
| Total Cost and Benefits over Time | |  
+-----------------------------------+--+  
| time | opt\_vals | opt\_benefit | opt\_costs |  
|-------:|-----------:|--------------:|------------:|  
| 1 | 3 | 0 | 286412 |  
| 2 | 3 | 0 | 134017 |  
| 3 | 3 | 1.92897e+06 | 135594 |  
| 4 | 3 | 1.98616e+06 | 137190 |  
| 5 | 3 | 2.04362e+06 | 138805 |  
| 6 | 3 | 2.10248e+06 | 140439 |  
| 7 | 3 | 2.16447e+06 | 142093 |  
| 8 | 3 | 2.22715e+06 | 143766 |  
| 9 | 3 | 2.29016e+06 | 145460 |  
| 10 | 3 | 2.35346e+06 | 147172 |  
  
Optimal Interventions  
  
+------------------------+-----+-----+-----+-----+-----+-----+-----+-----+-----+------+  
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |  
|------------------------+-----+-----+-----+-----+-----+-----+-----+-----+-----+------|  
| ('fflour33', 'Cities') | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  
| ('fflour33', 'North') | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  
| ('fflour33', 'South') | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  
+------------------------+-----+-----+-----+-----+-----+-----+-----+-----+-----+------+  
importing Jupyter notebook from /home/lordflaron/Documents/minimod/optimization\_work/demewoz\_lives\_saved/zinc\_effective\_coverage\_analysis.ipynb  
The autoreload extension is already loaded. To reload it, use:  
 %reload\_ext autoreload  
Changed zcube to 0  
Changed zflourzcube to zflour  
[Note]: Processing Data...  
[Note]: Creating Base Model with constraints  
  
 MiniMod Nutrition Intervention Tool  
 Optimization Method: MIN  
 Version: 0.0.6dev  
 Solver: CBC,  
 Show Output: True  
   
   
[Note]: Optimizing...  
[Note]: Optimal Solution Found  
+----------------------------+----------------------------+  
| MiniMod Solver Results | |  
| Method: | MIN |  
| Solver: | CBC |  
| Optimization Status: | OptimizationStatus.OPTIMAL |  
| Number of Solutions Found: | 1 |  
+----------------------------+----------------------------+  
+-----------------------------+------+  
| No. of Variables: | 90 |  
| No. of Integer Variables: | 90 |  
| No. of Constraints | 313 |  
| No. of Non-zeros in Constr. | 1278 |  
+-----------------------------+------+  
Interventions Chosen:  
+-------------------------------------------+-------------+  
| Minimum Benefit | 7.81179e+06 |  
| Objective Bounds | 6.06955e+06 |  
| Total Cost | 6.06955e+06 |  
| Total Children Effectively Covered (Zinc) | 7.81179e+06 |  
+-------------------------------------------+-------------+  
+------------------+----------+  
| Cost per Benefit | 0.776973 |  
+------------------+----------+  
+-----------------------------------+--+  
| Total Cost and Benefits over Time | |  
+-----------------------------------+--+  
| time | opt\_vals | opt\_benefit | opt\_costs |  
|-------:|-----------:|--------------:|------------:|  
| 1 | 3 | 833571 | 494593 |  
| 2 | 3 | 846033 | 343359 |  
| 3 | 3 | 858734 | 641157 |  
| 4 | 3 | 872058 | 588019 |  
| 5 | 3 | 885833 | 680607 |  
| 6 | 3 | 899011 | 666795 |  
| 7 | 3 | 912323 | 613962 |  
| 8 | 3 | 925599 | 622817 |  
| 9 | 3 | 938577 | 693359 |  
| 10 | 3 | 951450 | 724881 |  
  
Optimal Interventions  
  
+----------------------+-----+-----+-----+-----+-----+-----+-----+-----+-----+------+  
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |  
|----------------------+-----+-----+-----+-----+-----+-----+-----+-----+-----+------|  
| ('zflour', 'Cities') | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  
| ('zflour', 'North') | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  
| ('zflour', 'South') | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  
+----------------------+-----+-----+-----+-----+-----+-----+-----+-----+-----+------+  
Running with None  
[Note]: Processing Data...  
[Note]: Creating Base Model with constraints  
  
 MiniMod Nutrition Intervention Tool  
 Optimization Method: MIN  
 Version: 0.0.6dev  
 Solver: CBC,  
 Show Output: True  
   
   
[Note]: Optimizing...  
[Note]: Optimal Solution Found  
Running with time  
[Note]: Processing Data...  
[Note]: Creating Base Model with constraints  
  
 MiniMod Nutrition Intervention Tool  
 Optimization Method: MIN  
 Version: 0.0.6dev  
 Solver: CBC,  
 Show Output: True  
   
   
[Note]: Optimizing...  
[Note]: Optimal Solution Found  
Running with space  
[Note]: Processing Data...  
[Note]: Creating Base Model with constraints  
  
 MiniMod Nutrition Intervention Tool  
 Optimization Method: MIN  
 Version: 0.0.6dev  
 Solver: CBC,  
 Show Output: True  
   
   
[Note]: Optimizing...  
[Note]: Optimal Solution Found  
Running with both  
[Note]: Processing Data...  
[Note]: Creating Base Model with constraints  
  
 MiniMod Nutrition Intervention Tool  
 Optimization Method: MIN  
 Version: 0.0.6dev  
 Solver: CBC,  
 Show Output: True  
   
   
[Note]: Optimizing...  
[Note]: Optimal Solution Found  
+----------------------------+----------------------------+  
| MiniMod Solver Results | |  
| Method: | MIN |  
| Solver: | CBC |  
| Optimization Status: | OptimizationStatus.OPTIMAL |  
| Number of Solutions Found: | 1 |  
+----------------------------+----------------------------+  
+-----------------------------+------+  
| No. of Variables: | 90 |  
| No. of Integer Variables: | 90 |  
| No. of Constraints | 313 |  
| No. of Non-zeros in Constr. | 1278 |  
+-----------------------------+------+  
Interventions Chosen:  
+-------------------------------------------+-------------+  
| Minimum Benefit | 7.81179e+06 |  
| Objective Bounds | 6.06955e+06 |  
| Total Cost | 6.06955e+06 |  
| Total Children Effectively Covered (Zinc) | 7.81179e+06 |  
+-------------------------------------------+-------------+  
+------------------+----------+  
| Cost per Benefit | 0.776973 |  
+------------------+----------+  
+-----------------------------------+--+  
| Total Cost and Benefits over Time | |  
+-----------------------------------+--+  
| time | opt\_vals | opt\_benefit | opt\_costs |  
|-------:|-----------:|--------------:|------------:|  
| 1 | 3 | 833571 | 494593 |  
| 2 | 3 | 846033 | 343359 |  
| 3 | 3 | 858734 | 641157 |  
| 4 | 3 | 872058 | 588019 |  
| 5 | 3 | 885833 | 680607 |  
| 6 | 3 | 899011 | 666795 |  
| 7 | 3 | 912323 | 613962 |  
| 8 | 3 | 925599 | 622817 |  
| 9 | 3 | 938577 | 693359 |  
| 10 | 3 | 951450 | 724881 |  
  
Optimal Interventions  
  
+----------------------+-----+-----+-----+-----+-----+-----+-----+-----+-----+------+  
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |  
|----------------------+-----+-----+-----+-----+-----+-----+-----+-----+-----+------|  
| ('zflour', 'Cities') | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  
| ('zflour', 'North') | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  
| ('zflour', 'South') | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  
+----------------------+-----+-----+-----+-----+-----+-----+-----+-----+-----+------+  
importing Jupyter notebook from /home/lordflaron/Documents/minimod/optimization\_work/demewoz\_lives\_saved/vas\_effective\_coverage\_analysis.ipynb  
Changed cube to 0  
Changed maxoil to oil  
Changed maxoilcube to maxoil  
Changed oilcube to oil  
Changed oilcubevas to oilvas  
Changed maxoilcubevas to maxoilvas  
Changed cubevas to vas  
Changed cubeclinic to clinic  
Changed maxoilcubeclinic to maxoilclinic  
Changed oilcubeclinic to oilclinic  
Changed maxoilcube to oilcube  
Changed maxoilvas to oilvas  
Changed maxoilcubevas to oilcubevas  
Changed maxoilclinic to oilclinic  
Changed maxoilcubeclinic to oilcubeclinic  
[Note]: Processing Data...  
[Note]: Creating Base Model with constraints  
  
 MiniMod Nutrition Intervention Tool  
 Optimization Method: MIN  
 Version: 0.0.6dev  
 Solver: CBC,  
 Show Output: True  
   
   
[Note]: Optimizing...  
[Note]: Optimal Solution Found  
+----------------------------+----------------------------+  
| MiniMod Solver Results | |  
| Method: | MIN |  
| Solver: | CBC |  
| Optimization Status: | OptimizationStatus.OPTIMAL |  
| Number of Solutions Found: | 1 |  
+----------------------------+----------------------------+  
+-----------------------------+------+  
| No. of Variables: | 510 |  
| No. of Integer Variables: | 510 |  
| No. of Constraints | 454 |  
| No. of Non-zeros in Constr. | 6933 |  
+-----------------------------+------+  
Interventions Chosen:  
+-----------------------------------------+-------------+  
| Minimum Benefit | 1.24652e+07 |  
| Objective Bounds | 2.83901e+07 |  
| Total Cost | 2.83901e+07 |  
| Total Children Effectively Covered (VA) | 1.2496e+07 |  
+-----------------------------------------+-------------+  
+------------------+---------+  
| Cost per Benefit | 2.27193 |  
+------------------+---------+  
+-----------------------------------+--+  
| Total Cost and Benefits over Time | |  
+-----------------------------------+--+  
| time | opt\_vals | opt\_benefit | opt\_costs |  
|-------:|-----------:|--------------:|------------:|  
| 1 | 3 | 1.345e+06 | 3.56153e+06 |  
| 2 | 3 | 1.35927e+06 | 3.52776e+06 |  
| 3 | 3 | 1.05806e+06 | 1.72146e+06 |  
| 4 | 3 | 1.4436e+06 | 2.66859e+06 |  
| 5 | 3 | 1.46313e+06 | 2.70103e+06 |  
| 6 | 3 | 1.48317e+06 | 2.79537e+06 |  
| 7 | 3 | 1.50492e+06 | 2.76685e+06 |  
| 8 | 3 | 1.52759e+06 | 2.88426e+06 |  
| 9 | 3 | 1.55068e+06 | 2.89545e+06 |  
| 10 | 3 | 1.57489e+06 | 2.8678e+06 |  
  
Optimal Interventions  
  
+--------------------------+-----+-----+-----+-----+-----+-----+-----+-----+-----+------+  
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |  
|--------------------------+-----+-----+-----+-----+-----+-----+-----+-----+-----+------|  
| ('oilcube', 'Cities') | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  
| ('oilcube', 'South') | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  
| ('oilcubevas', 'Cities') | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  
| ('oilcubevas', 'North') | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  
| ('oilcubevas', 'South') | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  
+--------------------------+-----+-----+-----+-----+-----+-----+-----+-----+-----+------+  
Running with None  
[Note]: Processing Data...  
[Note]: Creating Base Model with constraints  
  
 MiniMod Nutrition Intervention Tool  
 Optimization Method: MIN  
 Version: 0.0.6dev  
 Solver: CBC,  
 Show Output: True  
   
   
[Note]: Optimizing...  
[Note]: Optimal Solution Found  
Running with time  
[Note]: Processing Data...  
[Note]: Creating Base Model with constraints  
  
 MiniMod Nutrition Intervention Tool  
 Optimization Method: MIN  
 Version: 0.0.6dev  
 Solver: CBC,  
 Show Output: True  
   
