ЛДЗ №1
Реализация SAT-решателя

1. За основу реализации был выбран алгоритм DPLL:

DPLL — рекурсивный алгоритм полного перебора с оптимизациями:

- Unit Propagation — автоматическое упрощение формулы
- Pure Literal Rule — устранение "чистых" литералов
- Decision Making — ветвление по выбору переменной и её полярности
- Backtracking — откат при конфликте

```
Algorithm DPLL
    Input: A set of clauses Φ.
    Output: A truth value indicating whether Φ is satisfiable.
function DPLL(Φ)
    // unit propagation:
    while there is a unit clause {l} in Φ do
        Φ ← unit-propagate(l, Φ);
    // pure literal elimination:
    while there is a literal l that occurs pure in Φ do
        Φ ← pure-literal-assign(l, Φ);
    // stopping conditions:
    if Φ is empty then
        return true;
    if Φ contains an empty clause then
        return false;
    // DPLL procedure:
    l ← choose-literal(Φ);
    return DPLL(Φ ∧ {l}) or DPLL(Φ ∧ {¬l});
```

В худшем случае сложность алгоритма: O(2^n) — полный перебор всех присваиваний
Практическая производительность: Значительно лучше за счёт оптимизаций.

2. К алгоритму была применена эвристика Variable State Independent Decaying Sum (VSIDS).

VSIDS — это эвристика, которая динамически отслеживает "активность" литералов на основе их участия в конфликтах поиска, присваивая каждой переменной два score'a (для положительной и отрицательной полярности), где активность растёт при bumping'e

конфликтующих литералов и затухает со временем через коэффициент γ < 1.

```
VSIDS(φ, σ):
    // Инициализация
    Δ ← Δ0
    ∀l ∈ L: a(l) ← ω · occ(l, φ)

    while ¬solved(φ, σ):
        // Затухание
        Δ ← Δ · β
        if Δ > Rth:
            ∀l ∈ L: a(l) ← a(l)/Rf
            Δ ← Δ/Rf

        // Выбор
        l* ← argmax_{l∈L, var(l)∉dom(σ)} a(l)
        v ← var(l*), p ← polarity(l*)

        // Ветвление
        σ' ← σ ∪ {v ↦ p}
        conflict ← DPLL(φ[v↦p], σ')

        if conflict:
            // Bumping конфликтующих литералов
            Lconflict ← analyzeConflict(conflict)
            ∀l ∈ Lconflict: a(l) ← a(l) + Δ
```

3. Была предпринята попытка оптимизации программы за счёт параллельного запуска различных конфигураций VSIDS и вывода самого быстрого из них (Portfolio solver). Однако запуск алгоритма на нескольких корутинах показал себя медленнее из-за вычислительных затрат на распараллеливание программы, но позволил одним запуском определять оптимальную стратегию для определённой КНФ.

Стратегии VSIDS:

```
var configs []slr.VSIDSConfig = []slr.VSIDSConfig{
    {
        Name:             "minisat-classic",
        InitialBumpInc:   1.0,   // Стандартный bump
        DecayFactor:      0.95,  // Классическое затухание
        RescaleThreshold: 1e100, // Стандартный порог
        RescaleFactor:    1e100,
    },
```

```
	{
		Name:              "rapid-decay",
		InitialBumpInc:    1.5,  // Более сильные bumps
		DecayFactor:       0.92, // Быстрее забывает старые
конфликты
		RescaleThreshold: 1e50, // Частая нормализация
		RescaleFactor:    1e50,
	},
	{
		Name:              "stable-longterm",
		InitialBumpInc:    0.5,   // Слабые bumps для стабильности
		DecayFactor:       0.98,  // Очень медленное затухание
		RescaleThreshold: 1e200, // Редкая нормализация
		RescaleFactor:    1e100,
	},
	{
		Name:              "init-heavy",
		InitialBumpInc:    0.1,   // Минимальные bumps
		DecayFactor:       0.99,  // Почти нет затухания
		RescaleThreshold: 1e300, // Практически без rescale
		RescaleFactor:    1e100,
	},
}
```

Пример определения оптимальной стратегии:

