Supplemental Material for 'Using lineage age to augment search space exploration in lexicase selection'

Karen Suzue, Charles Ofria, and Alexander Lalejini

2024-08-01

Contents

Chapter 1

Introduction

This is not intended as a stand-alone document, but as a companion to our manuscript.

1.1 About our supplemental material

As you may have noticed (unless you're reading a pdf version of this), our supplemental material is hosted using GitHub pages. We compiled our data analyses and supplemental documentation into this nifty web-accessible book using bookdown.

The source code/configuration files for this supplemental material can be found in this GitHub repository.

Our supplemental material includes the following:

- GP instruction set (Section 2)
- Experiment data availability (Section 3)
- GP data analysis + statistics (Section 4)

1.2 Contributing authors

- Karen Suzue
- Charles Ofria
- Alexander Lalejini

Chapter 2

SignalGP instruction set

Below, we document the instruction set used in our GP system for our experiments.

Abbreviations:

- EOP: End of program
- Reg: local register
  - Reg[0] indicates the value at the register specified by an instruction's first *argument*, Reg[1] indicates the value at the register specified by an instruction's second argument, and Reg[2] indicates the value at the register specified by the instruction's third argument.
  - Reg[0], Reg[1], *etc*: Register 0, Register 1, *etc*.
- Input: input buffer
  - Follows same scheme as Reg
- Output: output buffer
  - Follows same scheme as Reg
- Global: global memory buffer
  - Follows same scheme as Reg
- Arg: Instruction argument
  - Arg[i] indicates the i'th instruction argument (an integer encoded in the genome)
  - E.g., Arg[0] is an instruction's first argument

Instructions that would produce undefined behavior (e.g., division by zero) are treated as no operations.

2.1 Default Instructions

I.e., instructions used across all diagnostic tasks.

Instruction	Arguments Used	Description
Nop	0	No operation
Not	1	Reg[0] = !Reg[0]
Inc	1	Reg[0] = Reg[0] + 1
Dec	1	Reg[0] = Reg[0] - 1
Add	3	Reg[0] = Reg[1] + Reg[2]
Sub	3	Reg[0] = Reg[1] - Reg[2]
Mult	3	Reg[0] = Reg[1] * Reg[2]
Div	3	Reg[0] = Reg[1] / Reg[2]
Mod	3	Reg[0] = Reg[1] % Reg[2]
Nand	2	Reg[0] = !(R1g[0] & Reg[2])
TestEqu	3	Reg[0] = Reg[1] == Reg[2]
TestNEqu	3	Reg[0] = Reg[1] != Reg[2]
TestLess	3	Reg[0] = Reg[1] < Reg[2]
TestLessEqu	3	Reg[0] = Reg[1] <= Reg[2]
TestGreater	3	Reg[0] = Reg[1] > Reg[2]
TestGreaterEqu	3	Reg[0] = Reg[1] >= Reg[2]
SetMem	2	Reg[0] = Arg[1]
Terminal	1	Reg[0] = double value encoded by instruction tag
CopyMem	2	Reg[0] = Reg[1]
SwapMem	2	Swap(Reg[0], Reg[1])
InputToWorking	2	Reg[0] = Input[1]
WorkingToOutput	2	Output[1] = Reg[0]
If	1	If Reg[0] != 0, proceed. Otherwise skip to the next Close or EOP.

Instruction	Arguments Used	Description
While	1	While Reg[0] != 0, loop. Otherwise skip to next Close or EOP.
Close	0	Indicate the end of a control block of code (e.g., loop, if).
Break	0	Break out of current control flow (e.g., loop).
Call	0	Call a function, using this instruction's tag to determine which function is called.
Routine	0	Same as call, but local memory is shared. Sort of like a jump that will jump back when the routine ends.
Return	0	Return from the current function call.
WorkingToGlobal	2	Global[1] = Reg[0]
GlobalToWorking	2	Reg[1] = Global[0]
FullGlobalToWorking	0	Copy entire global memory buffer into working memory buffer
FullWorkingToGlobal	0	Copy entire working memory buffer into global memory buffer

Note that Nand performs a bitwise operation.

