

A global feature selection algorithm for the model selection step in the identification of cell signaling networks

Gustavo Estrela^{1,2,3}, Lulu Wu^{1,2}, Vincent Noël^{1,3}, Carlos Eduardo Ferreira², Hugo A. Armelin^{1,3}, Marco Dimas Gubitoso², Junior Barrera^{1,2}, and Marcelo S. Reis¹

¹Center of Toxins, Immune-response and Cell Signaling (CeTICS), Instituto Butantan, Brazil

²Instituto de Matemática e Estatística, Universidade de São Paulo, Brazil

³Laboratório Especial de Ciclo Celular (LECC), Instituto Butantan, Brazil

⁴Instituto de Química, Universidade de São Paulo, Brazil



Motivation

In the context of machine learning, feature selection problem consists in choosing a subset of features that best explains the classification with minimum redundancy. The space of solutions of this problem induces a boolean lattice and the cost function commonly describes U shaped curves on chains of this lattice, caused by growth of estimation errors as we add more features.

The U shaped curves justifies the reduction to the U-Curve problem: a special case of the feature selection problem where every chain of the search space describe U shaped curves. Many algorithms in the literature exploit this reduction and yet they show limitations regarding scalability, and that shows the need for new approaches on solving the U-Curve problem. To this end, we developed the Parallel U-Curve Search (PUCS).

