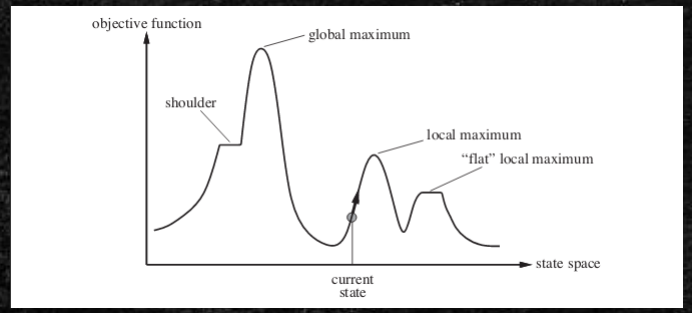
Evaluating the Performance of Hyper-Heuristics

Ruitao Feng
Supervisor: Pietro Oliveto

What are Hyper-Heuristics?



- High-level approaches to generate heuristics for a given problem
- Could be classified into two types:
  - Generation Hyper-heuristics
  - Selection Hyper-heuristics

Selection Hyper-Heuristic for MultiModal Optimisation

- Has been proven to escape local optima efficiently for toy benchmark like Cliff_d.
- Requirement: verify the performance for a natural combinatorial optimisation problem.

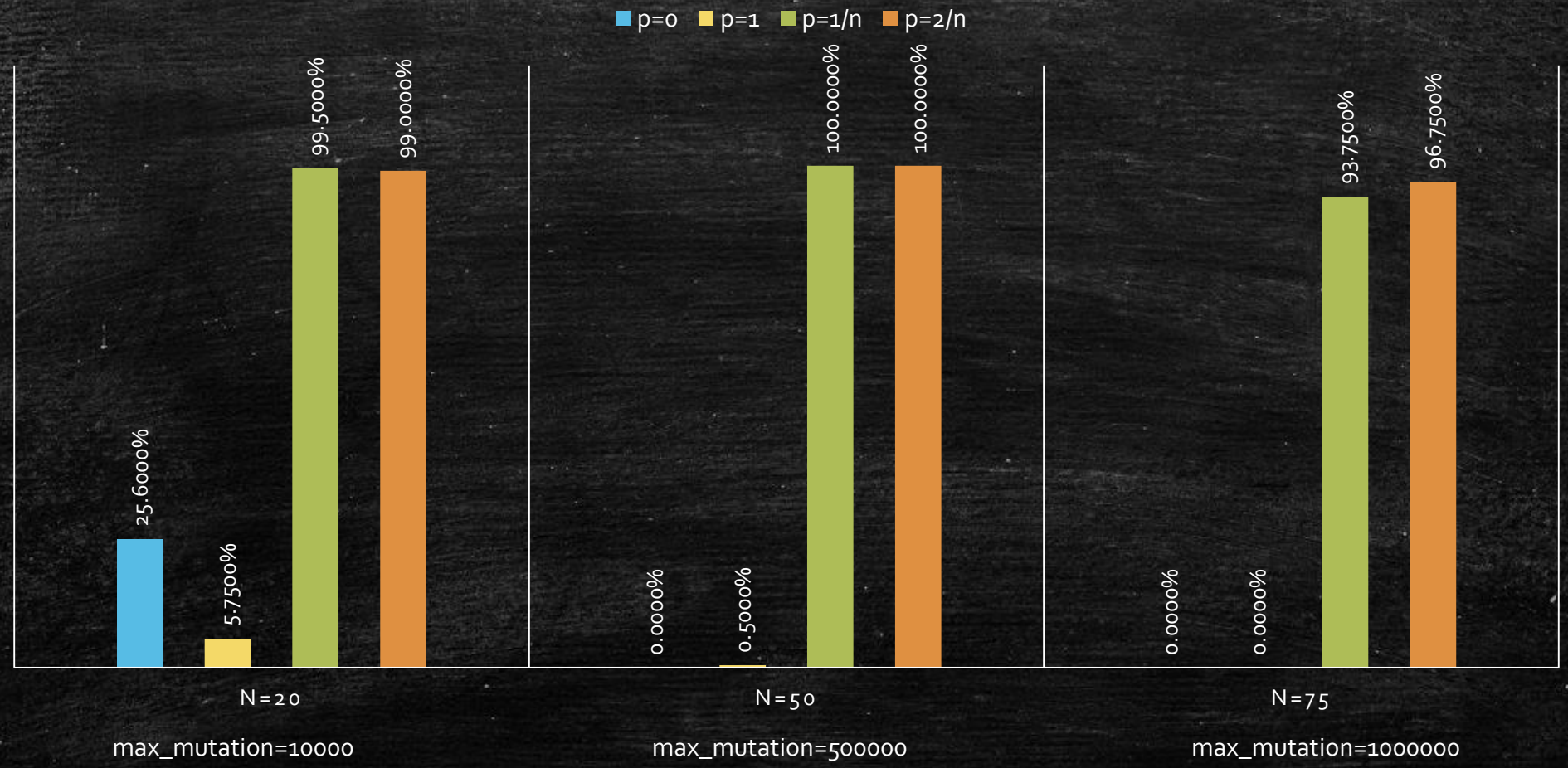
Algorithm 3 The Simple Acceptance Selection Hyper-heuristic [11]

