Benchmark Programs

Below we show all the experimental results and benchmark programs from the paper. Programs in Section 2 correspond to these in Table 2 (programs with expected affine-sensitivity), while programs in Section 3 correspond to those in Table 3 (programs with expected linear-sensitivity). See the full version of the paper (with appendix) at https://arxiv.org/abs/1902.04744.

1 BENCHMARKS

Table 2. Experimental Results for Expected Affine-sensitivity (with $\epsilon = 1$, $L = 1$)

Example	Time/sec	$\eta(\mathbf{b})$	K	d	M
mini-Roulette	5.97	$13 \cdot x - 13$	-13	11	13
rdwalk	3.91	$-5 \cdot x + 5000$	-5	1	5
prdwalk variant	4.88	$-0.2857 \cdot x + 285.7$	-1.429	5	0.2857
prspeed	4.31	$-1.7143 \cdot x + 1714.3$	-5.143	3	1.7143
race variant	5.43	$-1.43 \cdot h + 1.43 \cdot t$	-4.29	4	2.86
ad. rdwalk 2D	4.55	$-0.77 \cdot x + 0.77 \cdot y$	-2.31	3	1.54
ad. rdwalk 1D Variant	4.49	$2.86 \cdot x$	-2.86	2	2.86
American Roulette	7.66	$20.27 \cdot x - 20.27$	-20.27	35	20.27

Table 3. Experimental Results for Expected Linear-sensitivity (with $\epsilon = 1$, $L = 1$)

Example	Time/sec	$\eta(\mathbf{b})$	K	d	M	c
mini-roulette variant	12.69	$2.45 \cdot x - 2.45$	-4.91	9	2.45	22.08
single-room heating	5.21	$-0.833 \cdot x + 16.67$	-1.25	2.1	0.833	1.75
double-room heating	5.64	$-2.27 \cdot x_1 + 45.45$	-3.41	2.87	2.27	6.518
rdwalk variant	4.44	$-2.5 \cdot x + 2500$	-7.5	3	2.5	7.5
prdwalk	4.46	$-0.5714 \cdot x + 571.4$	-2.86	5	0.5714	2.86
prspeed variant	5.00	$-0.5333 \cdot x + 533.3$	-2.67	5	0.5333	2.67
race	5.17	$-2 \cdot h + 2 \cdot t$	-6	4	4	6
simple while loop	3.59	$-2 \cdot x + 2000$	-2	1	2	2
pollutant disposal	4.87	$n - 5$	-3	3	1	3
ad. rdwalk 2D variant	6.07	$-0.606 \cdot x + 0.606 \cdot y$	-2.424	4	1.212	2.424
ad. rdwalk 1D	5.16	$1.11 \cdot x$	-2.22	2	1.11	2.22
American roulette variant	14.95	$2.08 \cdot x - 2.08$	-4.15	35	2.08	72.63

2 BENCHMARK PROGRAMS FOR EXPECTED AFFINE-SENSITIVITY

```
while x ≥ 1 do
    if prob(6/65) then
        x := x + 1; w := w + 2
    else if prob(4/59) then
        x := x + 2; w := w + 3
    else if prob(3/55) then
        x := x + 3; w := w + 4
    else if prob(2/52) then
        x := x + 5; w := w + 5
    else if prob(1/50) then
        x := x + 11; w := w + 6
    else x := x − 1
fi fi fi fi fi od
```

Fig. 1. A Mini-roulette example

```
while x ≤ 1000 do
    if prob(0.6) then
        x := x + 1
    else
        x := x − 1
    fi
od
```

Fig. 2. rdwalk

```
while x ≤ 1000 do
    if prob(0.5) then
        x := x + 2
    else
        x := x + 5
    fi
od
```

Fig. 3. A Variant of prdwalk

```
while x ≤ 1000 do
    if prob(0.75) then
        x := x + 0
    else
        if prob(2/3) then
            x := x + 2
        else
            x := x + 3
        fi
    fi
od
```

Fig. 4. prspeed

```
while h ≤ t do
    t := t + 1;
    if prob(0.5) then
        h := h + 1
    else
        if prob(0.6)
            h := h + 4
        else
            skip
        fi
    fi
od
```

Fig. 5. A Variant of race

```
while x ≤ y do
    if prob(0.5) then
        if prob(0.7) then
            x := x + 3
        else
            y := y + 2
        fi
    else
        if prob(0.7) then
            x := x + 2
        else
            y := y + 1
        fi
    fi
od
```

Fig. 6. Adversarial random walk in two dimensions

```
while x ≥ 0 do
        x := x + 1;
        if prob(0.5) then
            if prob(0.9) then
                x := x − 2
            else
                x := x + 1
            fi
        else
            x := x − 1
        fi
od
```

Fig. 7. A Variant of Adversarial random walk in one dimension

```
while x ≥ 1 do
    if prob(1/304) then
        x := x + 35; w := w + 35
    else if prob(2/303) then
        x := x + 17; w := w + 17
    else if prob(3/301) then
        x := x + 11; w := w + 11
    else if prob(2/149) then
        x := x + 8; w := w + 8
    else if prob(5/294) then
        x := x + 6; w := w + 6
    else if prob(6/289) then
        x := x + 5; w := w + 5
    else if prob(12/283) then
        x := x + 2; w := w + 2
    else if prob(2/271) then
        x := x − 0.5
    else if prob(18/269) then
        x := x + 1; w := w + 1
    else if prob(2/251) then
        x := x − 0.5
    else
        x := x − 1
    fi fi fi fi fi fi fi fi fi fi
od
```