2.2 Problem-specific instructions

Each problem has problem-specific instructions for producing output.

2.2.1 Bouncing Balls

- SubmitOutput

2.2.2 Dice Game

- SubmitOutput

2.2.3 GCD

- SubmitOutput

2.2.4 Grade

- SubmitA
- SubmitB
- SubmitC
- SubmitD
- SubmitF

2.2.5 Snow Day

- SubmitOutput

Chapter 3

Data availability

3.1 Source code

Source code for this work can be accessed on GitHub: https://github.com/amlalejini/age-based-lex.

3.2 Experimental results

Data generated from our experiments used in analyses are available online, archived in an OSF repository: https://osf.io/efb4j/.

Chapter 4

Program synthesis experiments

```
experiment_slug <- "2024-05-20-inj-int"

working_directory <- paste0(
  "experiments/",
  experiment_slug,
  "/analysis/"
)

if (exists("bookdown_wd_prefix")) {
  working_directory <- paste0(
    bookdown_wd_prefix,
    working_directory
  )
}
```

4.1 Dependencies

```
library(tidyverse)
```

```
## Warning: package 'ggplot2' was built under R version
    4.2.3

## Warning: package 'dplyr' was built under R version
    4.2.3
```

```
## Warning: package 'stringr' was built under R version
   4.2.3

## — Attaching core tidyverse packages
   ————————————————————————————————————————————————
     tidyverse 2.0.0 —
## v dplyr     1.1.4     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## — Conflicts
   ————————————————————————————————————————————————
     tidyverse_conflicts() —
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r—lib
   .org/>) to force all conflicts to become errors

library(cowplot)

## Warning: package 'cowplot' was built under R version
   4.2.3

##
## Attaching package: 'cowplot'
##
## The following object is masked from 'package:lubridate
   ':
##
##     stamp

library(RColorBrewer)
library(khroma)
library(rstatix)

##
## Attaching package: 'rstatix'
##
## The following object is masked from 'package:stats':
##
##     filter

library(knitr)
library(kableExtra)
```

```
##
## Attaching package: 'kableExtra'
##
## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```
source("https://gist.githubusercontent.com/benmarwick/2
    a1bb0133ff568cbe28d/raw/
    fb53bd97121f7f9ce947837ef1a4c65a73bffb3f/geom_flat_
    violin.R")
library(ggpattern)
```

```
print(version)
```

```
##                _
## platform       aarch64-apple-darwin20
## arch           aarch64
## os             darwin20
## system         aarch64, darwin20
## status
## major          4
## minor          2.1
## year           2022
## month          06
## day            23
## svn rev        82513
## language       R
## version.string R version 4.2.1 (2022-06-23)
## nickname       Funny-Looking Kid
```

4.2 Setup

```
# Configure our default graphing theme
theme_set(theme_cowplot())
# Create a directory to store plots
plot_directory <- paste0(working_directory, "plots/")
dir.create(plot_directory, showWarnings=FALSE)
```