Choose bitstring $x \in \{0,1\}^n$ uniformly at random;
$time := 0$;
while termination criteria not satisfied **do**
    $x' \leftarrow \textsc{FlipRandomBit}(x)$
    with probability p $x := x'$ // with probability p select the AM operator
    otherwise
    if $f(x') > f(x)$ **then**
        $x := x'$ // with probability $1 - p$ select the OI operator
    $time := time + 1$;
end while
return x;

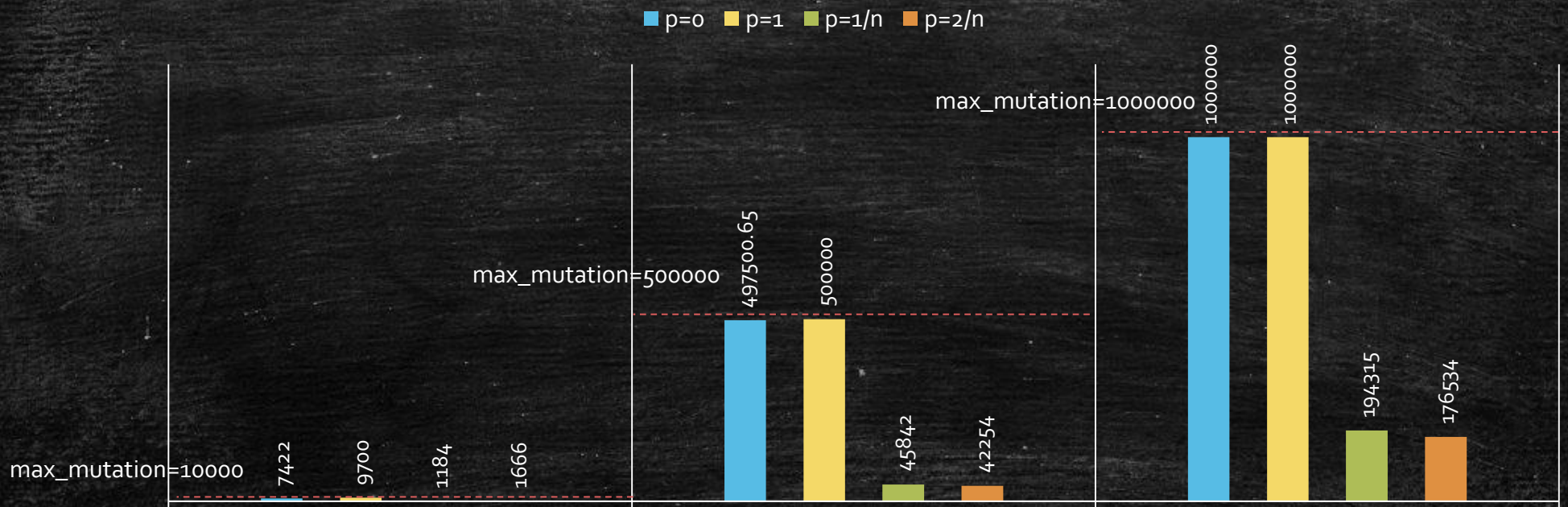
BENCHMARK PROBLEM: MAX-SAT

- Max-SAT is a generalisation of Boolean SATisfiability Problem (SAT)
  - Given a set of clauses, find the assignment for each variable that maximize the number of satisfied clauses.
  - SAT problem is the first to be proven as NP-complete, which means any NP problem could be reduced to SAT problem.
- Examples:
  - $(\neg x_1 \vee x_2) \wedge (\neg x_1 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_4) \wedge x_1 \wedge x_3$
- All problem instances tested in our project comes from SATLIB.