   
[Note]: Optimizing...  
[Note]: Optimal Solution Found  
Running with space  
[Note]: Processing Data...  
[Note]: Creating Base Model with constraints  
  
 MiniMod Nutrition Intervention Tool  
 Optimization Method: MIN  
 Version: 0.0.6dev  
 Solver: CBC,  
 Show Output: True  
   
   
[Note]: Optimizing...  
[Note]: Optimal Solution Found  
Running with both  
[Note]: Processing Data...  
[Note]: Creating Base Model with constraints  
  
 MiniMod Nutrition Intervention Tool  
 Optimization Method: MIN  
 Version: 0.0.6dev  
 Solver: CBC,  
 Show Output: True  
   
   
[Note]: Optimizing...  
[Note]: Optimal Solution Found  
+----------------------------+----------------------------+  
| MiniMod Solver Results | |  
| Method: | MIN |  
| Solver: | CBC |  
| Optimization Status: | OptimizationStatus.OPTIMAL |  
| Number of Solutions Found: | 1 |  
+----------------------------+----------------------------+  
+-----------------------------+------+  
| No. of Variables: | 510 |  
| No. of Integer Variables: | 510 |  
| No. of Constraints | 454 |  
| No. of Non-zeros in Constr. | 6933 |  
+-----------------------------+------+  
Interventions Chosen:  
+-----------------------------------------+-------------+  
| Minimum Benefit | 1.24652e+07 |  
| Objective Bounds | 2.83901e+07 |  
| Total Cost | 2.83901e+07 |  
| Total Children Effectively Covered (VA) | 1.2496e+07 |  
+-----------------------------------------+-------------+  
+------------------+---------+  
| Cost per Benefit | 2.27193 |  
+------------------+---------+  
+-----------------------------------+--+  
| Total Cost and Benefits over Time | |  
+-----------------------------------+--+  
| time | opt\_vals | opt\_benefit | opt\_costs |  
|-------:|-----------:|--------------:|------------:|  
| 1 | 3 | 1.345e+06 | 3.56153e+06 |  
| 2 | 3 | 1.35927e+06 | 3.52776e+06 |  
| 3 | 3 | 1.05806e+06 | 1.72146e+06 |  
| 4 | 3 | 1.4436e+06 | 2.66859e+06 |  
| 5 | 3 | 1.46313e+06 | 2.70103e+06 |  
| 6 | 3 | 1.48317e+06 | 2.79537e+06 |  
| 7 | 3 | 1.50492e+06 | 2.76685e+06 |  
| 8 | 3 | 1.52759e+06 | 2.88426e+06 |  
| 9 | 3 | 1.55068e+06 | 2.89545e+06 |  
| 10 | 3 | 1.57489e+06 | 2.8678e+06 |  
  
Optimal Interventions  
  
+--------------------------+-----+-----+-----+-----+-----+-----+-----+-----+-----+------+  
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |  
|--------------------------+-----+-----+-----+-----+-----+-----+-----+-----+-----+------|  
| ('oilcube', 'Cities') | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  
| ('oilcube', 'South') | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  
| ('oilcubevas', 'Cities') | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  
| ('oilcubevas', 'North') | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  
| ('oilcubevas', 'South') | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  
+--------------------------+-----+-----+-----+-----+-----+-----+-----+-----+-----+------+

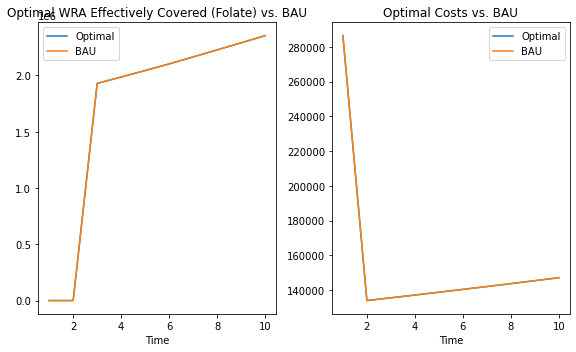
/usr/lib/python3.8/site-packages/geopandas/\_compat.py:84: UserWarning: The Shapely GEOS version (3.8.0-CAPI-1.13.1 ) is incompatible with the GEOS version PyGEOS was compiled with (3.8.1-CAPI-1.13.3). Conversions between both will be slow.  
 warnings.warn(  
/home/lordflaron/Documents/minimod/minimod/utils/plotting.py:250: MatplotlibDeprecationWarning: The 's' parameter of annotate() has been renamed 'text' since Matplotlib 3.3; support for the old name will be dropped two minor releases later.  
 df\_bubble\_final.apply(lambda x: ax.annotate(s = x.bubble\_name,  
/home/lordflaron/Documents/minimod/minimod/utils/plotting.py:250: MatplotlibDeprecationWarning: The 's' parameter of annotate() has been renamed 'text' since Matplotlib 3.3; support for the old name will be dropped two minor releases later.  
 df\_bubble\_final.apply(lambda x: ax.annotate(s = x.bubble\_name,  
/usr/lib/python3.8/site-packages/IPython/core/interactiveshell.py:3417: PerformanceWarning: indexing past lexsort depth may impact performance.  
 exec(code\_obj, self.user\_global\_ns, self.user\_ns)  
/home/lordflaron/Documents/minimod/minimod/utils/plotting.py:250: MatplotlibDeprecationWarning: The 's' parameter of annotate() has been renamed 'text' since Matplotlib 3.3; support for the old name will be dropped two minor releases later.  
 df\_bubble\_final.apply(lambda x: ax.annotate(s = x.bubble\_name,  
/home/lordflaron/Documents/minimod/minimod/utils/plotting.py:250: MatplotlibDeprecationWarning: The 's' parameter of annotate() has been renamed 'text' since Matplotlib 3.3; support for the old name will be dropped two minor releases later.  
 df\_bubble\_final.apply(lambda x: ax.annotate(s = x.bubble\_name,  
/home/lordflaron/Documents/minimod/minimod/utils/plotting.py:250: MatplotlibDeprecationWarning: The 's' parameter of annotate() has been renamed 'text' since Matplotlib 3.3; support for the old name will be dropped two minor releases later.  
 df\_bubble\_final.apply(lambda x: ax.annotate(s = x.bubble\_name,  
/home/lordflaron/Documents/minimod/minimod/utils/plotting.py:250: MatplotlibDeprecationWarning: The 's' parameter of annotate() has been renamed 'text' since Matplotlib 3.3; support for the old name will be dropped two minor releases later.  
 df\_bubble\_final.apply(lambda x: ax.annotate(s = x.bubble\_name,  
/usr/lib/python3.8/site-packages/IPython/core/interactiveshell.py:3417: PerformanceWarning: indexing past lexsort depth may impact performance.  
 exec(code\_obj, self.user\_global\_ns, self.user\_ns)  
/home/lordflaron/Documents/minimod/minimod/utils/plotting.py:250: MatplotlibDeprecationWarning: The 's' parameter of annotate() has been renamed 'text' since Matplotlib 3.3; support for the old name will be dropped two minor releases later.  
 df\_bubble\_final.apply(lambda x: ax.annotate(s = x.bubble\_name,  
/home/lordflaron/Documents/minimod/minimod/utils/plotting.py:250: MatplotlibDeprecationWarning: The 's' parameter of annotate() has been renamed 'text' since Matplotlib 3.3; support for the old name will be dropped two minor releases later.  
 df\_bubble\_final.apply(lambda x: ax.annotate(s = x.bubble\_name,  
/home/lordflaron/Documents/minimod/minimod/utils/plotting.py:250: MatplotlibDeprecationWarning: The 's' parameter of annotate() has been renamed 'text' since Matplotlib 3.3; support for the old name will be dropped two minor releases later.  
 df\_bubble\_final.apply(lambda x: ax.annotate(s = x.bubble\_name,  
/home/lordflaron/Documents/minimod/minimod/utils/plotting.py:250: MatplotlibDeprecationWarning: The 's' parameter of annotate() has been renamed 'text' since Matplotlib 3.3; support for the old name will be dropped two minor releases later.  
 df\_bubble\_final.apply(lambda x: ax.annotate(s = x.bubble\_name,  
/usr/lib/python3.8/site-packages/IPython/core/interactiveshell.py:3417: PerformanceWarning: indexing past lexsort depth may impact performance.  
 exec(code\_obj, self.user\_global\_ns, self.user\_ns)  
/home/lordflaron/Documents/minimod/minimod/utils/plotting.py:250: MatplotlibDeprecationWarning: The 's' parameter of annotate() has been renamed 'text' since Matplotlib 3.3; support for the old name will be dropped two minor releases later.  
 df\_bubble\_final.apply(lambda x: ax.annotate(s = x.bubble\_name,  
/home/lordflaron/Documents/minimod/minimod/utils/plotting.py:250: MatplotlibDeprecationWarning: The 's' parameter of annotate() has been renamed 'text' since Matplotlib 3.3; support for the old name will be dropped two minor releases later.  
 df\_bubble\_final.apply(lambda x: ax.annotate(s = x.bubble\_name,  
---------------------------------------------------------------------------  
TypeError Traceback (most recent call last)  
/usr/lib/python3.8/site-packages/ipywidgets/widgets/interaction.py in update(self, \*args)  
 254 value = widget.get\_interact\_value()  
 255 self.kwargs[widget.\_kwarg] = value  
--> 256 self.result = self.f(\*\*self.kwargs)  
 257 show\_inline\_matplotlib\_plots()  
 258 if self.auto\_display and self.result is not None:  
  
~/Documents/minimod/optimization\_work/demewoz\_lives\_saved/vas\_effective\_coverage\_analysis.ipynb in plot\_map\_model(m, \*\*kwargs)  
  
TypeError: plot\_map\_benchmark() got an unexpected keyword argument 'bau\_intervention\_bubbles'

interactive(children=(Dropdown(description='col', options=('lives\_saved', 'lives\_saved\_high'), value='lives\_sa…  
  
interactive(children=(Dropdown(description='col', options=('lives\_saved', 'lives\_saved\_high'), value='lives\_sa…



interactive(children=(Dropdown(description='col', options=('lives\_saved', 'lives\_saved\_high'), value='lives\_sa…

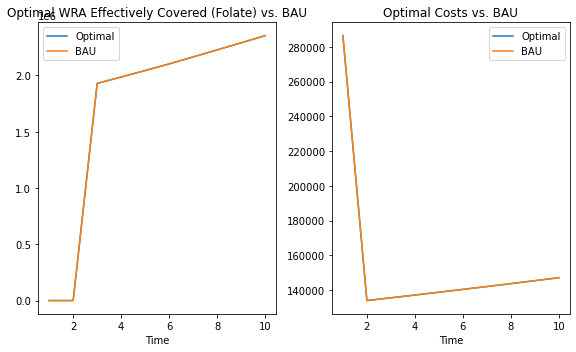




interactive(children=(IntSlider(value=1, description='time', max=10, min=1), Dropdown(description='optimum\_int…

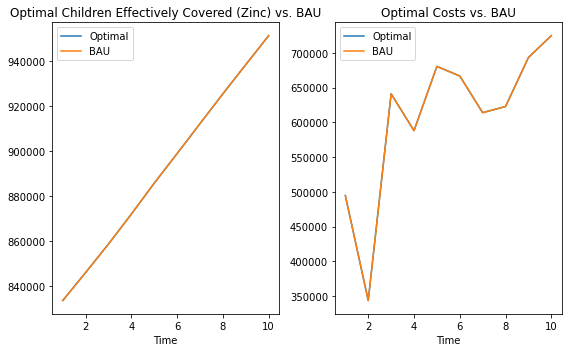


interactive(children=(Dropdown(description='m', options={None: <minimod.solvers.costsolver.CostSolver object a…  
  
interactive(children=(Dropdown(description='m', options={None: <minimod.solvers.costsolver.CostSolver object a…



interactive(children=(Dropdown(description='m', options={None: <minimod.solvers.costsolver.CostSolver object a…