4.2.1 Load summary data

```
summary_data_loc <- paste0(working_directory, "data/
    aggregate.csv")
summary_data <- read_csv(summary_data_loc)

## Rows: 2700 Columns: 61
## -- Column specification
    ------------------------------------------------------------------------

## Delimiter: ","
## chr (11): ANCESTOR_FILE_PATH, EVAL_MODE,
    ORG_INJECTION_MODE, POP_INIT_MODE, ...
## dbl (50): AGE_LEX_AGE_ORDER_LIMIT,
    EVAL_CPU_CYCLES_PER_TEST, MAX_ACTIVE_THRE...
##
## i Use `spec()` to retrieve the full column
    specification for this data.
## i Specify the column types or set `show_col_types =
    FALSE` to quiet this message.

summary_data <- summary_data %>%
  mutate(
    PROBLEM = as.factor(PROBLEM),
    SELECTION = as.factor(SELECTION),
    EVAL_MODE = as.factor(EVAL_MODE),
    NUM_COHORTS = as.factor(NUM_COHORTS),
    TEST_DOWNSAMPLE_RATE = as.factor(TEST_DOWNSAMPLE_RATE
        ),
    AGE_LEX_AGE_ORDER_LIMIT = as.factor(AGE_LEX_AGE_ORDER
        _LIMIT),
    RECOMB_PER_FUNC_SEQ_RECOMB_RATE = as.factor(RECOMB_
        PER_FUNC_SEQ_RECOMB_RATE),
    ORG_INJECTION_COUNT = as.factor(ORG_INJECTION_COUNT),
    ORG_INJECTION_MODE = factor(
      ORG_INJECTION_MODE,
      levels = c(
        "none",
        "random",
        "recombine-random",
        "recombine-complement"
      )
    ),
    inject_cond = str_c(SELECTION, ORG_INJECTION_MODE,
        sep = "_"),
    ORG_INJECTION_INTERVAL = as.factor(ORG_INJECTION_
        INTERVAL),
    .keep = "all"
  ) %>%
  mutate(
    inject_cond = factor(
      inject_cond,
      levels = c(
        "age-lexicase_random",
        "lexicase_random",
        "lexicase_none"
      ),
      labels = c(
        "age-lex_inj-rand",
        "lex_inj-rand",
        "lex_inj-none"
      )
    ),
    sol_descends_injected = elite_elite_age != update &
        found_solution == "1"
  )

solution_counts <- summary_data %>%
  group_by(
    PROBLEM,
    SELECTION,
    AGE_LEX_AGE_ORDER_LIMIT,
    ORG_INJECTION_COUNT,
    ORG_INJECTION_MODE,
    ORG_INJECTION_INTERVAL,
    inject_cond
  ) %>%
  summarize(
    solution_count = sum(found_solution == "1"),
    replicates = n(),
    no_solution_count = n() - sum(found_solution == "1"),
    elite_from_injected = sum(elite_elite_age != update),
    sol_from_injected = sum(elite_elite_age != update &
        found_solution == "1")
  )

## `summarise()` has grouped output by 'PROBLEM', '
    SELECTION',
## 'AGE_LEX_AGE_ORDER_LIMIT', 'ORG_INJECTION_COUNT', '
    ORG_INJECTION_MODE',
## 'ORG_INJECTION_INTERVAL'. You can override using the
     `.groups` argument.

# print(solution_counts, n=208)
```

```
solution_table <- kable(solution_counts) %>%
  kable_styling(latex_options = "striped", font_size =
     25)
save_kable(solution_table, paste0(plot_directory, "
   solution_counts_table.pdf"))
solution_table
```

4.2.2 Success rates across injection intervals

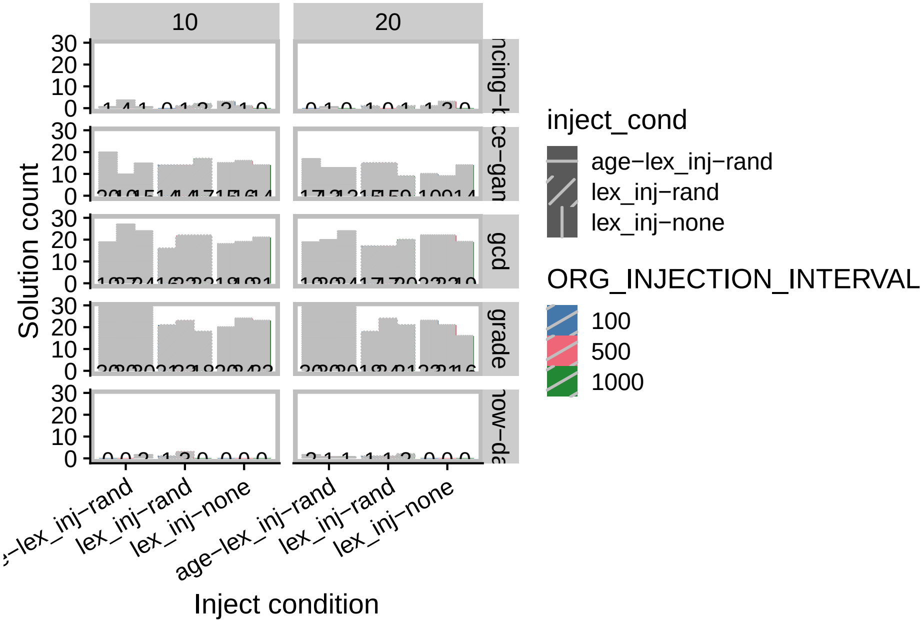
```
plot <- solution_counts %>%
  ggplot(
    aes(
      x = inject_cond,
      y = solution_count,
      fill = ORG_INJECTION_INTERVAL,
      pattern_angle = inject_cond
    )
  ) +
  geom_col_pattern(
    position = "dodge",
    pattern = "stripe",
    pattern_color = "gray",
    pattern_fill = "gray",
    pattern_spacing = 0.03,
    pattern_density = 0.05,
    # position = position_dodge(width = 0.4)
  ) +
  geom_text(
    aes(y = -0.7, label = solution_count),
    position = position_dodge(width=0.9)
  ) +
  scale_fill_bright() +
  scale_color_bright() +
  scale_x_discrete(
    name = "Inject␣condition"
  ) +
  scale_y_continuous(
    "Solution␣count"
  ) +
  facet_grid(
    PROBLEM ~ AGE_LEX_AGE_ORDER_LIMIT
  ) +
  theme(
      legend.position = "right",
```