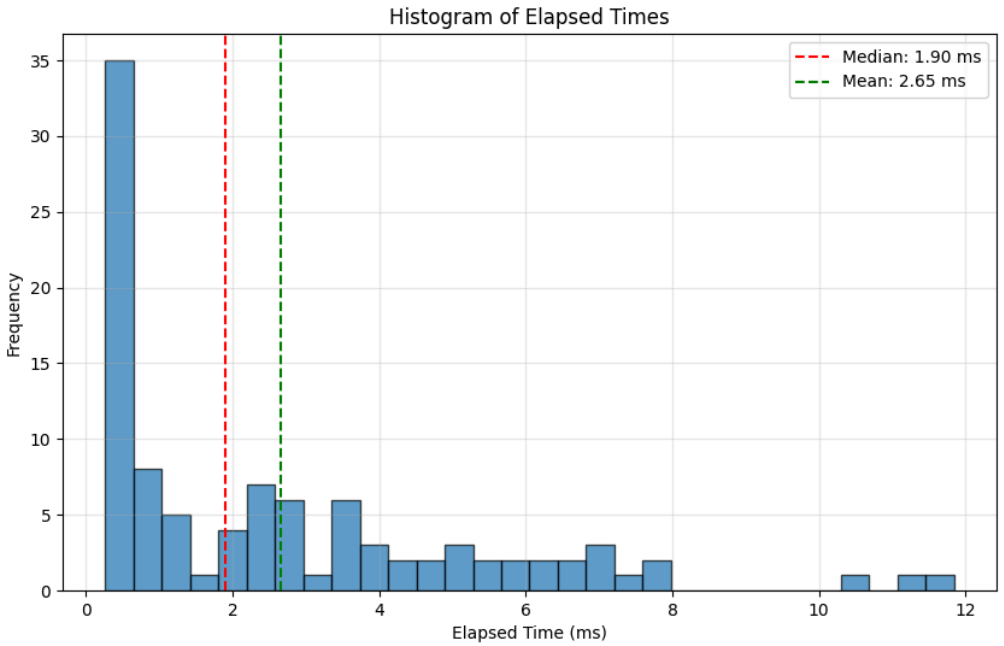
```
❯ go run satsolver -f cnfs/sat/big/CBS_k3_n100_m449_b90_4.cnf -c mc
Filename: cnfs/sat/big/CBS_k3_n100_m449_b90_4.cnf       Result: SAT
Elapsed time: 898.596718ms      Config: minisat-classic
❯ go run satsolver -f cnfs/sat/big/CBS_k3_n100_m449_b90_4.cnf -p
Filename: cnfs/sat/big/CBS_k3_n100_m449_b90_4.cnf       Result: SAT
Elapsed time: 1.649893738s      Config: stable-longterm
❯ go run satsolver -f cnfs/sat/big/CBS_k3_n100_m449_b90_4.cnf -c sl
Filename: cnfs/sat/big/CBS_k3_n100_m449_b90_4.cnf       Result: SAT
Elapsed time: 595.318732ms      Config: stable-longterm
```

4. Результаты бенчмарков:

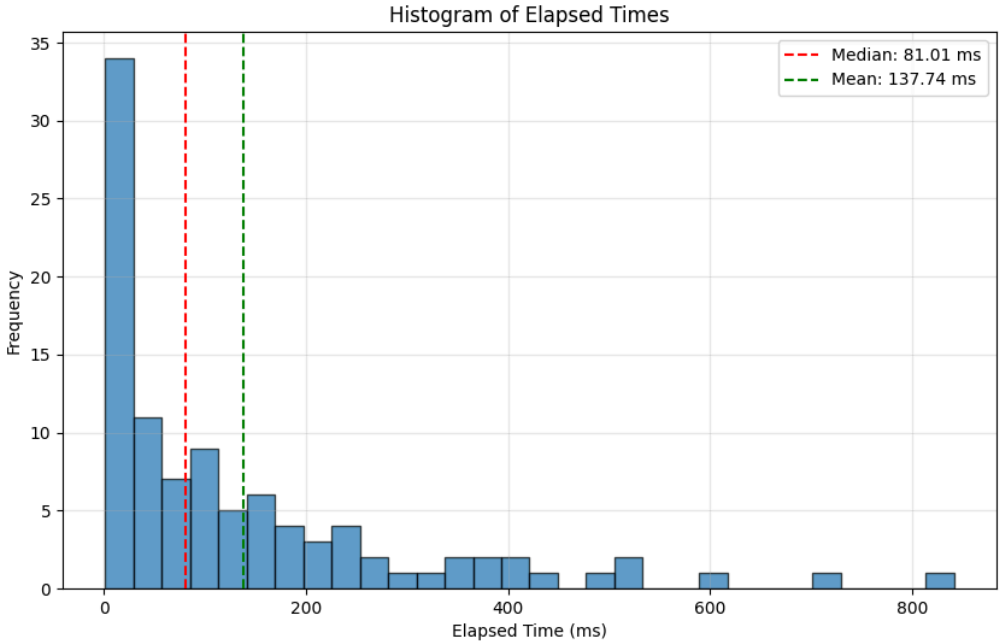
Во всех случаях полученные результаты были правильными.