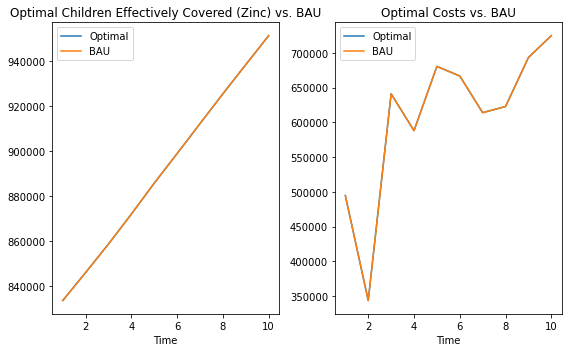




interactive(children=(IntSlider(value=1, description='time', max=10, min=1), Dropdown(description='optimum\_int…

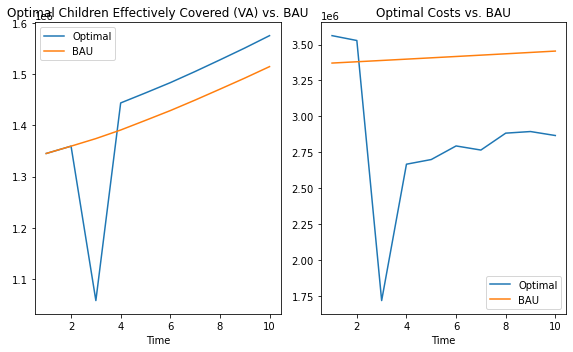


interactive(children=(Dropdown(description='m', options={None: <minimod.solvers.costsolver.CostSolver object a…  
  
interactive(children=(Dropdown(description='m', options={None: <minimod.solvers.costsolver.CostSolver object a…



interactive(children=(Dropdown(description='m', options={None: <minimod.solvers.costsolver.CostSolver object a…

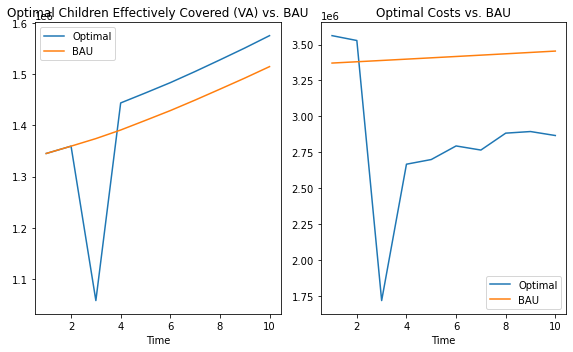




interactive(children=(IntSlider(value=1, description='time', max=10, min=1), Dropdown(description='optimum\_int…



interactive(children=(Dropdown(description='m', options={None: <minimod.solvers.costsolver.CostSolver object a…  
  
interactive(children=(Dropdown(description='m', options={None: <minimod.solvers.costsolver.CostSolver object a…



interactive(children=(Dropdown(description='m', options={None: <minimod.solvers.costsolver.CostSolver object a…

Optimally Chosen Interventions vs. BAU\*. {#tbl:df\_int\_opt}

|  |  |  |  |
| --- | --- | --- | --- |
|  | BAU\* | Optimal Intervention Chosen | Interventions Not Chosen |
|  | BAU\* | Optimal Intervention Chosen | Interventions Not Chosen |
| Zinc (Children Eff. Cov.) | Zinc Flour (95 mg/kg) | Zinc Flour (95 mg/kg) | Zinc Cube 600 mg/kg |
| VA (Children Eff. Cov.) | VA Oil 9 mg/kg + VAS-CHD | VA Oil 12 mg/kg + VAS-CHD (partial) + VA Cube (80 mg/kg) | VA Oil 12 mg/kg, VAS Routine |
| Folic Acid (WRA Eff. Cov.) | Folic Acid Flour 1.65 mg/kg | Folic Acid Flour 1.65 mg/kg | Folic Acid Cube 100 mg/kg |
| All (Lives Saved) | VA Oil 9 mg/kg + VAS-CHD + Zinc Flour 95 mg/kg + Folic Acid Flour 1.65 mg/kg | VA Oil 9 mg/kg + Zinc Flour 95 mg/kg + Folic Acid Flour 5 mg/kg + Folic Acid Cube 100 mg/kg | Zinc Cube 600 mg/kg, VA Oil 12 mg/kg, VAS Routine, Folic Acid Cube 100 mg/kg |
| All (Lives Saved, Alt.) | VA Oil 9 mg/kg + VAS-CHD + Zinc Flour 95 mg/kg + Folic Acid Flour 1.65 mg/kg | VA Oil 9 mg/kg + VAS-CHD (partial) + Zinc Flour (95 mg/kg) + Folic Acid Cube (100 mg/kg) + Folic Acid Flour (5 mg/kg) | Zinc Cube 600 mg/kg, VA Oil 12 mg/kg, VAS Routine, Folic Acid Cube 100 mg/kg |

\*@tbl:df\_int\_opt shows the set of optimal interventions chosen for each micronutrient simulation as well as their Business as Usual Scenario (BAU\*).

The BAU\* was chosen for each simulation so that it would be consistent across effective coverage and lives saved simulations. In the case of the micronutrient effective coverage simulations, the optimal solution is the BAU\* scenario for zinc and folic acid. For vitamin A, in contrast to the BAU\* scenario, fortified boullion cube is also chosen.

When we consider all interventions together, however, the optimal choice becomes different than the BAU\* (which is just a composite of the micronutrient scenarios’ BAU*).*[[1]](#footnote-35) *In contrast to the BAU*, the optimally chosen set of interventions may not include VAS campaign, and may include an addition of folic acid fortified boullion cube. Folic acid fortified flour is included at 1.65 mg/kg, not 5.0 mg/kg.

Note that, although some of the interventions chosen optimally are the same as in the BAU\*, the timing and spatial distribution of each intervention may not be the same. Since MINIMOD chooses the optimal set of interventions across space and time, it may be that the appearance of an intervention may happen earlier or later in time than others and may only occur in certain parts of the country.

The BAU\* scenarios all assume a constant set of interventions across space and time.

We can also see the differences in how each region is affected after 10 years in terms of accumulated benefits and costs.

Total Benefits and Costs Across Space

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | % North | % South | % Cities | National |
|  | CB | 0.26 | 0.33 | 0.41 | 8,923,189.41 |
|  | CC | 0.24 | 0.37 | 0.39 | 6,069,548.82 |
|  | CB | 0.64 | 0.20 | 0.16 | 13,069,585.95 |
|  | CC | 0.64 | 0.20 | 0.16 | 23,736,675.25 |
|  | CB | 0.18 | 0.33 | 0.49 | 17,096,476.30 |
|  | CC | 0.24 | 0.37 | 0.39 | 1,550,946.68 |
|  | CB | 0.37 | 0.35 | 0.28 | 26,614.00 |
|  | CC | 0.27 | 0.33 | 0.40 | 11,511,343.00 |
|  | CB | 0.45 | 0.30 | 0.25 | 32,849.00 |
|  | CC | 0.42 | 0.26 | 0.31 | 14,577,066.00 |
|  |  |  |  |  |  |

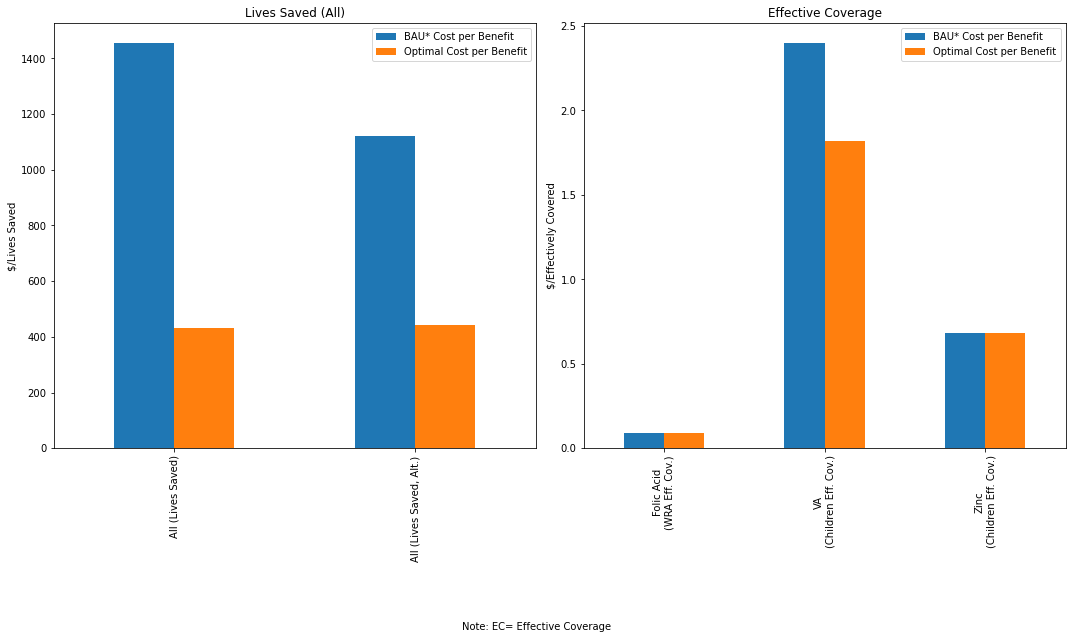
\*@tbl:opt\_space shows the accumulated benefits and costs for each simulation across space. Note that the units of the benefits are dependent on the simulation.

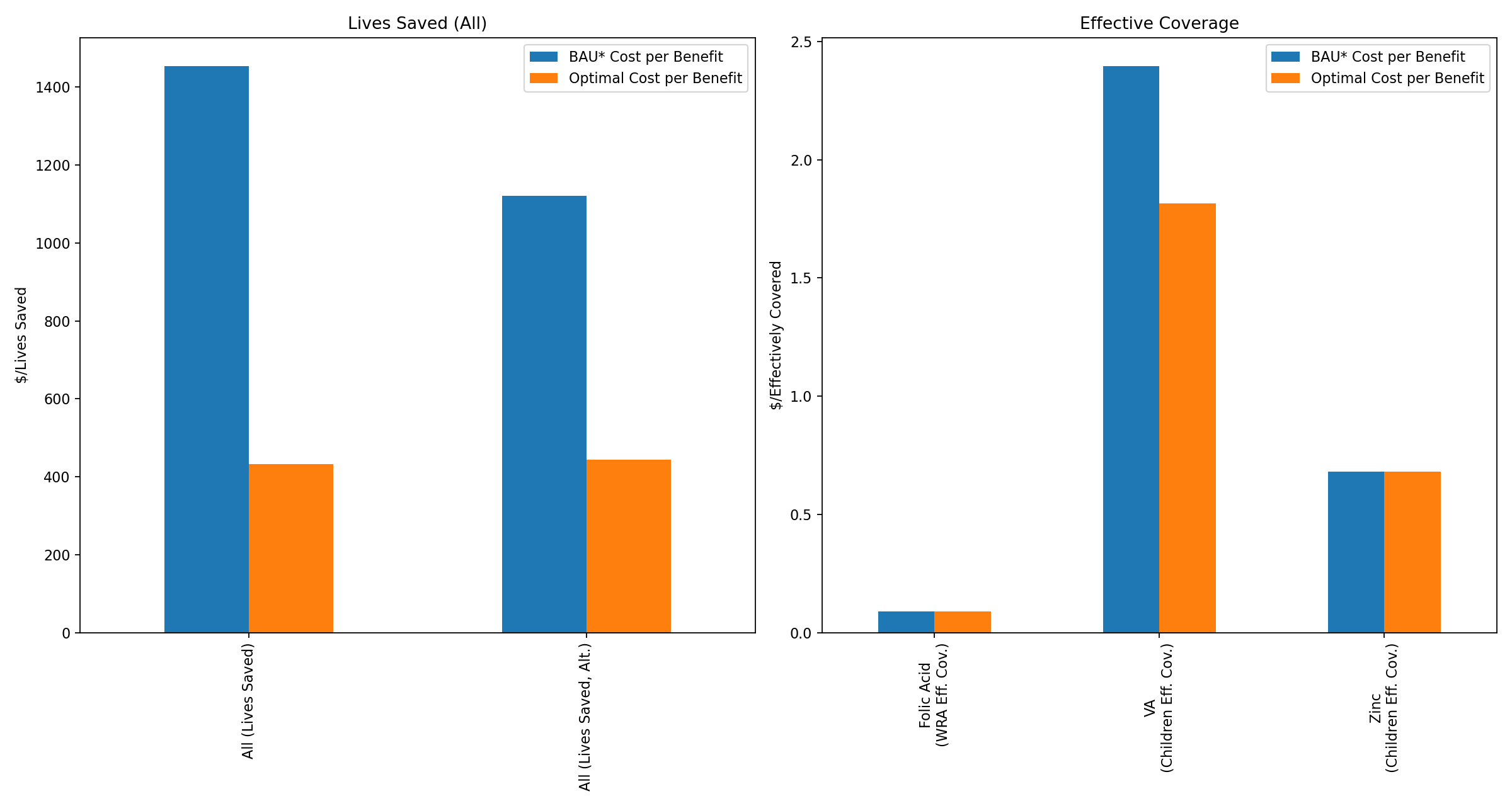
We can also see how benefits accumulate over time, in +@tbl:opt\_time.