PROBLEM	SELECTION	AGE_LEX_AGE_
bouncing-balls	age-lexicase	10
bouncing-balls	age-lexicase	10
bouncing-balls	age-lexicase	10
bouncing-balls	age-lexicase	20
bouncing-balls	age-lexicase	20
bouncing-balls	age-lexicase	20
bouncing-balls	lexicase	10
bouncing-balls	lexicase	10
bouncing-balls	lexicase	10
bouncing-balls	lexicase	10
bouncing-balls	lexicase	10
bouncing-balls	lexicase	10
bouncing-balls	lexicase	20
bouncing-balls	lexicase	20
bouncing-balls	lexicase	20
bouncing-balls	lexicase	20
bouncing-balls	lexicase	20
bouncing-balls	lexicase	20
dice-game	age-lexicase	10
dice-game	age-lexicase	10
dice-game	age-lexicase	10
dice-game	age-lexicase	20
dice-game	age-lexicase	20
dice-game	age-lexicase	20

```
    axis.text.x = element_text(
      angle = 30,
      hjust = 1
    ),
    panel.border = element_rect(color = "gray", size =
        2)
  )
```

```
## Warning: The `size` argument of `element_rect()` is
    deprecated as of ggplot2 3.4.0.
## i Please use the `linewidth` argument instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see
    where this warning was generated.
```

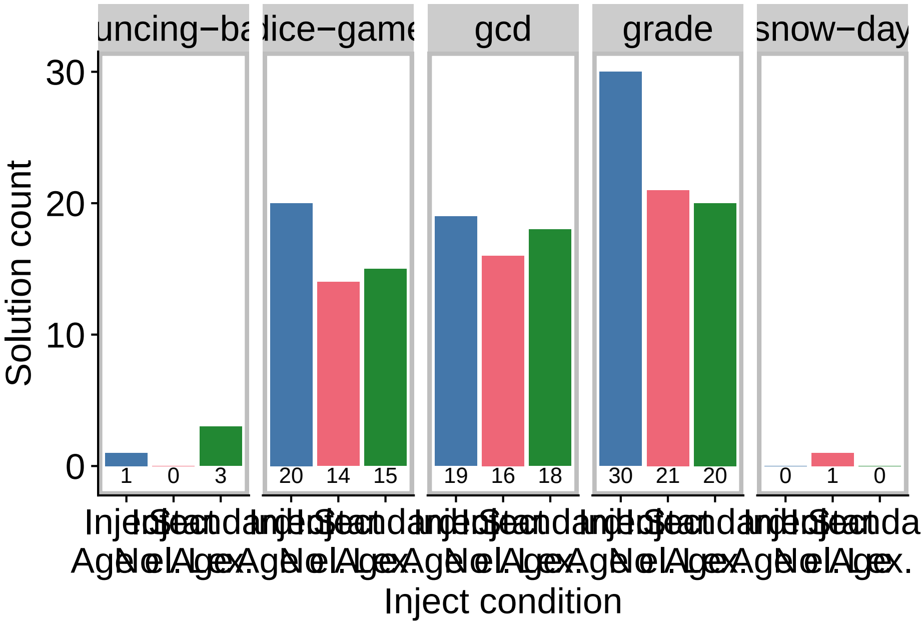
```
ggsave(
  filename = paste0(plot_directory, "solutions-bar.pdf"),
  plot = plot,
  width = 10,
  height = 15
)
plot
```



