EcoSearch: A Constant-Delay Best-First Search Algorithm for Program Synthesis

Théo Matricon, Nathanaël Fijalkow, Guillaume Lagarde

cnrs LaBRI université de BORDEAUX

Motivations — Enumerative Program Synthesis

Specification → **Program Synthesis** → Program

Logical formulas

A program P such that $\forall x, \phi(x, P(x)) = \text{True}$

Natural language

"A program that removes odd elements and sort the rest"

Set of I/O examples

```
[1, 5, 4, 2]  ⟶  [2, 4]
[6, 3, 0, 8]  ⟶  [0, 6, 8]
```

Problem Formulation

Best-first search algorithms

Given a *heuristic cost function* w : Program $\to \mathbb{R}_{>0}$, as a WCFG

Find fast efficient algorithms to enumerate programs in the exact order of non-increasing weights

Some previous work

- 2017. *A**, Alur et al.
- 2018. *Euphony*, Lee et al.
- 2021. *Dreamcoder*, Ellis et al.
- 2022. *TF-Coder*, Shi et al.
- 2022. *Heap Search*, Fijalkow et al.
- 2023. *Bee Search*, Ameen and Lelis.

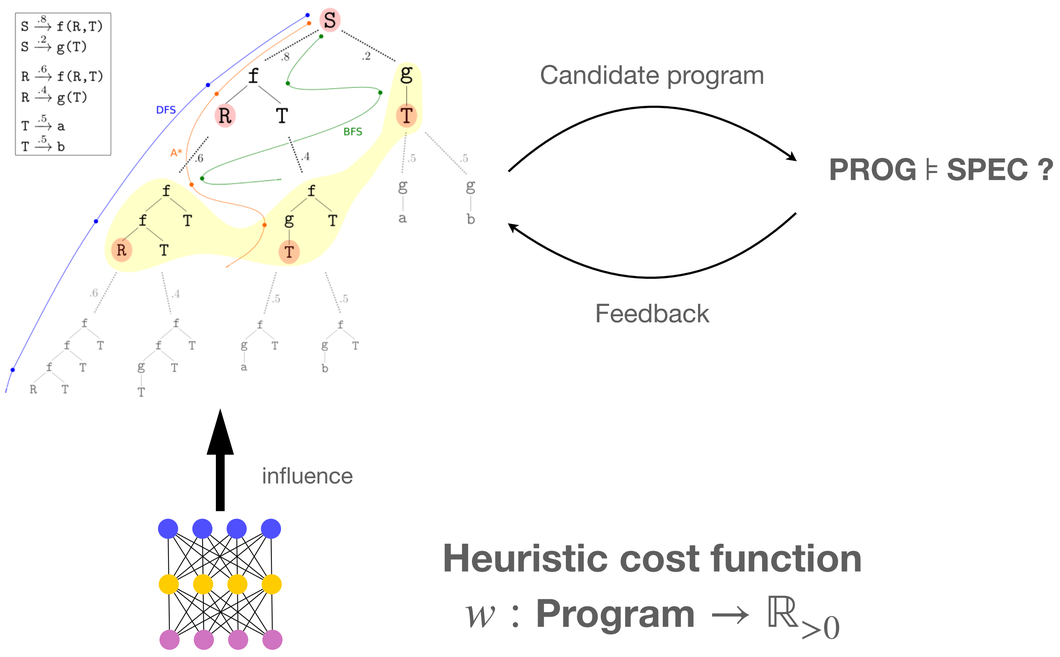
SOTA

- Bottom-up enumeration
- Delay $O(\log n)$

Is $O(\log n)$ optimal? Can we achieve O(1)?