Optimal Cumulative Benefits over Time

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Zinc (Children EC) | VA (Children EC) | Folic Acid (WRA EC) | All (LS) | All (LS, Alt.) |
|  | CB | CB | CB | CB | CB |
| **Time** |  |  |  |  |  |
| **1** | 833,571 | 6,011,853 | 0 | 2,290 | 3,741 |
| **2** | 1,679,604 | 7,371,118 | 0 | 4,611 | 7,524 |
| **3** | 2,538,338 | 2,521,598 | 1,928,972 | 6,962 | 11,350 |
| **4** | 3,410,396 | 3,965,198 | 3,915,128 | 9,658 | 14,299 |
| **5** | 4,296,229 | 5,428,330 | 5,958,749 | 12,390 | 17,287 |
| **6** | 5,195,240 | 6,911,502 | 8,061,227 | 15,160 | 20,317 |
| **7** | 6,107,563 | 8,416,423 | 10,225,700 | 17,967 | 23,387 |
| **8** | 7,033,162 | 9,944,016 | 12,452,854 | 20,811 | 26,498 |
| **9** | 7,971,739 | 11,494,698 | 14,743,015 | 23,695 | 29,653 |
| **10** | 8,923,189 | 13,069,586 | 17,096,476 | 26,614 | 32,849 |
|  |  |  |  |  |  |

To illustrate how different these outcomes are to the BAU\* scenarios, +@fig:bau\_comp shows the difference in cost-per child on a bar graph for each simulation, side by side.





In the following sections, we will look at relevant maps that will show us how accumulated benefits evolve through time. For these sections we will use shorthand for interventions, outlined in the following table:

|  |  |
| --- | --- |
| Intervention | Abbreviation |
| Zinc Flour (95 mg/kg) | zflour |
| VA Oil 9mg/kg | oil |
| VAS-CHD | vas |
| VA Cube (80 mg/kg) | cube |
| Folic Acid Cube (100 mg/kg) | fcube |
| Folic Acid Flour (5 mg/kg) | fflour |
| Folic Acid Flour (1.65 mg/kg) | fflour33 |

## Effective Coverage Simulations

### Vitamin A

*# Load data* geo\_df = gpd.read\_file("examples/data/maps/cameroon/CAM.shp")

*# Now we create the boundaries for North, South and Cities* *# Based on "Measuring Costs of Vitamin A..., Table 2"* north = r"Adamaoua|Nord|Extreme-Nord" south = r"Centre|Est|Nord-Ouest|Ouest|Sud|Sud-Ouest" cities= r"Littoral" *# Duala* *# Yaounde is in Mfoundi* geo\_df.loc[**lambda** df: df[ADM1].str.contains(north), space] = North geo\_df.loc[**lambda** df: df[ADM1].str.contains(south), space] = South geo\_df.loc[**lambda** df: df[ADM1].str.contains(cities), space] = Cities geo\_df.loc[**lambda** df: df[ADM2].str.contains(r"Mfoundi"), space] = Cities

*# Now we aggregate the data to the space variable* agg\_geo\_df = geo\_df.dissolve(by = space)

vas.models[None].plot\_map\_benchmark(intervention = None, time = 1, optimum\_interest = cb, bench\_intervention = oilvas, map\_df = agg\_geo\_df, merge\_key = space, intervention\_in\_title = False, intervention\_bubbles= True, intervention\_bubble\_names = [oil, vas, cube], bau\_intervention\_bubble\_names = [oil, vas, cube], save = optimization\_work/demewoz\_lives\_saved/reports/multi\_mn\_plus\_lives\_saved/vas1.png)

plt.gcf().text(0.5,-.05, Note: Effective coverage (in millions), ha=center)

vas.models[None].plot\_map\_benchmark(intervention = None, time = 3, optimum\_interest = cb, bench\_intervention = oilvas, map\_df = agg\_geo\_df, merge\_key = space, intervention\_in\_title = False, intervention\_bubbles= True, intervention\_bubble\_names = [oil, vas, cube], bau\_intervention\_bubble\_names = [oil, vas, cube], save = optimization\_work/demewoz\_lives\_saved/reports/multi\_mn\_plus\_lives\_saved/vas3.png)

plt.gcf().text(0.5,-.05, Note: Effective coverage (in millions), ha=center)

vas.models[None].plot\_map\_benchmark(intervention = None, time = 10, optimum\_interest = cb, bench\_intervention = oilvas, map\_df = agg\_geo\_df, merge\_key = space, intervention\_in\_title = False, intervention\_bubbles= True, intervention\_bubble\_names = [oil, vas, cube], bau\_intervention\_bubble\_names = [oil, vas, cube], save = optimization\_work/demewoz\_lives\_saved/reports/multi\_mn\_plus\_lives\_saved/vas10.png)

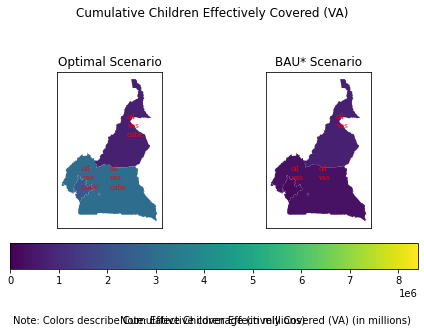
plt.gcf().text(0.5,-.05, Note: Effective coverage (in millions), ha=center)

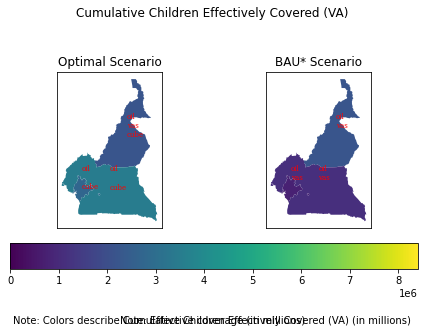
fig, (ax1, ax2) = plt.subplots(1,2, figsize=(10,6))

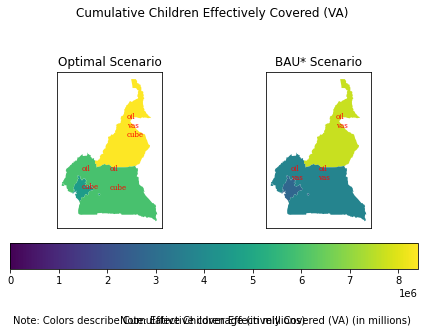
vas.models[None].plot\_bau\_time(b, ax=ax1) vas.models[None].plot\_bau\_time(c, ax=ax2) ax1.set\_ylabel("Millions Effectively Covered") ax2.set\_ylabel("USD (in millions)")

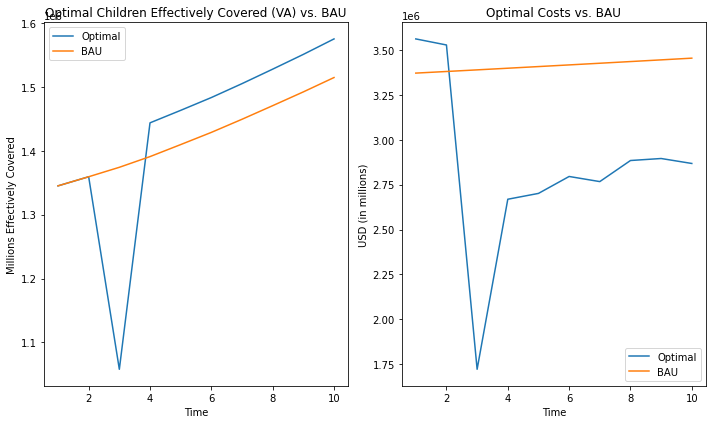
plt.tight\_layout() plt.savefig(optimization\_work/demewoz\_lives\_saved/reports/multi\_mn\_plus\_lives\_saved/vas\_b\_c.png, dpi=160, bbox\_inches=tight)

/home/lordflaron/Documents/minimod/minimod/utils/plotting.py:250: MatplotlibDeprecationWarning: The 's' parameter of annotate() has been renamed 'text' since Matplotlib 3.3; support for the old name will be dropped two minor releases later.  
 df\_bubble\_final.apply(lambda x: ax.annotate(s = x.bubble\_name,  
/home/lordflaron/Documents/minimod/minimod/utils/plotting.py:250: MatplotlibDeprecationWarning: The 's' parameter of annotate() has been renamed 'text' since Matplotlib 3.3; support for the old name will be dropped two minor releases later.  
 df\_bubble\_final.apply(lambda x: ax.annotate(s = x.bubble\_name,  
/home/lordflaron/Documents/minimod/minimod/utils/plotting.py:250: MatplotlibDeprecationWarning: The 's' parameter of annotate() has been renamed 'text' since Matplotlib 3.3; support for the old name will be dropped two minor releases later.  
 df\_bubble\_final.apply(lambda x: ax.annotate(s = x.bubble\_name,  
/home/lordflaron/Documents/minimod/minimod/utils/plotting.py:250: MatplotlibDeprecationWarning: The 's' parameter of annotate() has been renamed 'text' since Matplotlib 3.3; support for the old name will be dropped two minor releases later.  
 df\_bubble\_final.apply(lambda x: ax.annotate(s = x.bubble\_name,  
/home/lordflaron/Documents/minimod/minimod/utils/plotting.py:250: MatplotlibDeprecationWarning: The 's' parameter of annotate() has been renamed 'text' since Matplotlib 3.3; support for the old name will be dropped two minor releases later.  
 df\_bubble\_final.apply(lambda x: ax.annotate(s = x.bubble\_name,  
/home/lordflaron/Documents/minimod/minimod/utils/plotting.py:250: MatplotlibDeprecationWarning: The 's' parameter of annotate() has been renamed 'text' since Matplotlib 3.3; support for the old name will be dropped two minor releases later.  
 df\_bubble\_final.apply(lambda x: ax.annotate(s = x.bubble\_name,