4.3 Interval 100 with age order limit 10

4.3.1 Problem-solving success

```
plot <- solution_counts %>%
  filter(ORG_INJECTION_INTERVAL == "100" & AGE_LEX_AGE_
     ORDER_LIMIT == "10") %>%
  ggplot(
    aes(
      x = inject_cond,
      y = solution_count,
      fill = inject_cond
    )
  ) +
  geom_col() +
  geom_text(
    aes(y = -0.7, label = solution_count),
    position = position_dodge(width=0.9)
  ) +
  scale_fill_bright() +
  scale_color_bright() +
  scale_x_discrete(
    name = "Inject␣condition",
    limits = c("age-lex_inj-rand", "lex_inj-rand", "lex_
        inj-none"),
    labels = c("Inject\nAge␣el.", "Inject\nNo␣Age", "
        Standard\nLex.")
  ) +
  scale_y_continuous(
    "Solution␣count"
  ) +
  facet_wrap(
    ~ PROBLEM,
    nrow = 1
  ) +
  theme(
      legend.position = "none",
      axis.text = element_text(
        size = 18
      ),
      axis.title = element_text(
        size = 18
      ),
      strip.text = element_text(
        size = 18
      ),
      # axis.text.x = element_text(
      #   angle = 30,
      #   hjust = 1
      # ),
      panel.border = element_rect(color = "gray", size =
         2)
    )
ggsave(
  filename = paste0(plot_directory, "solutions-bar-i100-
     ol10.pdf"),
  plot = plot,
  width = 18,
  height = 8
)
plot
```