Cost-Guided Program Synthesis

Combinatorial Search — **Evaluation**



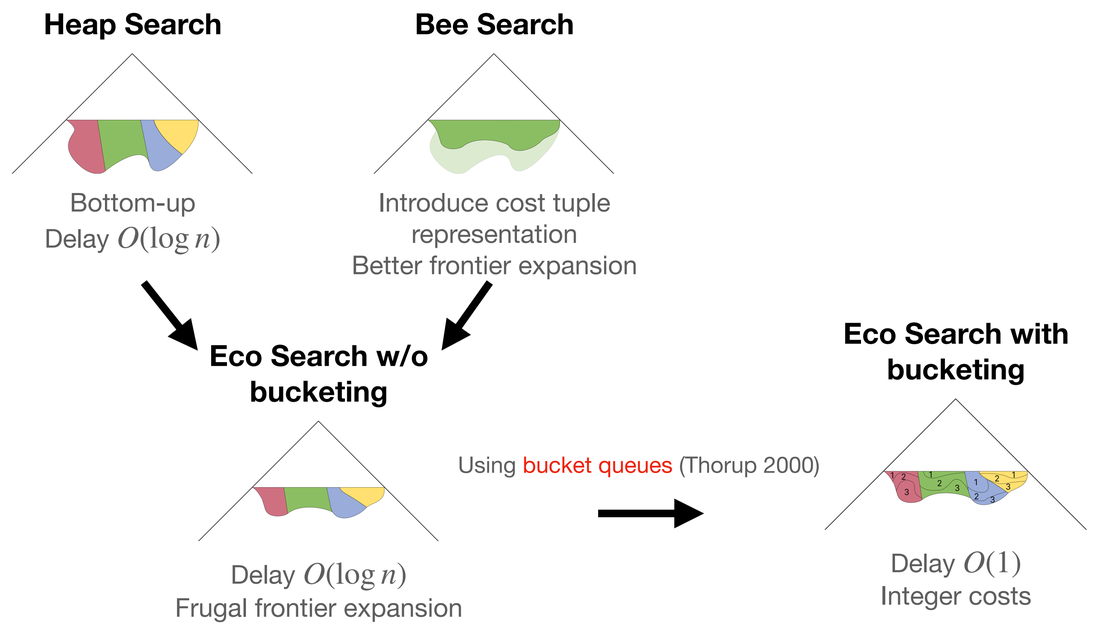
Heuristic cost function

w : **Program** $\to \mathbb{R}_{>0}$

EcoSearch — Our contribution

We provide a new best-first bottom-up search algorithm

- Theoretical guarantee → Constant delay, i.e., in time $O(1)$ between programs
- Performs well on experiments



Heuristics as Weighted Context-free Grammars

CFG

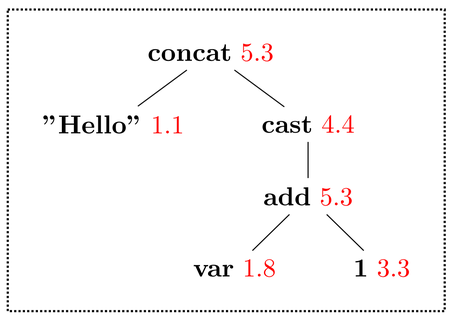
r_1 : str $\to$ "Hello"
r_2 : str $\to$ "World"
r_3 : str $\to$ cast(int)
r_4 : str $\to$ concat(str, str)
r_5 : int $\to$ var
r_6 : int $\to$ 1
r_7 : int $\to$ add(int, int)

WCFG

r_1 : str $\to$ "Hello"	cost : 1.1
r_2 : str $\to$ "World"	cost : 2.0
r_3 : str $\to$ cast(int)	cost : 4.4
r_4 : str $\to$ concat(str, str)	cost : 5.3
r_5 : int $\to$ var	cost : 1.8
r_6 : int $\to$ 1	cost : 3.3
r_7 : int $\to$ add(int, int)	cost : 5.3

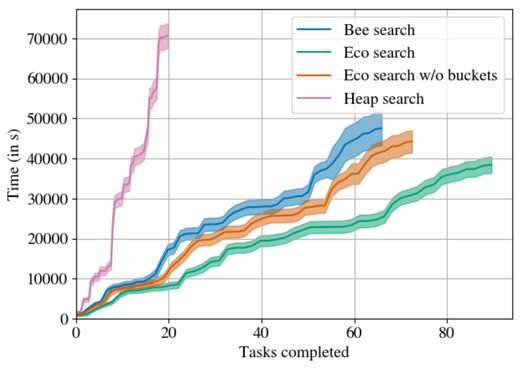
concat("Hello", cast(add(var,1)))

A WCFG *induces* a cost function w

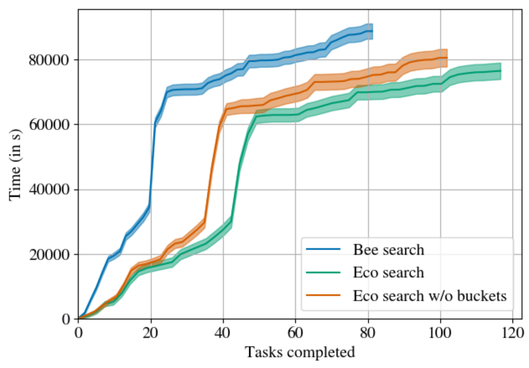


Cost =
$5.3 + 1.1 + 4.4 + 5.3 + 1.8 + 3.3 = 21.2$

Experiments



On the **FlashFill dataset**
- String manipulation
- 200 tasks from SyGuS
- Timeout of 300s



On the **DeepCoder dataset** (Balog et al.)
- Integer list manipulation
- 200 tasks
- Timeout of 300s