Тестовые запуски при VSIDS конфигурации классического Minisat алгоритма дали следующие результаты:

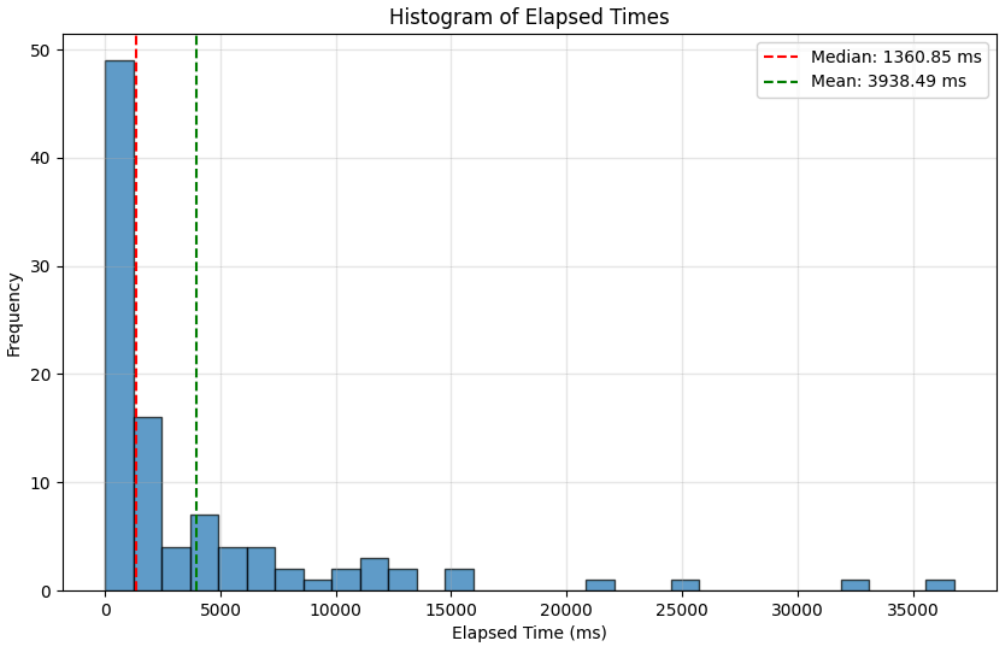
```
Dataset: 100 файлов из uf50-218
Median time: 1.8961 ms
Mean time: 2.6517 ms
Min time: 0.2588 ms
Max time: 11.8425 ms
```



