**Plan for article extension on “Bayesian Optimization with Additive Kernels for the Calibration of Simulation Models to Perform Cost-Effectiveness Analysis”**

**Published article summary:**

1. Introduction
2. Background
   1. Bayesian Optimization (BO), Gaussian Processes, Additive kernels
3. Methodology
   1. Description of simplified simulation model for lung cancer
   2. Optimization methods tested: Nelder-Mead, Simulated Annealing, PSO and BO
   3. Description of hyperparameter tuning used for BO
4. Results
   1. BO takes fewer iterations but more calibration time than the alternatives.
   2. Critical simulation times (i.e. simulation time before BO becomes the fastest calibration method) are reasonably low, but increase with the problem dimensionality.
   3. Additive kernels reduce even further the number of iterations needed.
5. Discussion
   1. In the number of dimensions tested, critical simulation times are reasonable for the models we use and calibration can be a competitive alternative.
   2. Additive kernels can help by increasing the exploration efficiency and needing even fewer iterations than non-additive kernels (if the problem has an additive structure, like ours).
   3. ...

**Suggestions for extension 1: Constrained BO with additive kernels**

So far we haven’t considered the constraints that these simulation models usually have. We can do additional tests by performing the calibration with one simple constraint (e.g. p1 < p2), see how the number of iterations change compared to the result 4.2 and how it affects the critical simulation time. After this, we can expand the number of constraints to 2 and more to see the trend in execution time and number of iterations as the problem becomes more constrained. In order to do this we will have to decide which constraint-aware optimization methods (e.g. PSO) to obtain similar comparison results as those published.

Constrained BO methods [Gardner 2014] can use the constraints to not only avoid choosing invalid solutions but also to guide the exploration process to reach valid solutions in fewer iterations. Despite that fact, we have seen in previous tests (**in other models**) that calibrations that consider several dependent constraints (while assuming independence) can have a hard time converging to valid solutions. If we fail to produce valid solutions in a reasonable time for this model we can try the following approach to break the problem into manageable pieces.

**Suggestions for extension 2: Stepwise (constrained) calibration**

The suggested method is pictured in the attached slides. The motivation is that our simulation models have a sequential time structure of *k* age groups that are simulated in order, so that the logic for each age group influences only the subsequent groups and not the preceding ones. Regarding the parameters, we can differentiate between age-dependent (p1j) and age-independent (p.j) parameters, calibrating them as follows:

p.j := initial values

p1j , ..., pkj := initial values

for i=1 to k

pij‘ := calibrate pij, given p.j and the previously calibrated p1j’ , ..., p(i-1)j’ (ending the simulation at age group i)

p.j’ := calibrate p.j given p1j’ , ..., pkj’

p.j’’ , p1j’’ , ..., pkj’’ := calibrate all parameters with initial guess p.j’ , p1j’ , ..., pkj’ (fine-tuning)

There is no guarantee that this method finds optimal solutions due to its greedy nature. Nevertheless, we can have some intuitive reasons on why this approximation might work well for our simulation models:

* These optimization subtasks have an inherent order in the simulation (age groups), so the chronological stepwise approach is logically consistent.
* Each subtask might impose suboptimal choices for the following subtasks and their parameters, but the additive structure in our model (result 4.3) seems to indicate that the interactions between parameters might not be too complex so the error introduced might be low.
* Smaller sub-optimization tasks are well-suited for BO due to a lower dimensionality, reducing the overall calibration time (and hopefully finding similarly good solutions as well).
* Since dealing with many (dependent) constraints might be challenging, breaking down the problem in subtasks might allow us to consider smaller subsets of restrictions. This might allow us to avoid or reduce the potential problems with multiple dependent constraints discussed above.