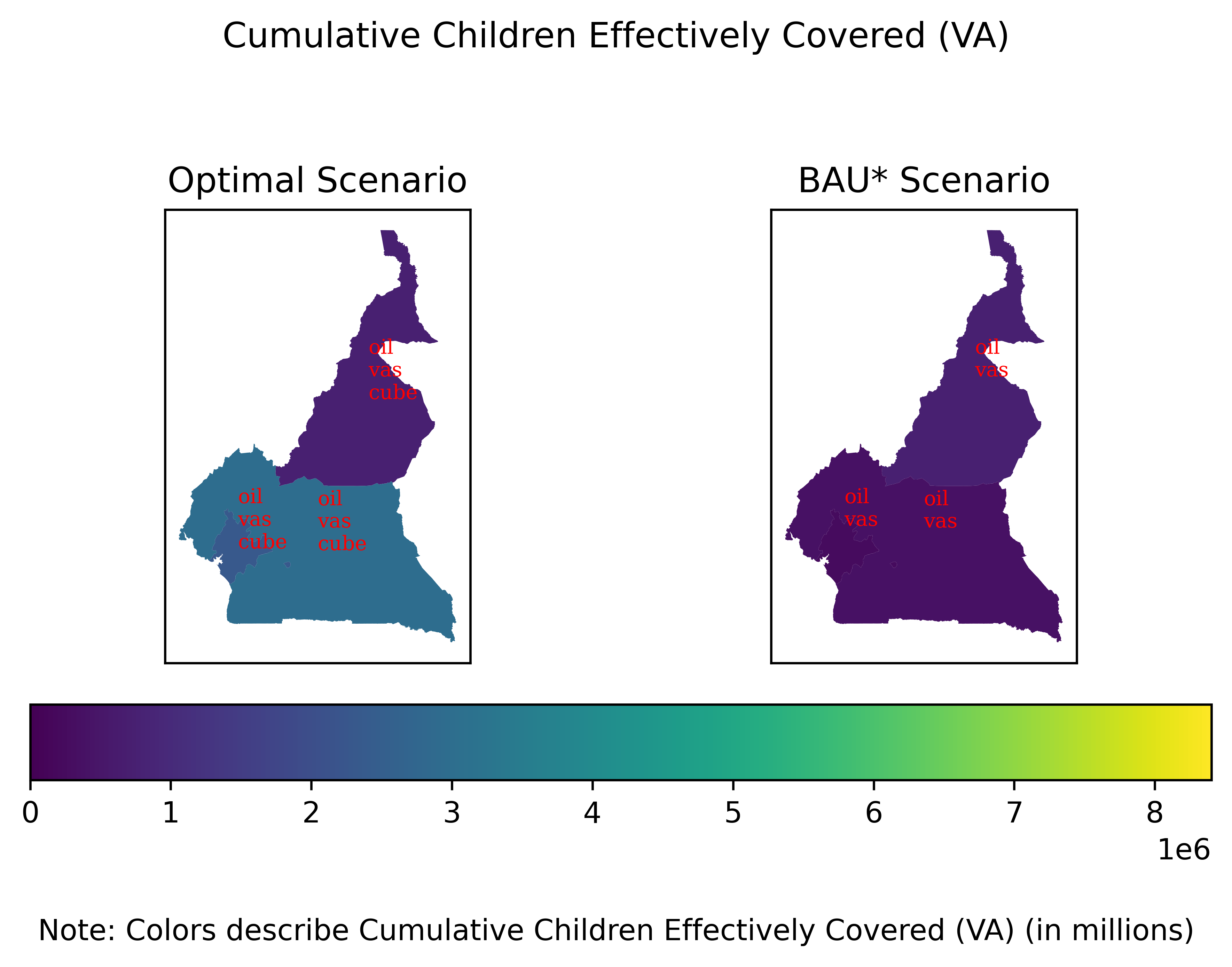


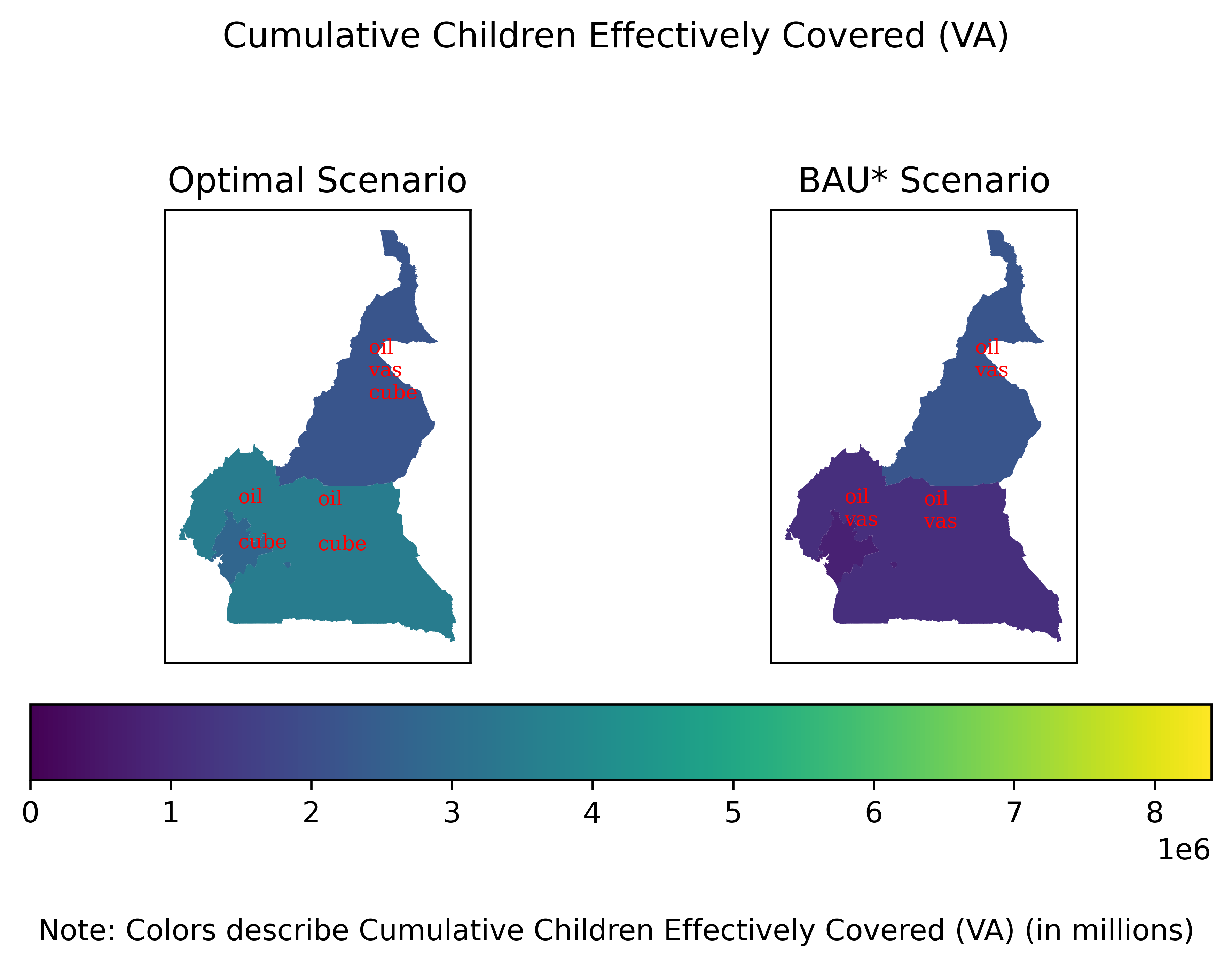


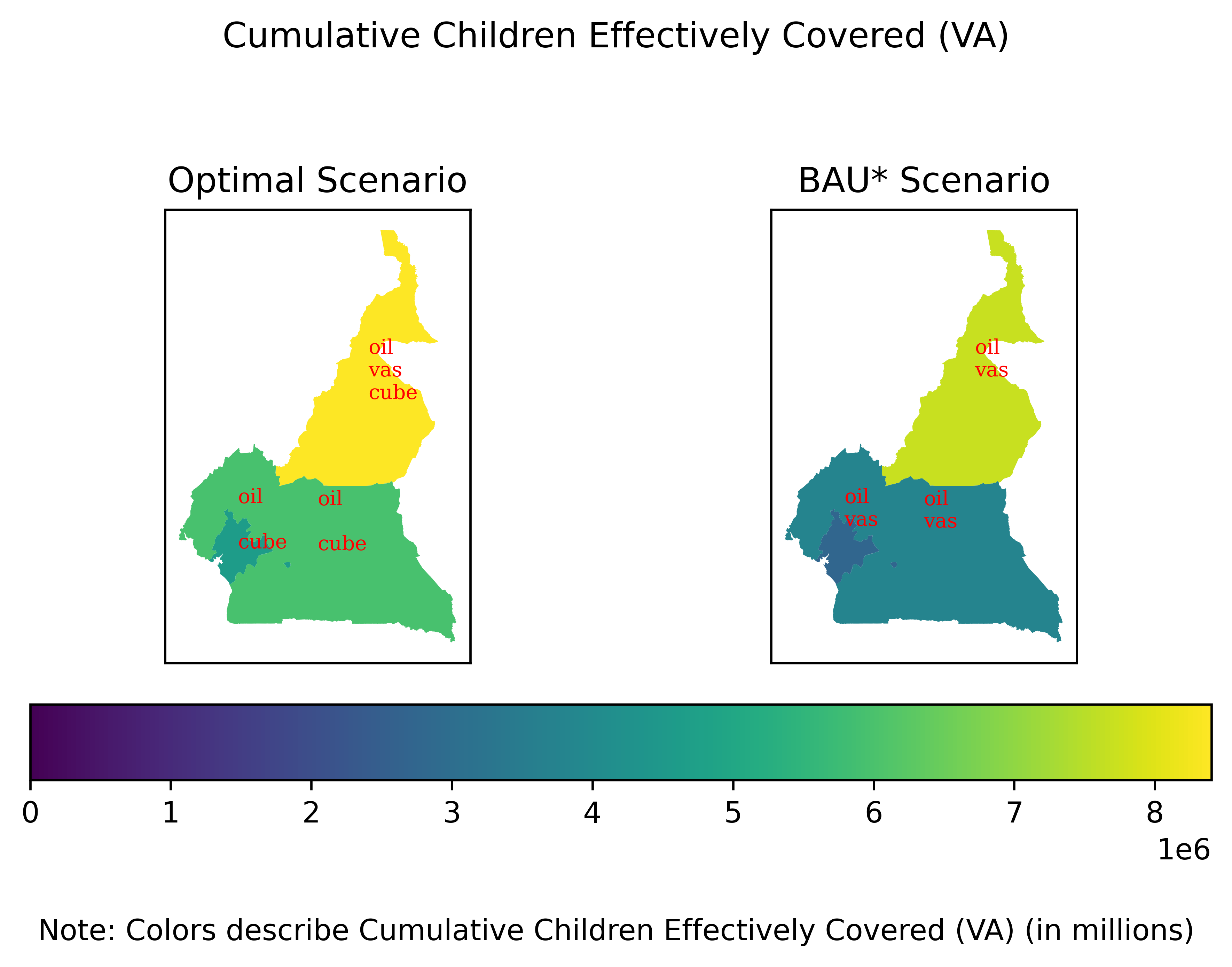




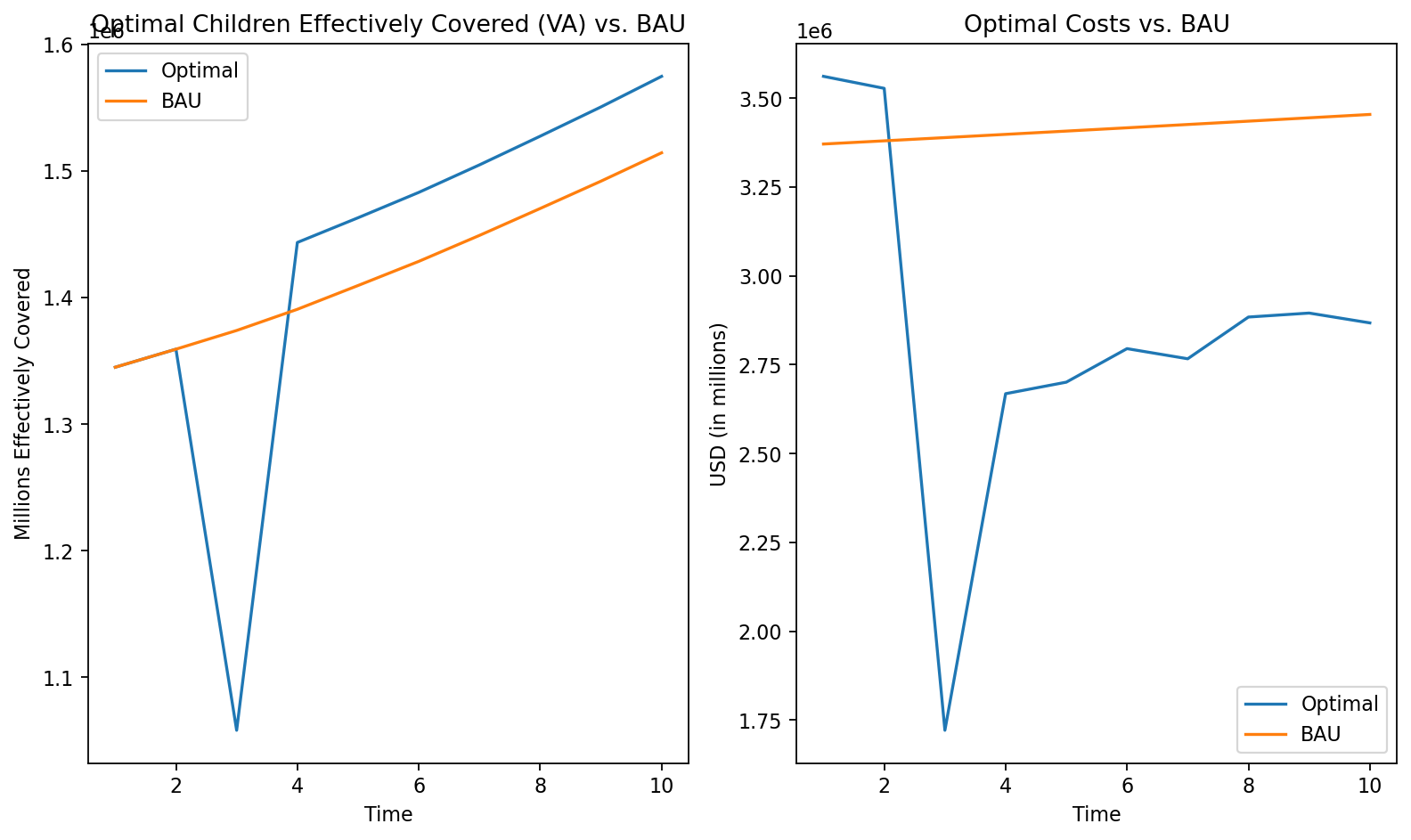
For the Vitamin A simulations, we start with fortified oil (at 75% coverage), fortified cube and a VAS routine as can be seen in +@fig:vas1. By Period 4 (in +@fig:vas3), the south and cities stop VAS campaigns, but proceed with cube and oil interventions. This continues until the last period. If we compare this to the BAU\* scenario, accumulated benefits have reached