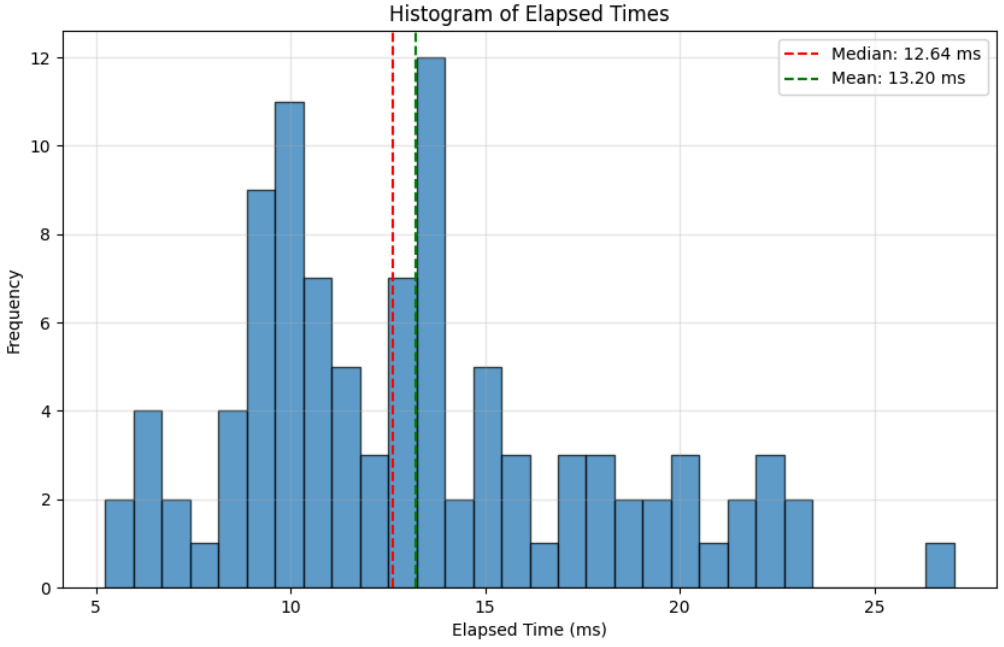

```
Dataset: 100 файлов из uf100-430
Median time: 81.0138 ms
Mean time: 137.7428 ms
Min time: 1.0327 ms
Max time: 841.7125 ms
```



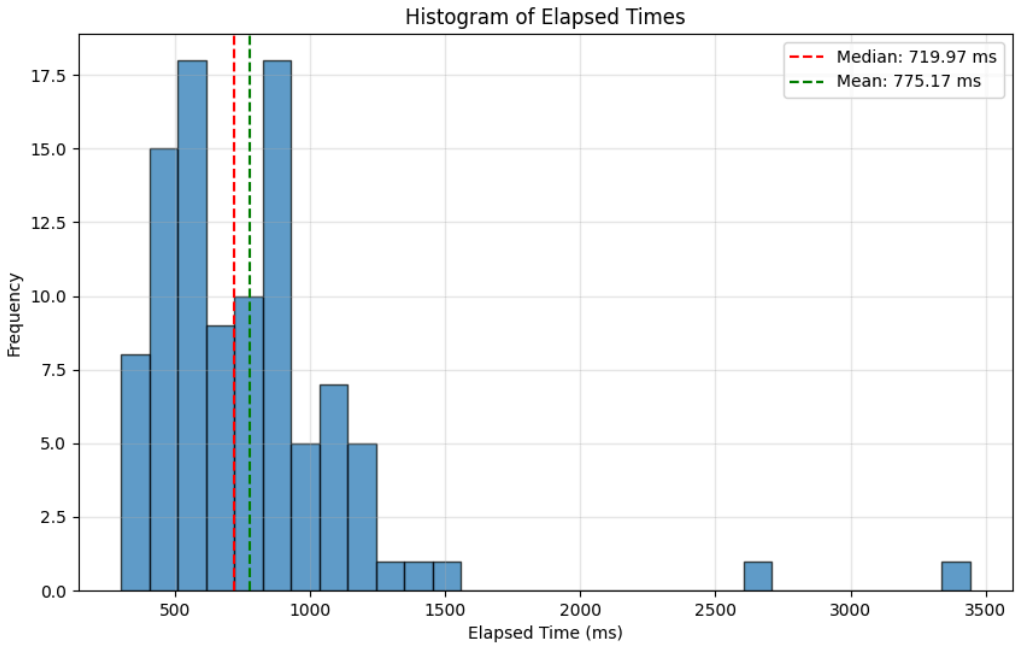
Dataset: 100 файлов из uf150-645
Median time: 1360.8518 ms
Mean time: 3938.4946 ms
Min time: 2.8397 ms
Max time: 36783.8523 ms



