

INTRODUCTION

Need to make decisions based on stochastic data, implicitly defined by stochastic simulations.

- **Example:** how to use wind power as a backup power supply with the variability of wind and the uncertainty of sufficient power generation
- **Challenge:** making decisions over controllable parameters, when the computation of even one outcome may involve a computationally expensive stochastic simulation

Relational Databases

Store and manipulate large amounts of data efficiently

Probabilistic Databases

Databases capable of managing large amounts of uncertain data with each tuple having a probability associated with it corresponding to its likelihood.

Statistical Databases

Usually relational databases containing statistical data that have been extended to allow for advanced statistical analysis techniques.

Problems:

- Probabilistic and statistical databases deal with explicit probability distributions, whereas for many classes of problems the probability distributions are complex and defined implicitly through stochastic simulation

Stochastic Simulations

Allow users to create a complex model to simulate an event and serves as an implicitly defined stochastic model.

- **Problems:**
 - Time consuming to make decisions based on trial and error, even after applying heuristics to speed up the process
 - Need to separately express the space of controllable variables – the search space

Our Approach: – an SQL-like language and system for

- Easy stochastic decision problem formulation
- (Optimally) efficient computation of top-k answers using simulation budget optimization

SimQL: Syntax, Semantics, and Canonical Implementation

- Syntax: extension of SQL with stochastic simulation functions and Exp, Variance, Prob(condition), Prob_topK(k)
- Example: Output of electricity from the wind based on the average annual wind speed along with other deterministic attributes can be defined as:

```
CREATE OR REPLACE VIEW simql.vwWndPwr AS
SELECT wind_power_id, location, annual_avg_speed,
       simpleSim (annual_avg_speed) AS generated_kWH
FROM simql.wind_power_info;
```

Where

- wind_power_id, location and annual_avg_speed come from the table wind_power_info
- simpleSim is a stochastic simulation defined as a Java stored procedure and takes the annual_avg_speed as input

The next query computes the probability of the tuple being one of the top-k answers: **SimQL**

```
SELECT location, annual_avg_speed,
       simql.Prob_topk('generated_kWH', 'vwWndPwr',
                     wind_power_id,10)
FROM simql.vwWndPwr;
```

Algorithm for Top-k Answers Based on Simulation Budget Optimization

Input:

C – total computation budget

DB – a budget delta, i.e., constant part of the budget to be used in one iteration

P_D the desired statistical confidence (e.g., 95%) that the top-k selection is correct

Structures:

B – portion of the budget consumed so far

P_B – the statistical confidence that the current top-k selection is correct

Initialization:

Step 1: Each tuple (S_1, S_2, \dots, S_t) is sampled (i.e., simulation run on it) with a CONSTANT computational budget (which is part of C)

Step 2: Compute the confidence P_B that the current top-k tuples are correct. Compute the consumed budget B used by the initialization process.

While $P_B < P_D$, and $B < C$,

Step 3: Allocate the delta budget DB to each tuple proportional to:

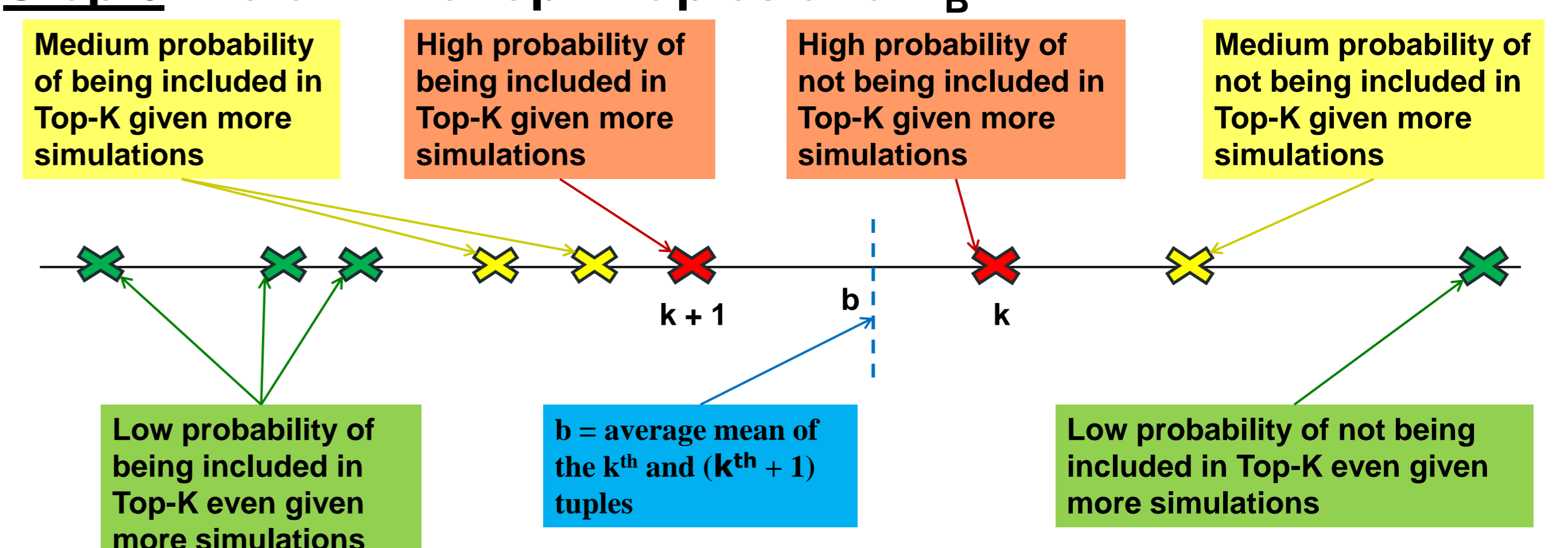
$$\frac{\sigma^2/n}{(b - \text{mean})^2}$$

Step 4: Run simulations for each tuple within allocated budget

Step 5: Recompute P_B and B.

End of While loop

Step 6: Return the top-k tuples and P_B .



CONCLUSION

• **Claim:** the Algorithm is computationally optimal in the sense that:

- In each iteration the probability of having selected the top-k is maximal within the delta budget DB
- Given fixed DB and P_D , and unlimited overall budget, the overall computational time is minimal
- Given fixed DB and the total budget C, and $P_D=1$, the algorithm guarantees the maximal probability of correct selection of top-k

• SimQL can be applied to many areas where decision making is based on traditional simulation techniques, such as emergency responder scenarios and manufacturing.

• Future work on SimQL will include budget optimizations for special cases of problems.

References

1. Alexander Brodsky, X. Sean Wang, "Decision-Guidance Management Systems (DGMS): Seamless Integration of Data Acquisition, Learning, Prediction and Optimization," Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008), 2008.
2. Chen, C. H., He, D., Fu, M. C., and Lee, L. H. 2008. Efficient simulation budget allocation for selecting an optimal subset. *INFORMS J. Comput.* 20, 579-595.