around the same levels in the South and cities, but with higher benefits in the North.





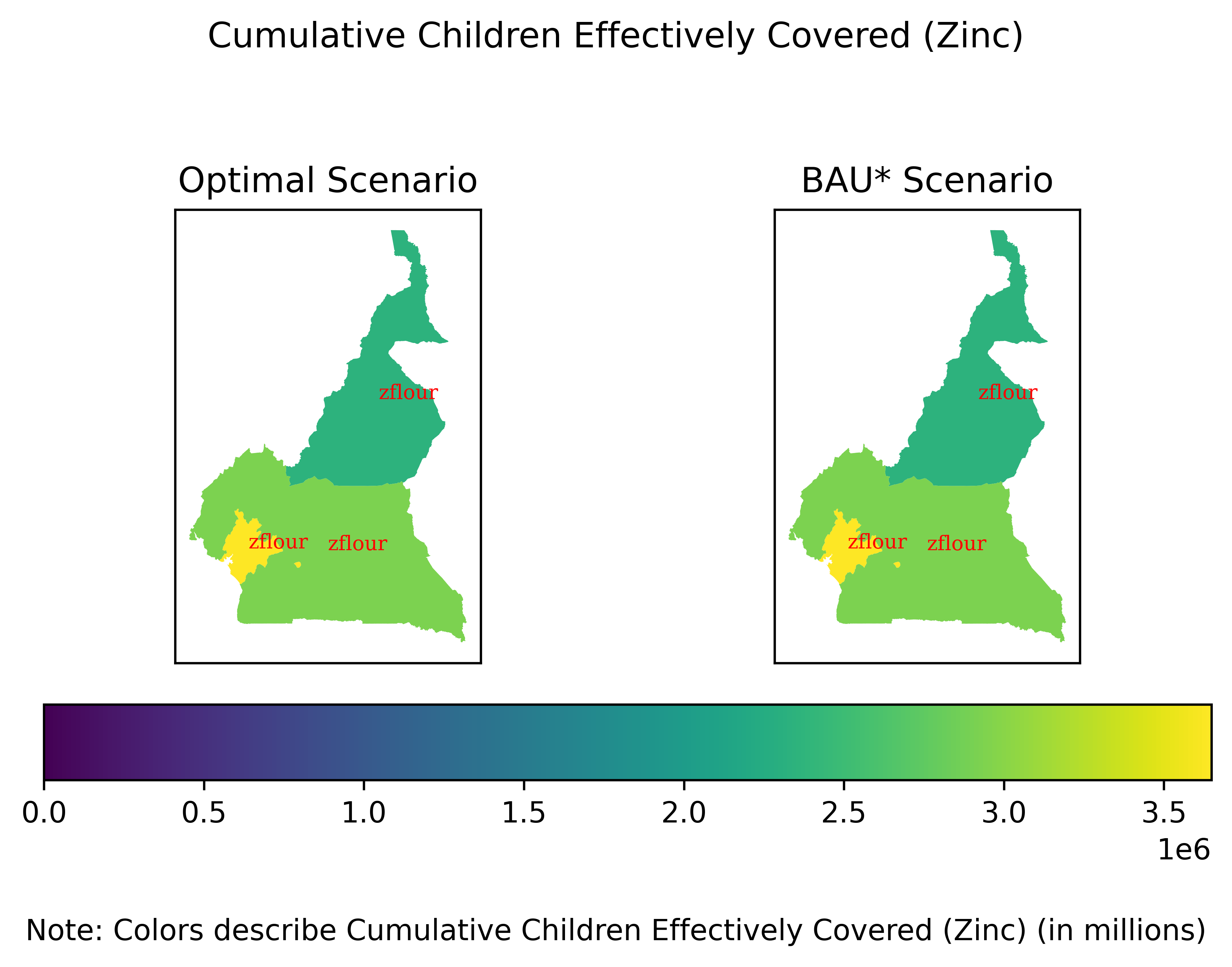


To illustrate the differences between benefits and costs between the optimal and BAU\* scenarios, we compare the two in +@fig:vas\_b\_c.



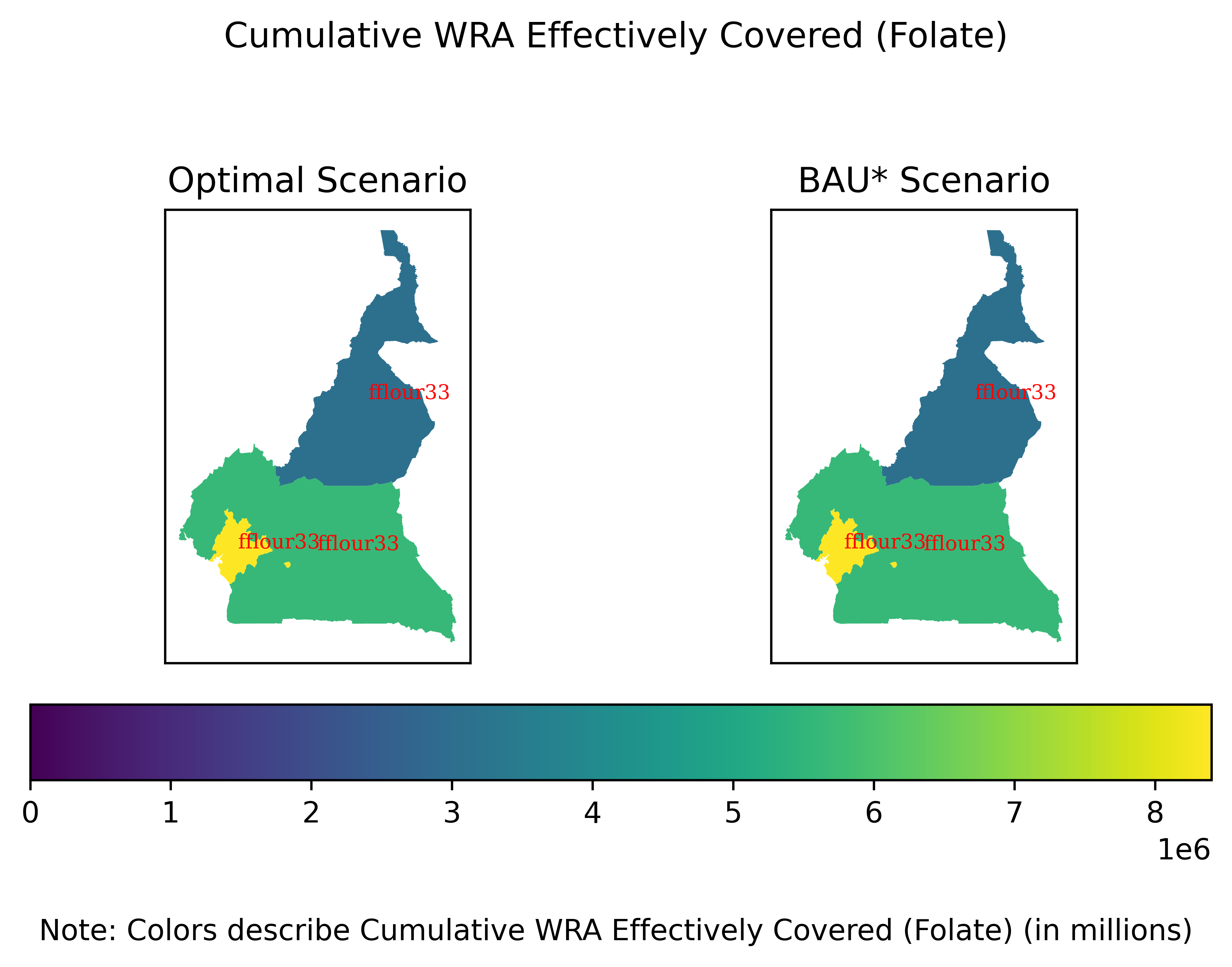
### Zinc

For zinc interventions, we see that zinc fortified flour is chosen, which is the same as the BAU\* scenario. This leads to the highest benefits being in the cities, followed by the south and then the North.