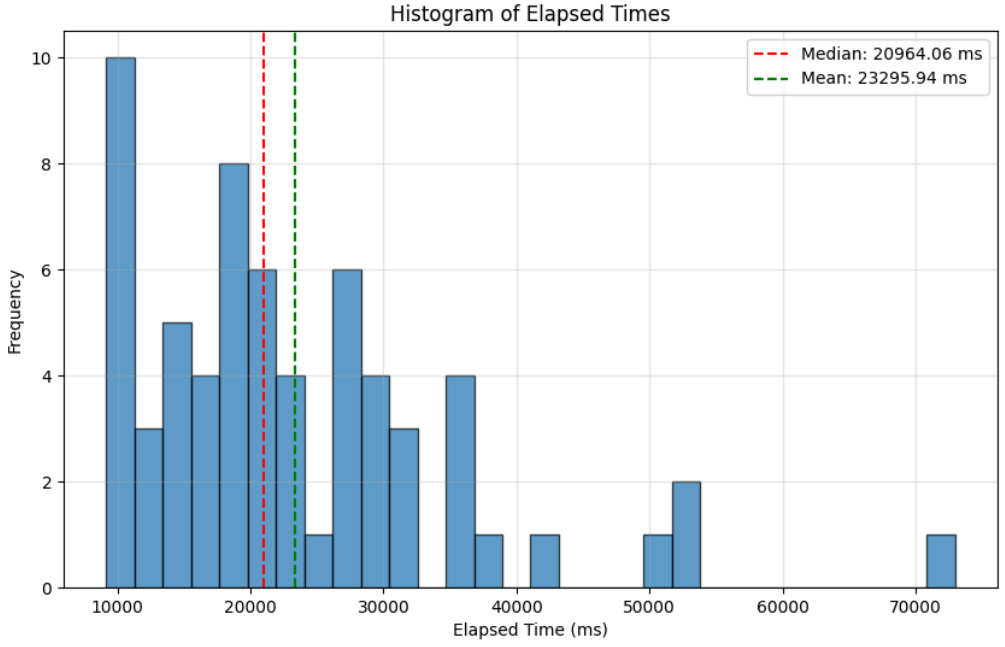
Dataset: 100 файлов из uuf50-218
Median time: 12.6369 ms
Mean time: 13.2006 ms
Min time: 5.2307 ms
Max time: 27.0507 ms



Dataset: 100 файлов из uuf100-430
Median time: 719.9710 ms
Mean time: 775.1748 ms
Min time: 300.7160 ms
Max time: 3442.5768 ms

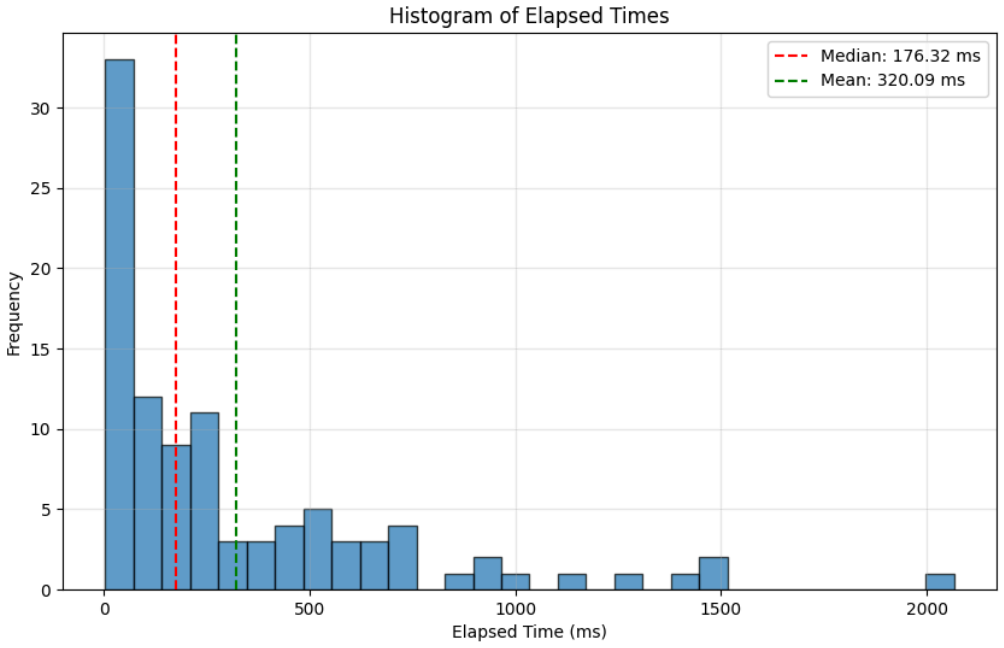


Dataset: 65 файлов из uuf150-645
Median time: 20964.0577 ms
Mean time: 23295.9401 ms
Min time: 9130.5735 ms
Max time: 72983.9223 ms