Simulations

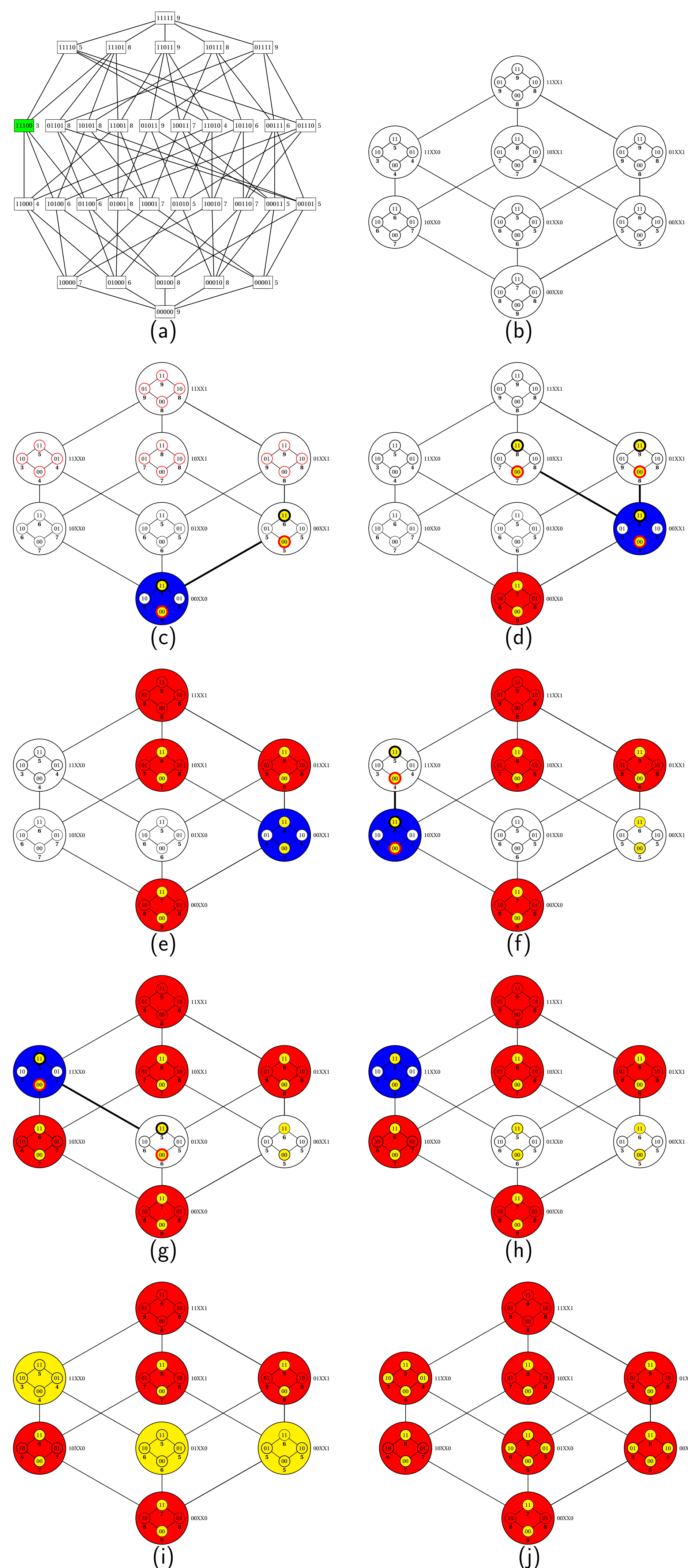


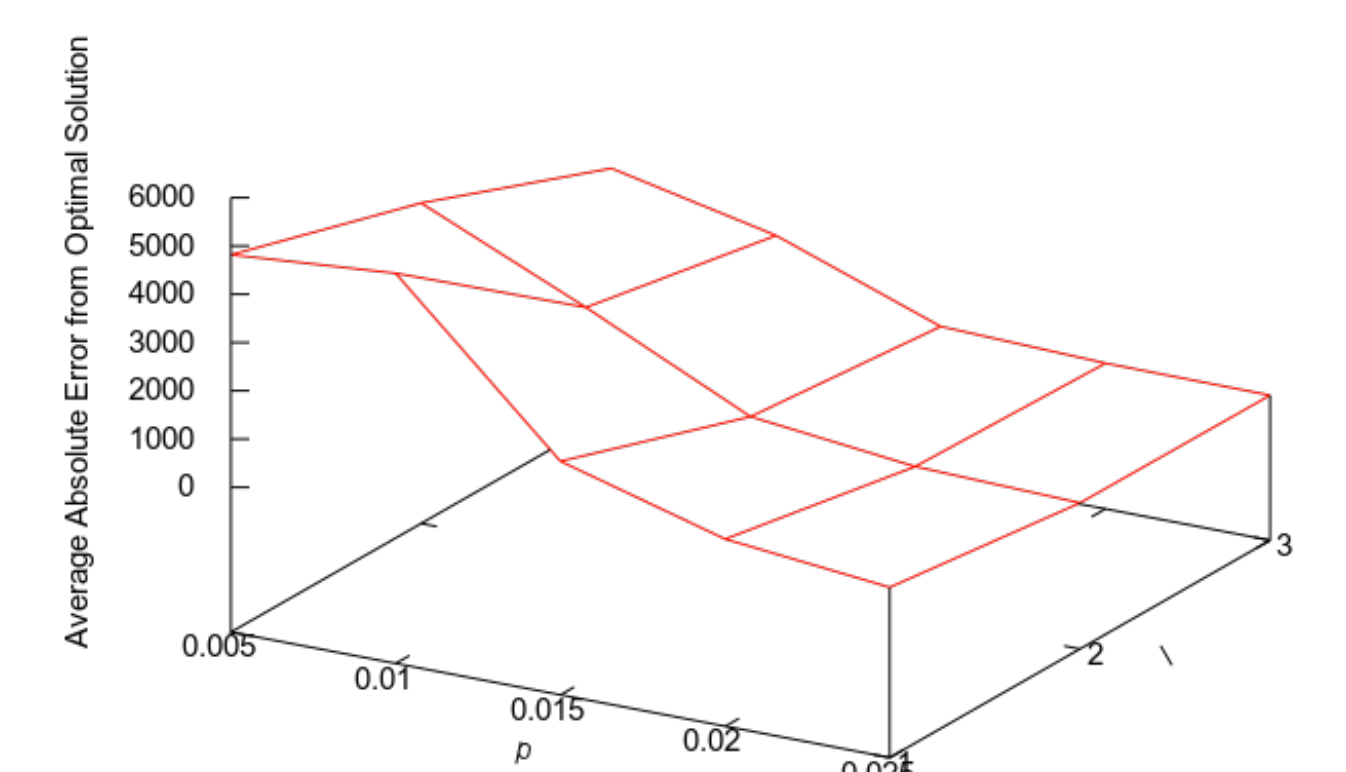
Figure: PUCS dynamics on an instance of the U-Curve problem.

Acknowledgements



Results

We implemented the PUCS algorithm on C++ language and used OpenMP to parallelize the code. Using a server with 64 cores and 256 gigabytes of memory we were able to confirm our expectations that the algorithm can find solutions as good as it possible (be optimal) as long as we increase the parameters p and l .



We used the *featsel* framework to benchmark PUCS with other algorithms, such as Exhaustive Search (ES), Sequential Forward Selection (SFS) and Backward Feature Selection (BFS).