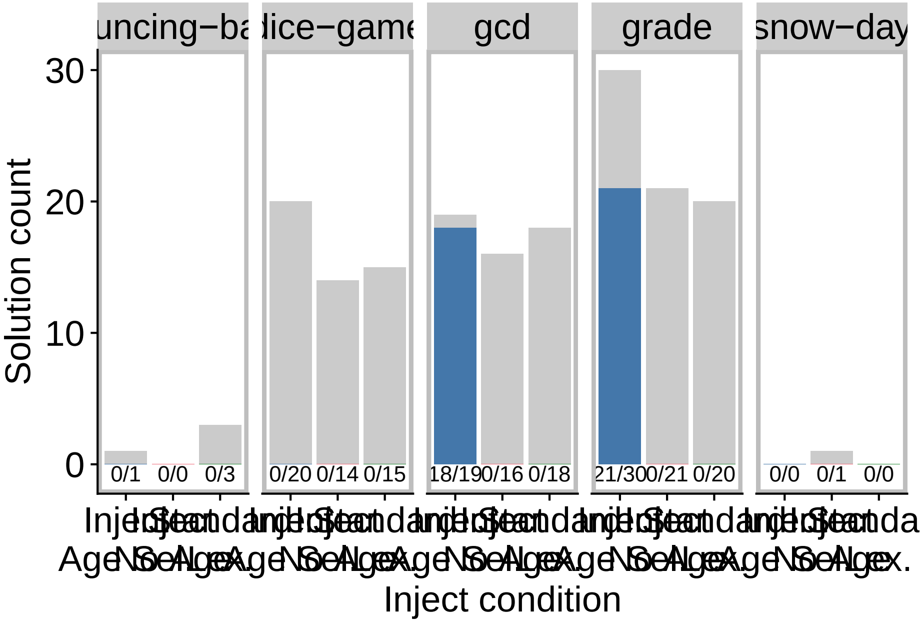
4.3.1.1 Statistics

```
sol_stats_data <- solution_counts %>%
  ungroup() %>%
  select(
    PROBLEM,
    ORG_INJECTION_INTERVAL,
    AGE_LEX_AGE_ORDER_LIMIT,
    inject_cond,
    solution_count,
    no_solution_count
  )

fisher_results <- data.frame(
  comparison = character(),
  group1 = character(),
  group2 = character(),
  n = integer(),
  p = double(),
  p.adj = double(),
  p.adj.signif = character()
)


# ORG_INJECTION_INTERVAL
# AGE_LEX_AGE_ORDER_LIMIT
inj_intervals <- levels(sol_stats_data$ORG_INJECTION_
   INTERVAL)
age_limits <- levels(sol_stats_data$AGE_LEX_AGE_ORDER_
   LIMIT)
problems <- levels(sol_stats_data$PROBLEM)

for (inj_interval in inj_intervals) {
  for (age_limit in age_limits) {
    for (problem in problems) {
      ft_results <- sol_stats_data %>%
        filter(
          PROBLEM == problem & AGE_LEX_AGE_ORDER_LIMIT ==
              age_limit & ORG_INJECTION_INTERVAL == inj_
             interval
        ) %>%
        select(inject_cond, solution_count, no_solution_
           count) %>%
        column_to_rownames(var = "inject_cond") %>%
        pairwise_fisher_test(
          p.adjust.method = "holm"
        ) %>%
        add_significance("p.adj")

      ft_results <- ft_results %>%
        mutate(
          problem = rep(problem, nrow(ft_results)),
          inj_interval = rep(inj_interval, nrow(ft_
              results)),
          age_limit = rep(age_limit, nrow(ft_results)),
          .keep = "all"
        ) %>%
        relocate(problem, inj_interval, age_limit)

      fisher_results <- rbind(
        fisher_results,
        ft_results
      )
    }
  }
}

fisher_results <- as.tibble(fisher_results)

## Warning: `as.tibble()` was deprecated in tibble 2.0.0.
## i Please use `as_tibble()` instead.
## i The signature and semantics have changed, see `?
   as_tibble`.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see
   where this warning was generated.

fisher_results <- fisher_results %>%
  mutate(
    # comparison = as.factor(comparison),
    problem = as.factor(problem),
    inj_interval = as.factor(inj_interval),
    age_limit = as.factor(age_limit),
    group1 = as.factor(group1),
    group2 = as.factor(group2),
  ) %>%
  group_by(
    problem
  )

fisher_table <- kbl(fisher_results) %>% kable_styling()
save_kable(fisher_table, paste0(plot_directory, "stats_
   table.pdf"))
fisher_table
```