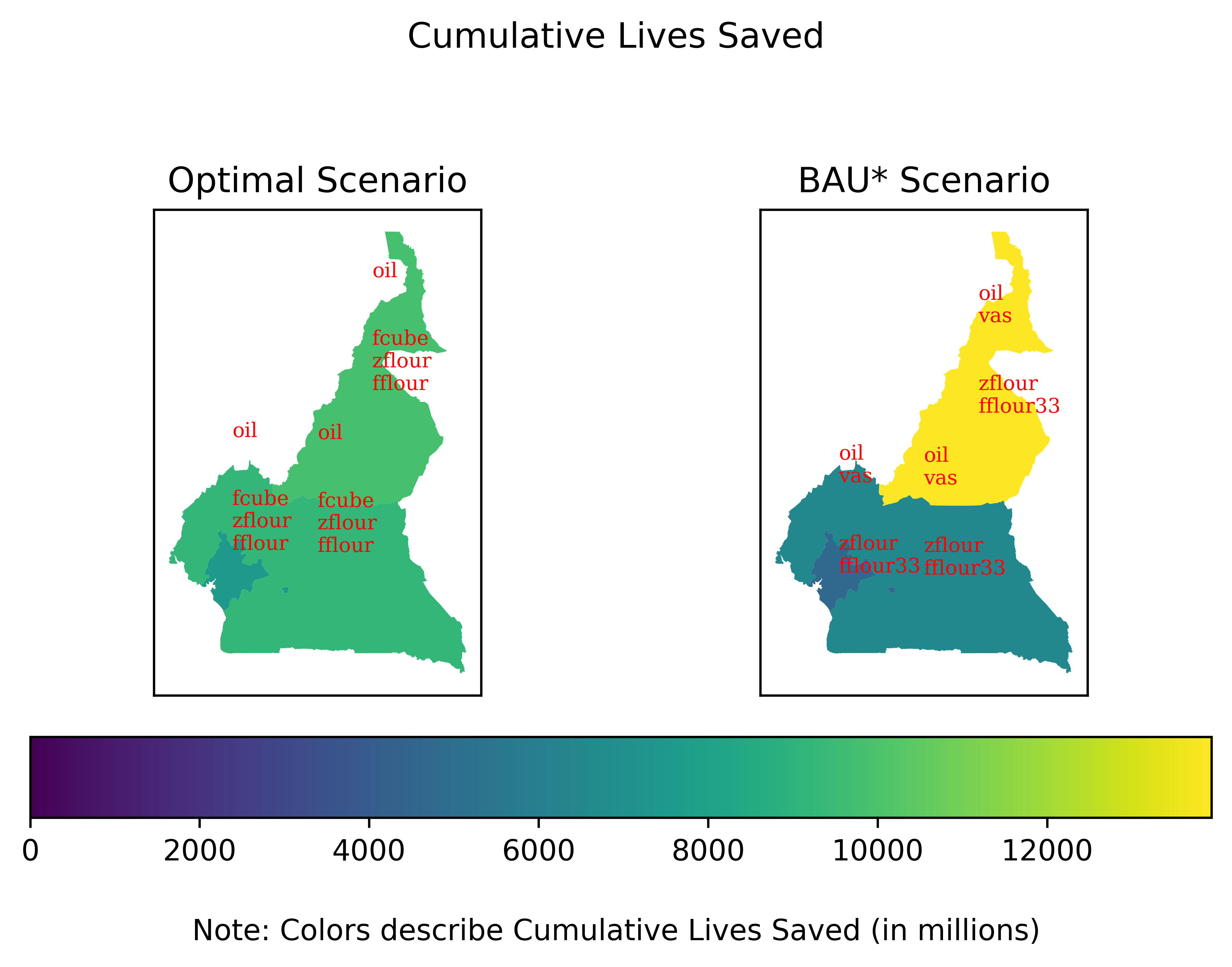
### Folic Acid

Folic Acid is the same as the zinc interventions, in that the BAU\* intervention is chosen as the optimal intervention. Cities are disproportionately affected, compared to the South and the North.

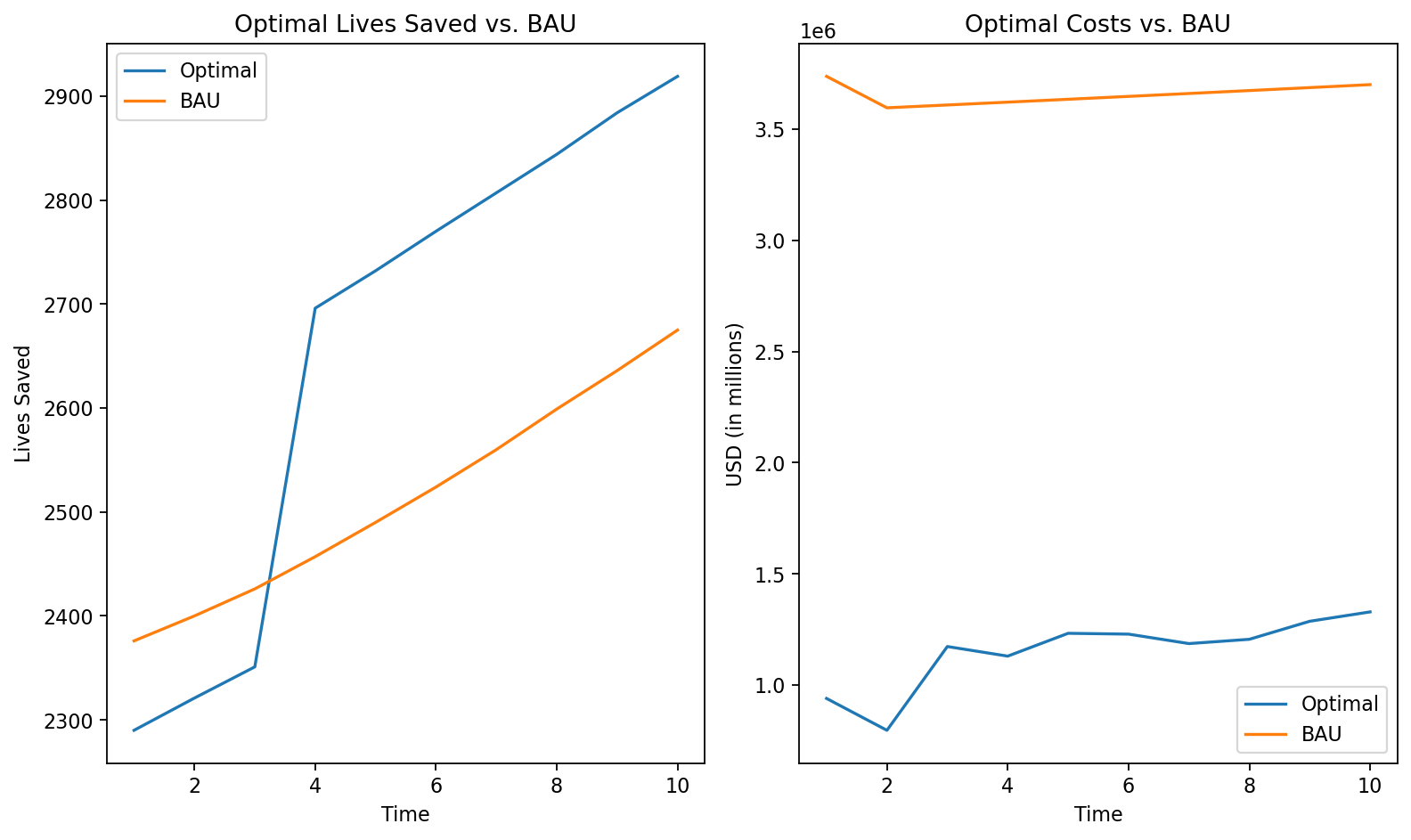


## Lives Saved

For Lives Saved, we find that the same intervention is chosen for all periods everywhere leading to accumulated benefits that are more proportionately distributed across the country, while the BAU\* scenario has higher benefits in the North.