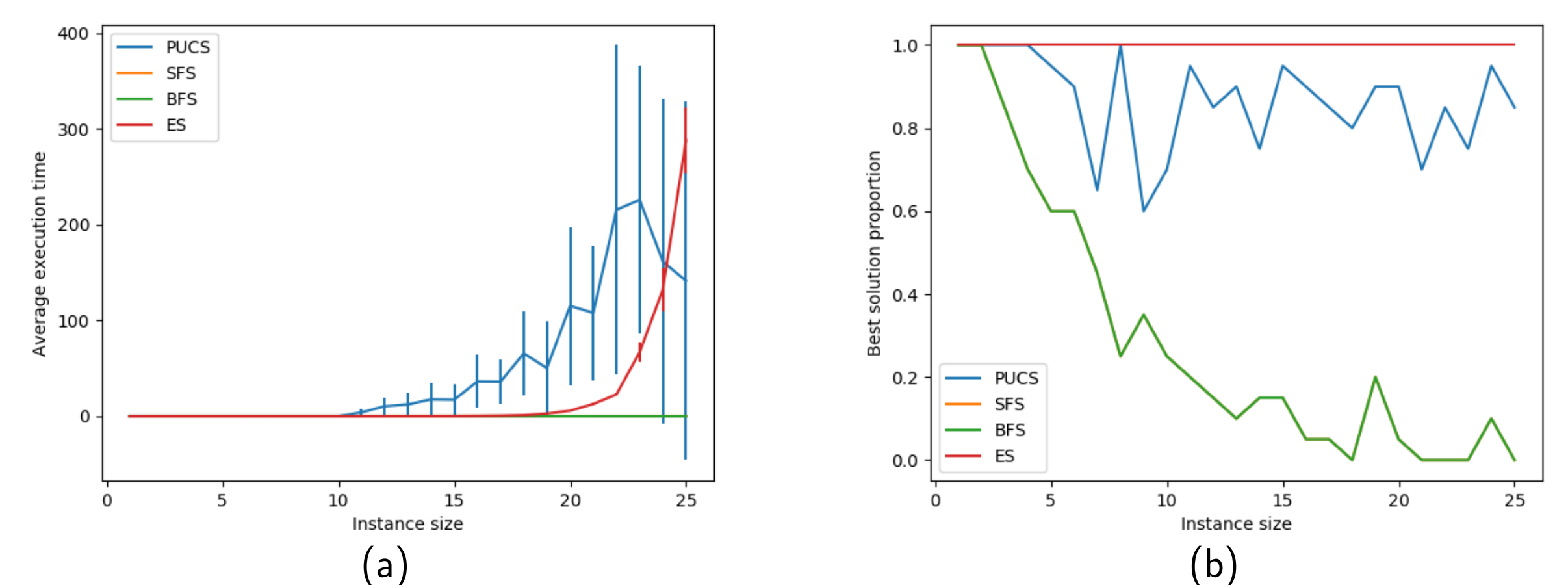


Figure: Average execution time and average number of times each algorithm found the best solution. For these instances we used an ES as a base for the PUCS algorithm.