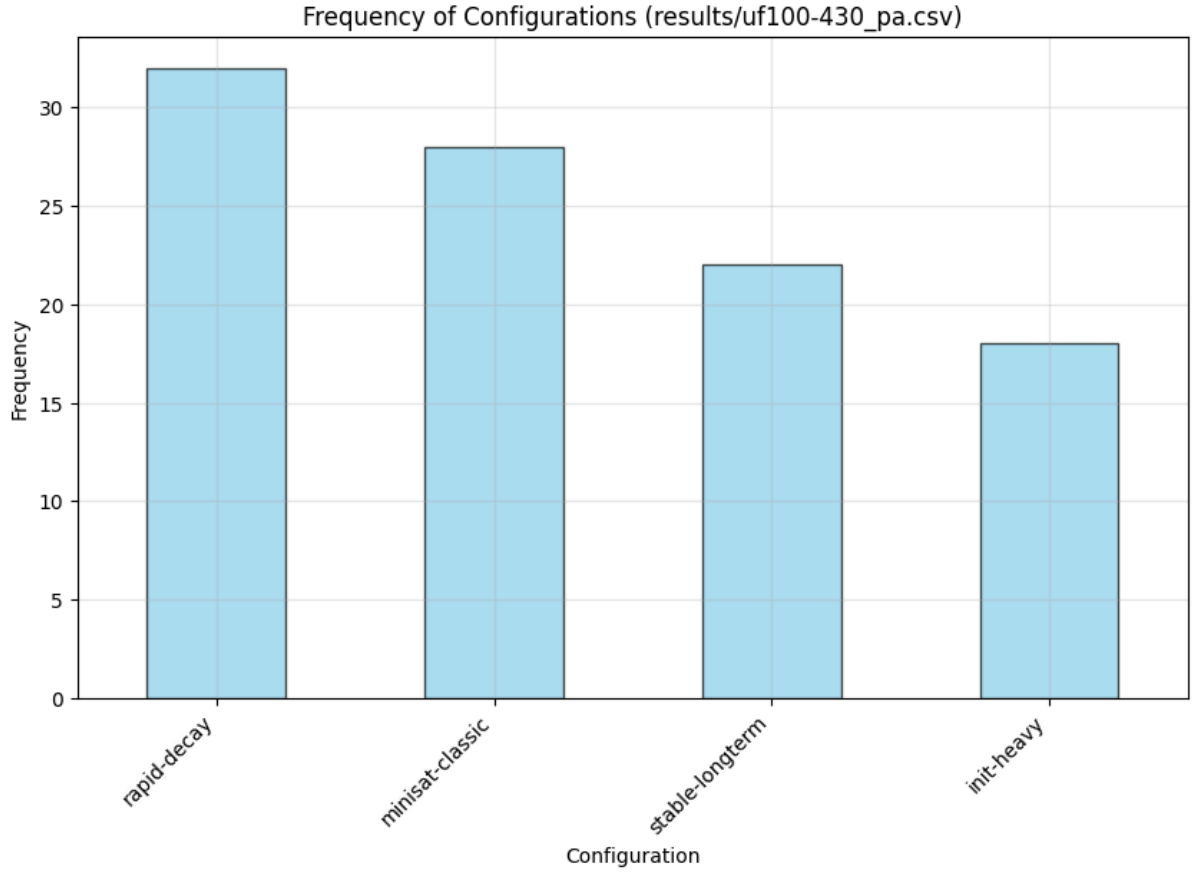
Запуск параллельной реализации на датасете uf100-430:

```
Dataset: uf100-430
Median time: 176.3166 ms
Mean time: 320.0944 ms
Min time: 2.8816 ms
Max time: 2066.0320 ms
```



Гистограммы однопоточного и многопоточного запуска имеют схожую форму распределения, однако значения времени выполнения во второй гистограмме в среднем примерно в два раза превышают соответствующие значения в первой гистограмме.