Fig. 8. American Roulette

3 BENCHMARK PROGRAMS FOR EXPECTED LINEAR-SENSITIVITY

$r_1 \sim unif(1,2), r_2 \sim unif(2,3)$,
$r_3 \sim unif(3,4), r_4 \sim unif(4,5)$,
$r_5 \sim unif(8,9), r_6 \sim unif(1,2)$;
while $x \geq 1$ **do**
  if **prob**$(\frac{6}{65})$ **then**
    $x := x + r_1 ; w := w + 2$
  else if **prob**$(\frac{4}{59})$ **then**
    $x := x + r_2 ; w := w + 3$
  else if **prob**$(\frac{3}{55})$ **then**
    $x := x + r_3 ; w := w + 4$
  else if **prob**$(\frac{2}{52})$ **then**
    $x := x + r_4 ; w := w + 5$
  else if **prob**$(\frac{1}{50})$ **then**
    $x := x + r_5 ; w := w + 6$
  else $x := x - r_6$
fi fi fi fi fi od

Fig. 9. A variant of Mini-roulette example

$w \sim unif(-0.3, 0.3)$;
while $0 \leq x \leq 20$ **do**
  $x := x + 0.03 * (10 - x) + 1.5 + w$;
  $n := n + 1$
od

Fig. 10. Single-Room Heating

$w_1 \sim unif(-0.3, 0.3), w_2 \sim unif(-0.2, 0.2)$;
while $0 \leq x_1 \leq 20 \wedge 0 \leq x_2 \leq 20$ **do**
$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} := \begin{pmatrix} x_1 + 0.03 * (10 - x_1) + 0.04 * (x_2 - x_1) + 1.5 + w_1 \\ x_2 + 0.02 * (10 - x_2) + 0.04 * (x_1 - x_2) + w_2 \end{pmatrix}$;
$n := n + 1$
od

Fig. 11. Double-Room Heating

$r \sim unif(1,3)$;
while $x \leq 1000$ **do**
  if **prob**(0.6) **then**
    $x := x + r$
  else
    $x := x - r$
  fi
od

Fig. 12. A variant of rdwalk

```
r1 ~ unif(0,2), r2 ~ unif(0,5);
while x ≤ 1000 do
    if prob(0.5) then
        x := x + r1
    else
        x := x + r2
    fi
od
```

Fig. 13. prdwalk

```
r ~ unif(2,4);
while h ≤ t do
    t := t + 1;
    if prob(0.5) then
        h := h + r
    else
        skip
    fi
od
```

Fig. 15. race

```
r1 ~ unif(1,2), r2 ~ unif(2,3),
r3 ~ unif(3,5);
while x ≤ 1000 do
    if prob(0.75) then
        x := x + r1
    else
        if prob(2/3) then
            x := x + r2
        else
            x := x + r3
        fi
    fi
od
```

Fig. 14. A Variant of prspeed

```
r ~ unif(0,1);
while x ≤ 1000 do
  x := x + r
od
```

Fig. 16. A Simple Probabilistic While Loop

```
r ~ unif(0, 1);
while x ≤ 1000 do
  x := x + r
od
```

Fig. 17. A Simple Probabilistic While Loop

```
r1, r2 ~ unif(2, 4), r1', r2' ~ unif(1, 2);
while x ≤ y do
  if prob(0.5) then
    if prob(0.7) then
      x := x + r1
    else
      y := y + r1'
    fi
  else
    if prob(0.7) then
      x := x + r2
    else
      y := y + r2'
    fi
  fi
od
```

Fig. 18. A Variant of adversarial random walk in two dimensions

```
r ~ unif(−1, 1);
while x ≥ 0 do
        x := x + r;
        if prob(0.5) then
            if prob(0.9) then
                x := x − 1
            else
                x := x + 1
            fi
        else
            x := x − 1
        fi
od
```

Fig. 19. Adversarial random walk in one dimension

```
r1 ~ unif(30, 35), r2 ~ unif(12, 17),
r3 ~ unif(9, 11), r4 ~ unif(7, 8),
r5 ~ unif(5, 6), r6 ~ unif(3, 5)
r7 ~ unif(2, 3), r7′ ~ unif(0.5, 1),
r8 ~ unif(1, 2), r8′ ~ unif(0.5, 1),
r9 ~ unif(1, 2);
while x ≥ 1 do
    if prob(1/304) then
        x := x + r1; w := w + 35
    else if prob(2/303) then
        x := x + r2; w := w + 17
    else if prob(3/301) then
        x := x + r3; w := w + 11
    else if prob(2/149) then
        x := x + r4; w := w + 8
    else if prob(5/294) then
        x := x + r5; w := w + 6
    else if prob(6/289) then
        x := x + r6; w := w + 5
    else if prob(12/283) then
        x := x + r7; w := w + 2
    else if prob(2/271) then
        x := x − r7′
    else if prob(18/269) then
        x := x + r8; w := w + 1
    else if prob(2/251) then
        x := x − r8′
    else
        x := x − r9
    fi fi fi fi fi fi fi fi fi fi
od
```

Fig. 20. A Variant of American Roulette