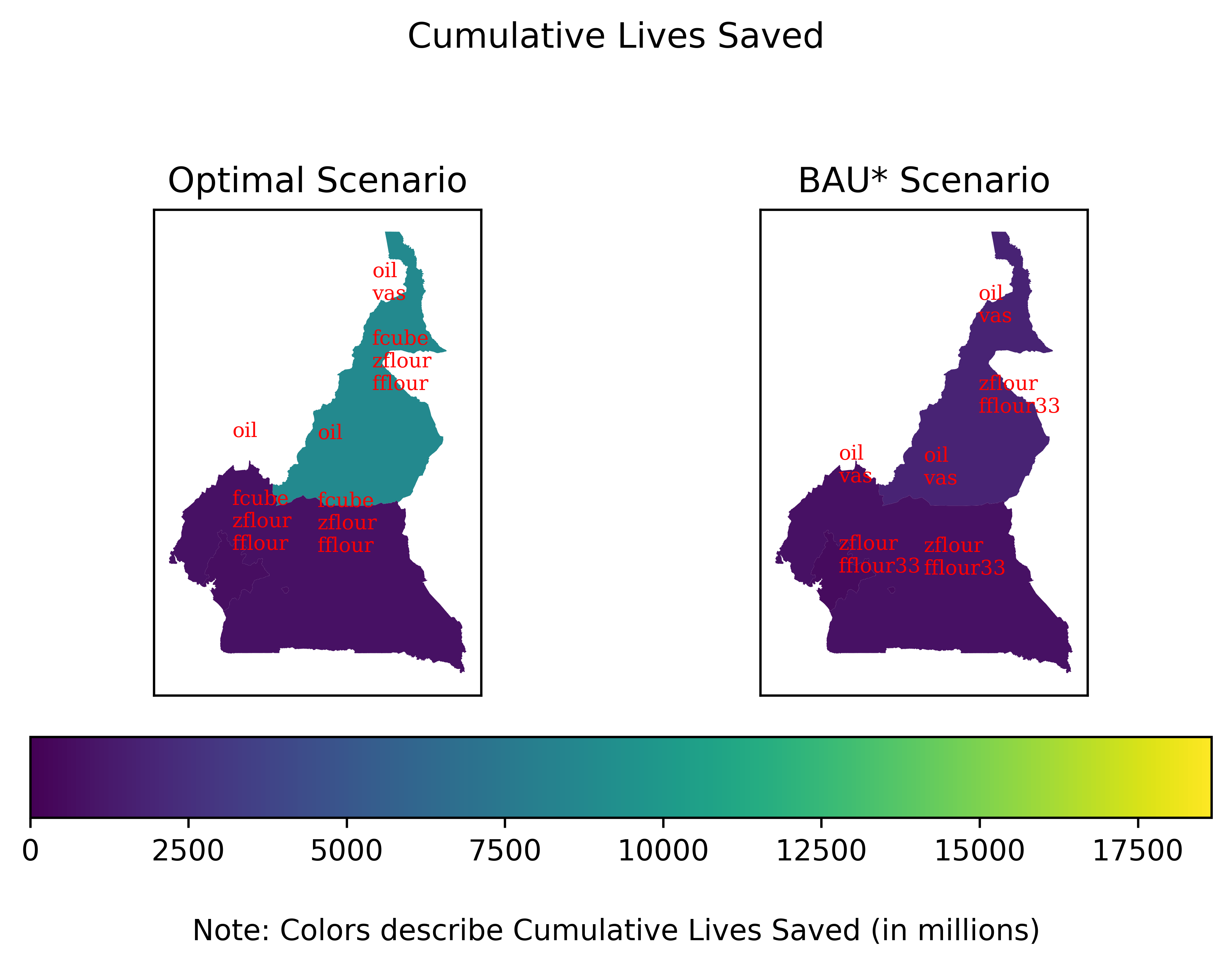


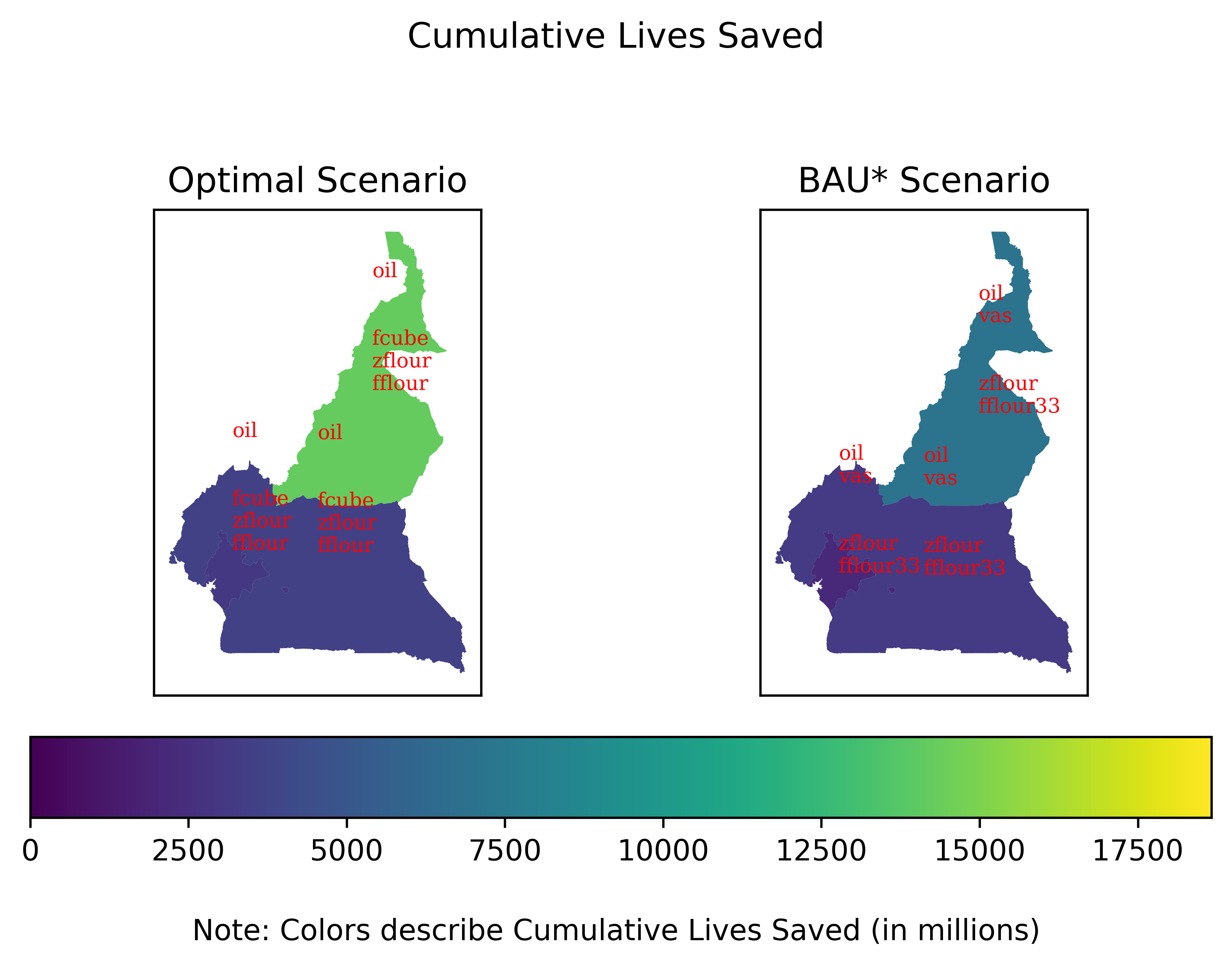
We also show the difference in per-year benefits and costs in +@fig:ls\_b\_c.

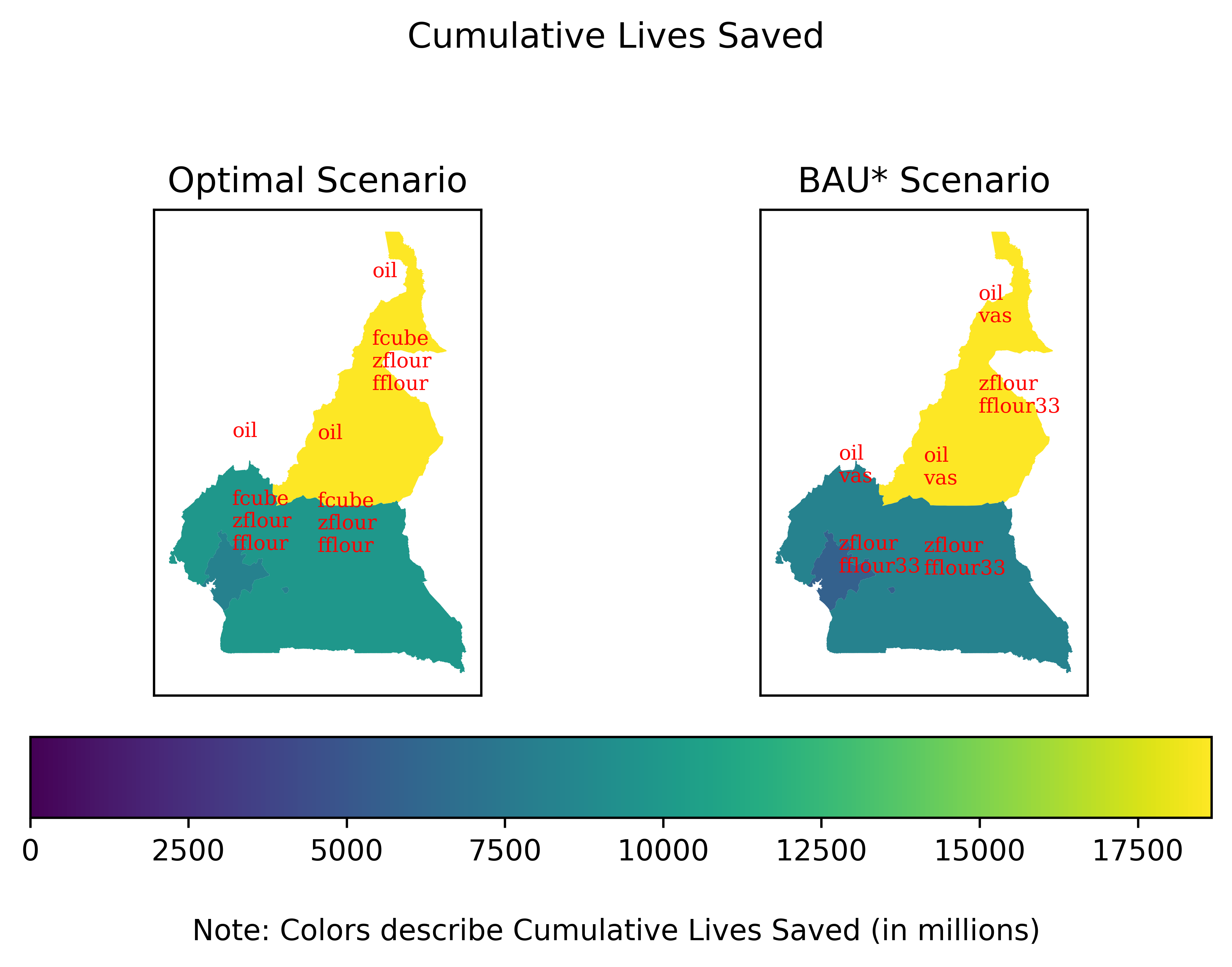


### Lives Saved Alternative Definition

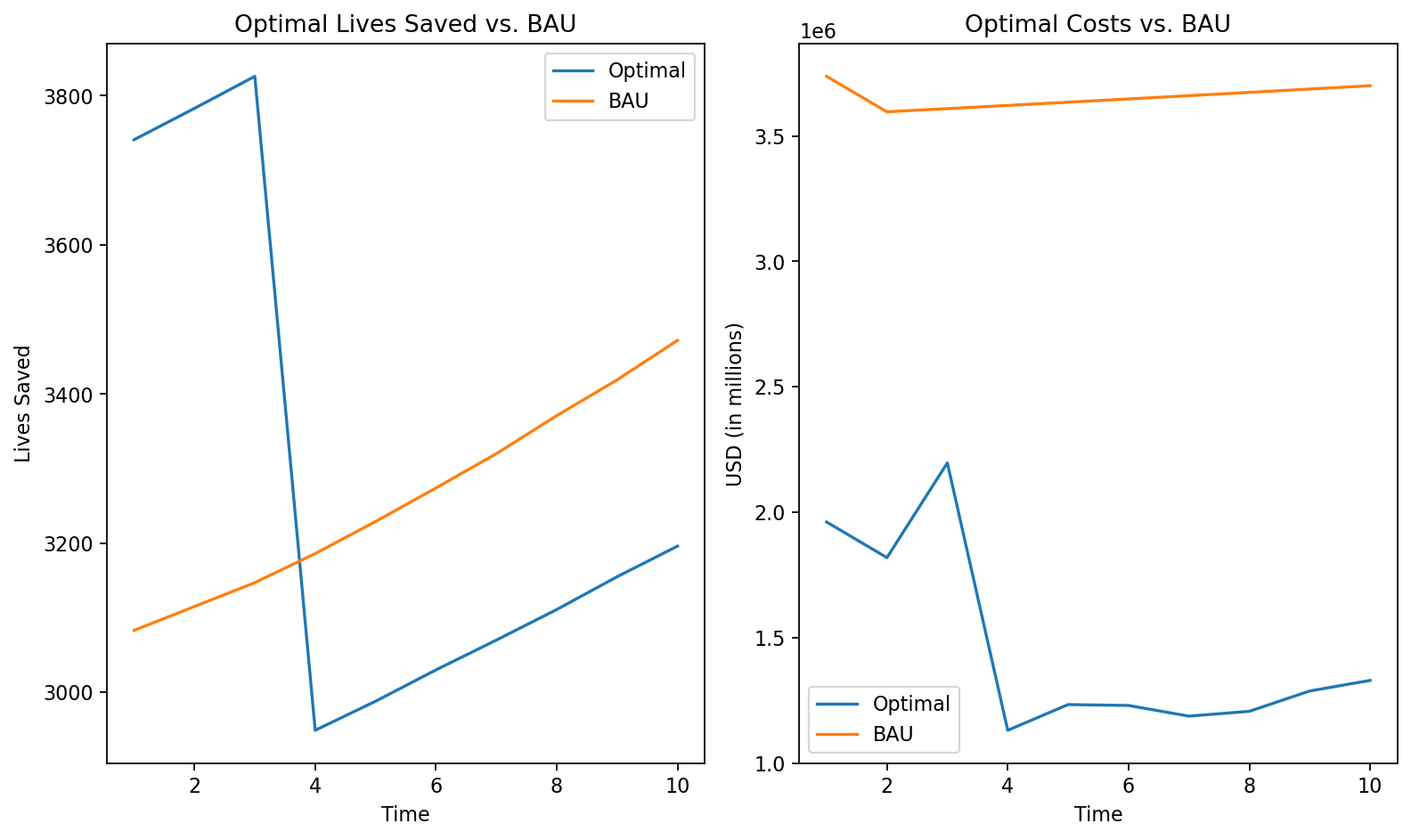
For the alternative definition of lives saved, we find that the same interventions are chosen across the country, apart from VAS campaigns in the north in periods 1-3 (+@fig:lshigh1). By period 4, we find that VAS campaigns stop and for the rest of time, the same interventions are used for all of the country. This leads to a similar geographic distribution by the end (+@fig:lshigh10) as the other lives saved definition, but with relatively higher benefits for the North.







We also show the difference in per-year benefits and costs in +@fig:lshigh\_b\_c.



1. Notice that there are two alternative definitions for the lives saved estimates of each nutritional intervention. The resulting optimally chosen interventions are similar, but the alternative specification includes VAS campaign as well. [↑](#footnote-ref-35)