4.3.2 Proportion of solutions that descend from injected programs

problem	inj_interval	age_limit	group1	group2	n	p	p.adj	p.adj.s
bouncing-balls	100	10	age-lex_inj-rand	lex_inj-none	60	6.12e-01	1.00e+00	ns
bouncing-balls	100	10	age-lex_inj-rand	lex_inj-rand	60	1.00e+00	1.00e+00	ns
bouncing-balls	100	10	lex_inj-none	lex_inj-rand	60	2.37e-01	7.11e-01	ns
dice-game	100	10	age-lex_inj-rand	lex_inj-none	60	2.95e-01	5.90e-01	ns
dice-game	100	10	age-lex_inj-rand	lex_inj-rand	60	1.92e-01	5.76e-01	ns
dice-game	100	10	lex_inj-none	lex_inj-rand	60	1.00e+00	1.00e+00	ns
gcd	100	10	age-lex_inj-rand	lex_inj-none	60	1.00e+00	1.00e+00	ns
gcd	100	10	age-lex_inj-rand	lex_inj-rand	60	6.01e-01	1.00e+00	ns
gcd	100	10	lex_inj-none	lex_inj-rand	60	7.95e-01	1.00e+00	ns
grade	100	10	age-lex_inj-rand	lex_inj-none	60	7.97e-04	2.39e-03	**
grade	100	10	age-lex_inj-rand	lex_inj-rand	60	1.94e-03	3.88e-03	**
grade	100	10	lex_inj-none	lex_inj-rand	60	1.00e+00	1.00e+00	ns
snow-day	100	10	age-lex_inj-rand	lex_inj-none	60	1.00e+00	1.00e+00	ns
snow-day	100	10	age-lex_inj-rand	lex_inj-rand	60	1.00e+00	1.00e+00	ns
snow-day	100	10	lex_inj-none	lex_inj-rand	60	1.00e+00	1.00e+00	ns
bouncing-balls	100	20	age-lex_inj-rand	lex_inj-none	60	1.00e+00	1.00e+00	ns
bouncing-balls	100	20	age-lex_inj-rand	lex_inj-rand	60	1.00e+00	1.00e+00	ns
bouncing-balls	100	20	lex_inj-none	lex_inj-rand	60	1.00e+00	1.00e+00	ns
dice-game	100	20	age-lex_inj-rand	lex_inj-none	60	1.19e-01	3.57e-01	ns
dice-game	100	20	age-lex_inj-rand	lex_inj-rand	60	7.96e-01	7.96e-01	ns
dice-game	100	20	lex_inj-none	lex_inj-rand	60	2.95e-01	5.90e-01	ns
gcd	100	20	age-lex_inj-rand	lex_inj-none	60	5.80e-01	1.00e+00	ns
gcd	100	20	age-lex_inj-rand	lex_inj-rand	60	7.92e-01	1.00e+00	ns
gcd	100	20	lex_inj-none	lex_inj-rand	60	2.79e-01	8.37e-01	ns
grade	100	20	age-lex_inj-rand	lex_inj-none	60	1.05e-02	2.10e-02	*
grade	100	20	age-lex_inj-rand	lex_inj-rand	60	1.24e-04	3.72e-04	***
grade	100	20	lex_inj-none	lex_inj-rand	60	2.67e-01	2.67e-01	ns
snow-day	100	20	age-lex_inj-rand	lex_inj-none	60	4.92e-01	1.00e+00	ns
snow-day	100	20	age-lex_inj-rand	lex_inj-rand	60	1.00e+00	1.00e+00	ns
snow-day	100	20	lex_inj-none	lex_inj-rand	60	1.00e+00	1.00e+00	ns
bouncing-balls	500	10	age-lex_inj-rand	lex_inj-none	60	3.53e-01	1.00e+00	ns
bouncing-balls	500	10	age-lex_inj-rand	lex_inj-rand	60	3.53e-01	1.00e+00	ns
bouncing-balls	500	10	lex_inj-none	lex_inj-rand	60	1.00e+00	1.00e+00	ns
dice-game	500	10	age-lex_inj-rand	lex_inj-none	60	1.92e-01	5.76e-01	ns
dice-game	500	10	age-lex_inj-rand	lex_inj-rand	60	4.30e-01	8.60e-01	ns
dice-game	500	10	lex_inj-none	lex_inj-rand	60	7.97e-01	8.60e-01	ns
gcd	500	10	age-lex_inj-rand	lex_inj-none	60	3.03e-02	9.09e-02	ns
gcd	500	10	age-lex_inj-rand	lex_inj-rand	60	1.81e-01	3.62e-01	ns
gcd	500	10	lex_inj-none	lex_inj-rand	60	5.80e-01	5.80e-01	ns
grade	500	10	age-lex_inj-rand	lex_inj-none	60	2.37e-02	4.74e-02	*
grade	500	10	age-lex_inj-rand	lex_inj-rand	60	1.05e-02	3.15e-02	*
grade	500	10	lex_inj-none	lex_inj-rand	60	1.00e+00	1.00e+00	ns
snow-day	500	10	age-lex_inj-rand	lex_inj-none	60	1.00e+00	1.00e+00	ns
snow-day	500	10	age-lex_inj-rand	lex_inj-rand	60	2.37e-01	7.11e-01	ns
snow-day	500	10	lex_inj-none	lex_inj-rand	60	2.37e-01	7.11e-01	ns
bouncing-balls	500	20	age-lex_inj-rand	lex_inj-none	60	6.12e-01	1.00e+00	ns
bouncing-balls	500	20	age-lex_inj-rand	lex_inj-rand	60	1.00e+00	1.00e+00	ns
bouncing-balls	500	20	lex_inj-none	lex_inj-rand	60	2.37e-01	7.11e-01	ns
dice-game	500	20	age-lex_inj-rand	lex_inj-none	60	4.22e-01	8.44e-01	ns
dice-game	500	20	age-lex_inj-rand	lex_inj-rand	60	7.96e-01	8.44e-01	ns
dice-game	500	20	lex_inj-none	lex_inj-rand	60	1.87e-01	5.61e-01	ns
gcd	500	20	age-lex_inj-rand	lex_inj-none	60	7.79e-01	1.00e+00	ns
gcd	500	20	age-lex_inj-rand	lex_inj-rand	60	5.96e-01	1.00e+00	ns

```
plot <- solution_counts %>%
  filter(ORG_INJECTION_INTERVAL == "100" & AGE_LEX_AGE_
     ORDER_LIMIT == "10") %>%
  ggplot(
    aes(
      x = inject_cond,
      # y = solution_count,
      y = sol_from_injected ,
      fill = inject_cond
    )
  ) +
  geom_col(aes(y = solution_count), fill = "grey", alpha
     = 0.8) +
  geom_col() +
  geom_text(
    aes(y = -0.7, label = paste0(sol_from_injected , "/",
       solution_count)),
    position = position_dodge(width=0.9)
  ) +
  scale_fill_bright() +
  scale_color_bright() +
  scale_x_discrete(
    name = "Inject␣condition",
    limits = c("age-lex_inj-rand", "lex_inj-rand", "lex_
       inj-none"),
    labels = c("Inject\nAge␣Sel.", "Inject\nNo␣Age", "
       Standard\nLex.")
  ) +
  scale_y_continuous(
    "Solution␣count"
  ) +
  facet_wrap(
    ~ PROBLEM,
    nrow = 1
  ) +
  theme(
      legend.position = "none",
      axis.text = element_text(
        size = 18
      ),
      axis.title = element_text(
        size = 18
      ),
      strip.text = element_text(
        size = 18
      ),
      panel.border = element_rect(color = "gray", size =
          2)
    )
ggsave(
  filename = paste0(plot_directory, "solutions-bar-i100-
      ol10-prop-descend.pdf"),
  plot = plot,
  width = 18,
  height = 8
)
plot
```