Success Rate of finding global optima within max_mutation step



Average runtime to find the best fitness



	n=20	n=50	n=75
p=0	7422	497500.65	1000000
p=1	9700	500000	1000000
p=1/n	1184	45842	194315
p=2/n	1666	42254	176534

Conclusion

- The Hyper-Heuristcs which switches between elitism and non-elitism performs much better than elitism or non-elitism operator alone for the NP-complete Max-SAT problem instances.
- Non-elitism contributes a lot in escaping local optima.

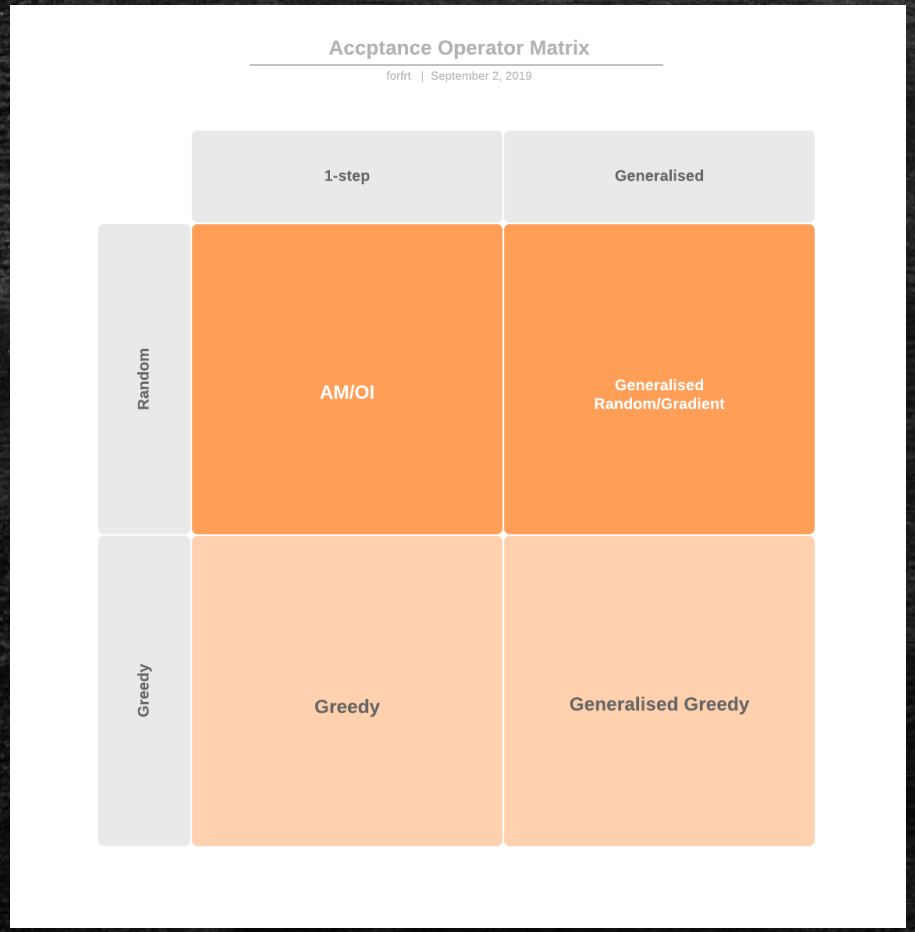
Extension: Selection Hyper-Heuristics Framework

- Hot-Plugging Modules:
  - Benchmark Function and Mutation Operators
  - Acceptance Operators

Algorithm 1 Simple Selection Heuristics Framework

procedure HYPER-HEURISTIC
  $x \sim H$ ▷ H denotes the search space of solutions
  while Termination criteria not satisfied **do**
    Acc $\sim D_p(AO_1, AO_2, \cdots, AO_n)$ ▷ Select Acceptance operators
    Var $\sim D_p(NO_1, NO_2, \cdots, NO_n)$ ▷ Select Mutation operators
    $x' \sim \mathbf{Var}(x)$
    if $\mathbf{ACC}(x, x')$ **then** $x \leftarrow x'$
  end while
  return b

Extension: More Acceptance Operators



1. AllMove and OnlyImprovement
2. Greedy
3. Generalised Random/Gradient
4. Generalised Greedy

Thank you for your time