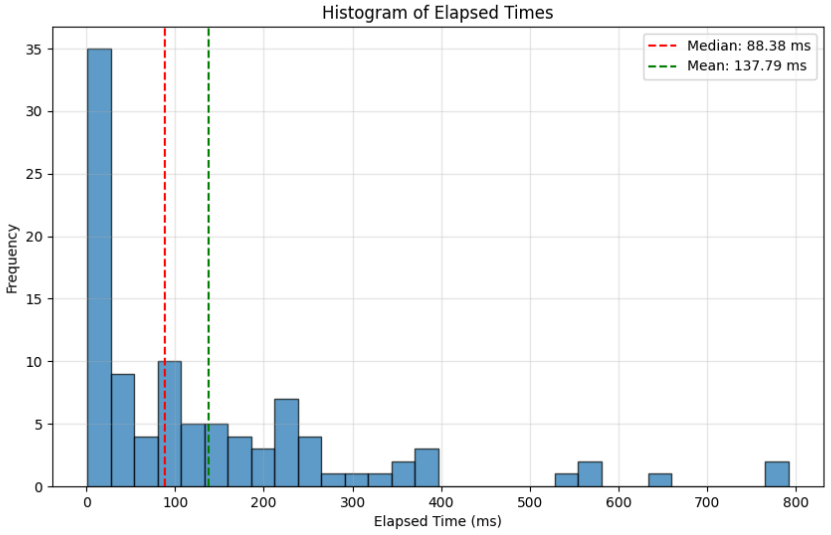
Гистограмма частот оптимальных стратегий:



Запустим датасет с использованием лучшей стратегии rapid-decay и худшей стратегии init-heavy, чтобы пронаблюдать разницу:

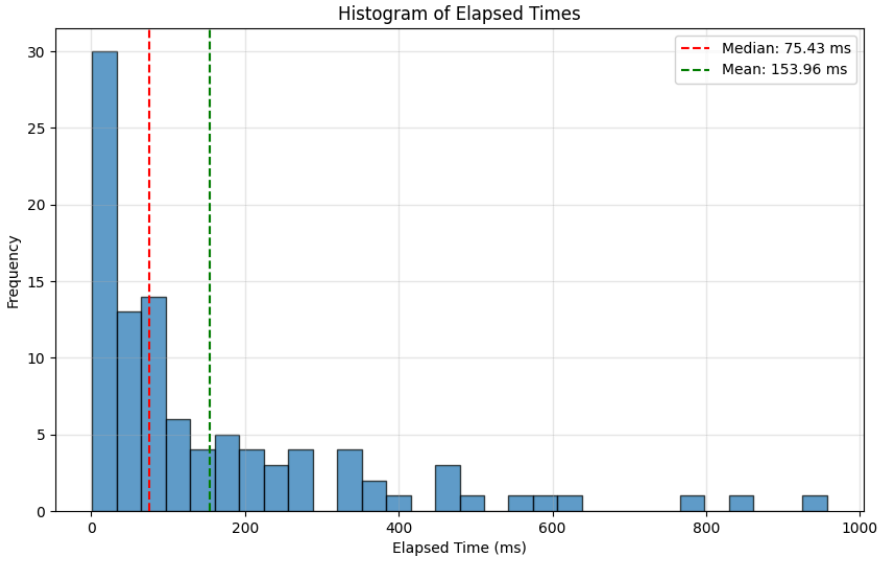
rapid-decay:

```
Dataset: uf100-430_rd.csv
Median time: 88.3754 ms
Mean time: 137.7913 ms
Min time: 1.0387 ms
Max time: 791.8797 ms
```



init-heavy:

```
Dataset: uf100-430_ih.csv
Median time: 75.4342 ms
Mean time: 153.9637 ms
Min time: 1.0177 ms
Max time: 957.1324 ms
```



Ссылка на исходный код: https://github.com/Rendex451/satsolver