4.3.3 Lineage age distributions

Distributions of lineage age for solutions that descended from injected programs

```
plot <- summary_data %>%
  filter(
    inject_cond == "age-lex_inj-rand" & ORG_INJECTION_
        INTERVAL == "100" &
    AGE_LEX_AGE_ORDER_LIMIT == "10" & PROBLEM %in% c("gcd
        ", "grade") &
    sol_descends_injected
  ) %>%
  ggplot(
    aes(
      x = PROBLEM,
      # y = elite_elite_age,
      y = elite_elite_age / update,
      fill = PROBLEM
    )
  ) +
  # geom_flat_violin(
  #   position = position_nudge(x = .2, y = 0),
  #   alpha = .8,
  #   adjust = 1.5
  # ) +
  geom_point(
    mapping = aes(color = PROBLEM),
    position = position_jitter(width = .2),
    size = .5,
    alpha = 0.8
  ) +
  geom_boxplot(
    width = .2,
    outlier.shape = NA,
    alpha = 0.5
  ) +
  scale_y_continuous(
    name = "Elite␣age␣(as␣%␣of␣elapsed␣generations)"
  ) +
  scale_x_discrete(
    name = "Problem"
  ) +
  scale_fill_dark() +
  scale_color_dark() +
  # facet_wrap(
  #   ~PROBLEM,
  #   nrow = 1
  #   # ncol = 1
  # ) +
  # coord_flip() +
  theme(
    legend.position = "none",
    axis.text.x = element_text(
      angle = 30,
      hjust = 1
    ),
    panel.border = element_rect(color = "gray", size = 2)
  )
ggsave(
  filename = paste0(plot_directory, "elite-age-prop.pdf")
      ,
  plot = plot,
  width = 6,
  height = 4
)
```