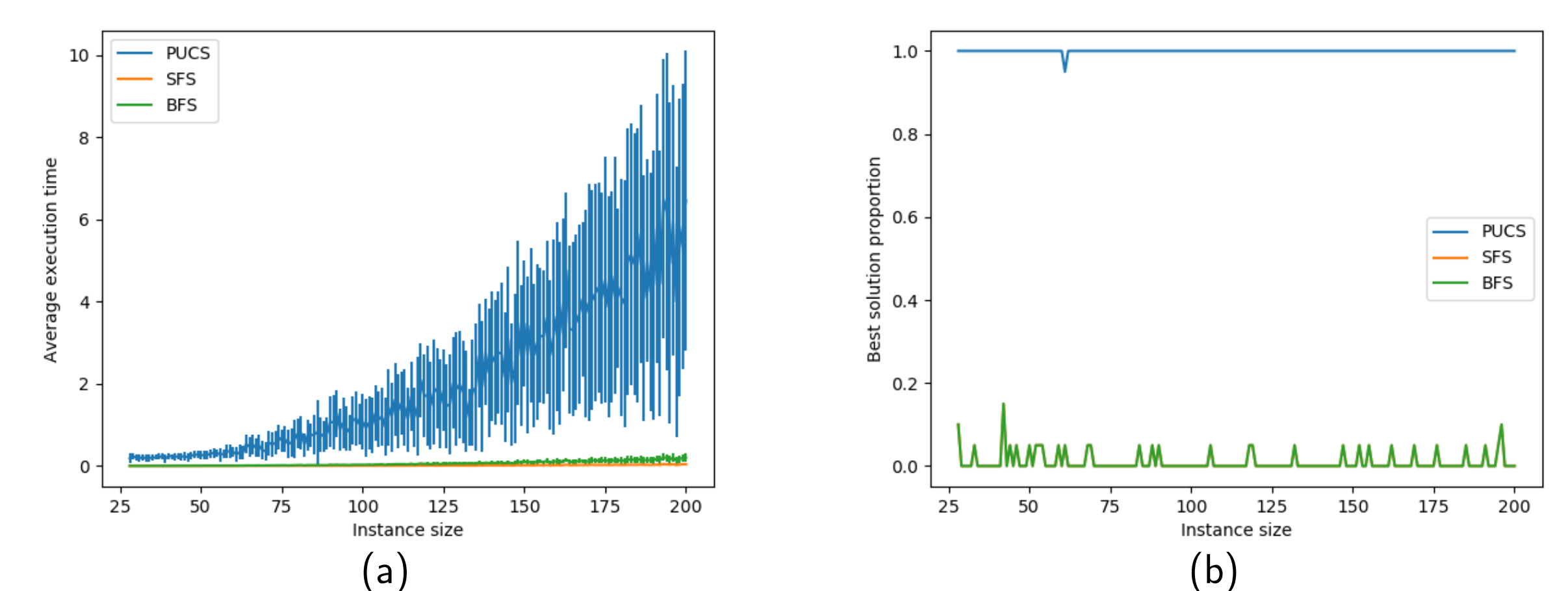


Figure: Average execution time and average number of times each algorithm found the best solution. For these instances we used an SFS as a base for the PUCS algorithm.

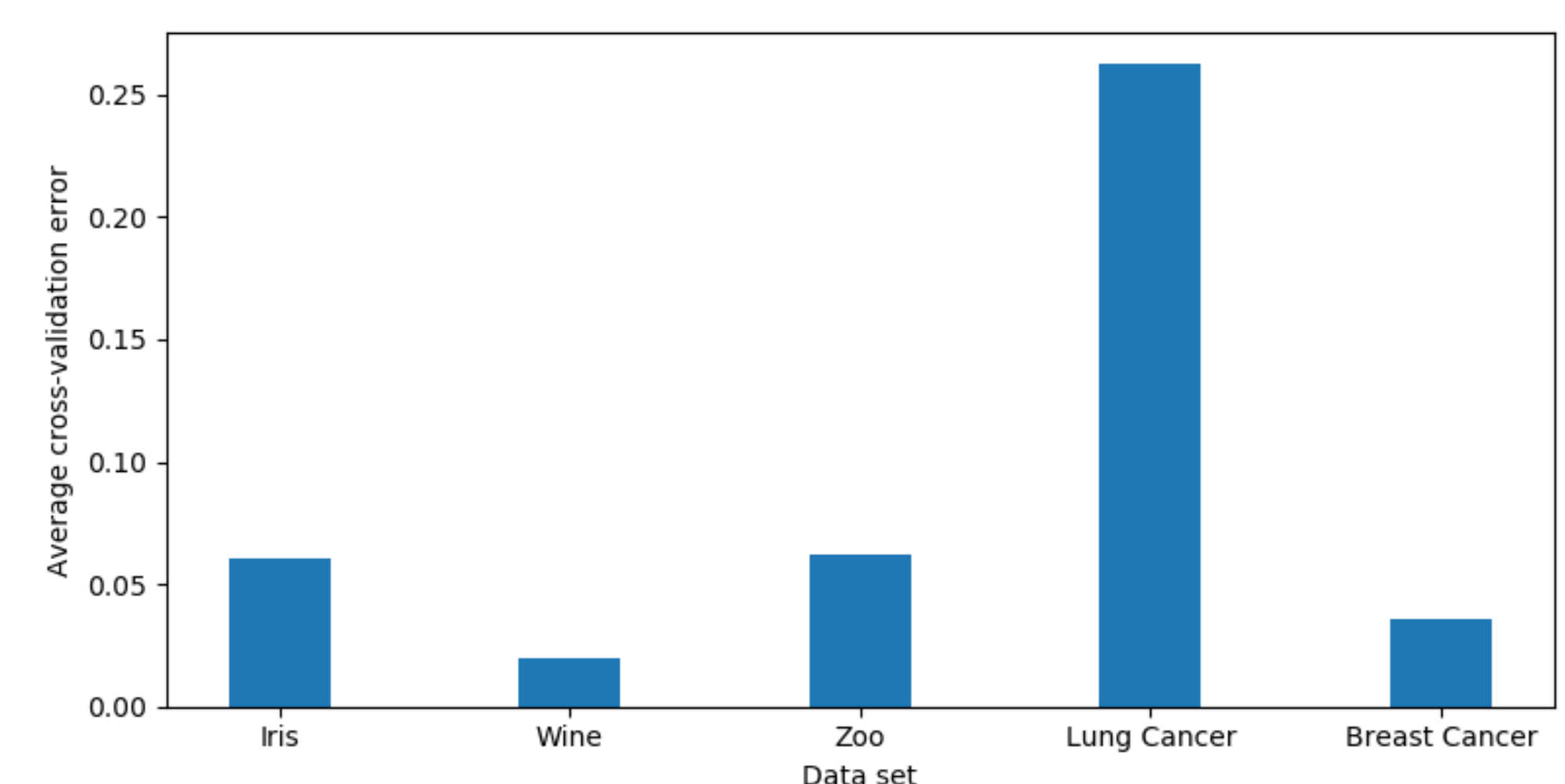


Figure: Caption

Conclusion