LEVERHULME TRUST

University of St Andrews

A portfolio-based analysis method for competition results

Nguyen Dang

University of St Andrews, UK

nttd@st-andrews.ac.uk

LEVERHULME TRUST

University of St Andrews

A portfolio-based analysis method for competition results

/ˈpɛŋgwɪn/

Nguyen

University of St Andrews, UK

nttd@st-andrews.ac.uk

- Competitions are useful resources for comparing performance of different solving approaches
  - MiniZinc Challenges, SAT competition series, Internal Planning competitions,...

- Competitions are useful resources for comparing performance of different solving approaches
  - MiniZinc Challenges, SAT competition series, Internal Planning competitions,...
- Typical competition setting:
  - A set of benchmark instances (from different problems)
  - Competition ranking: based on *average performance across all instances*

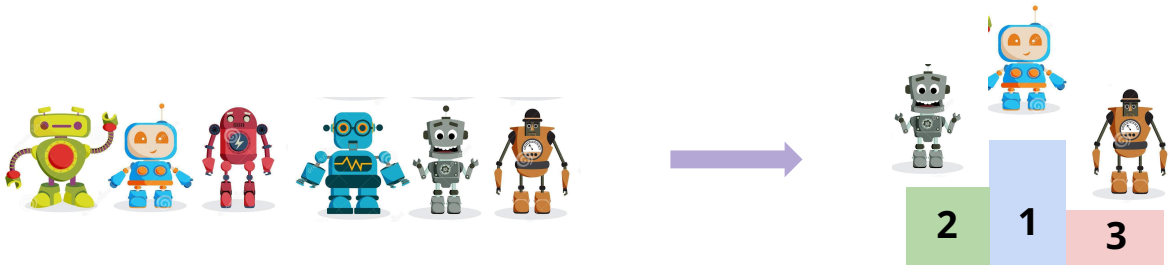


Image source: https://www.dreamstime.com/

- Competitions are useful resources for comparing performance of different solving approaches
  - MiniZinc Challenges, SAT competition series, Internal Planning competitions,...
- Typical competition setting:
  - A set of benchmark instances (from different problems)
  - Competition ranking: based on *average performance across all instances*

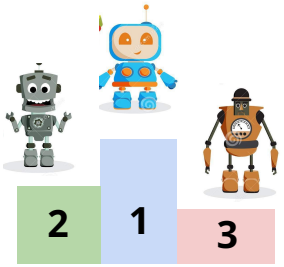
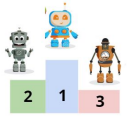


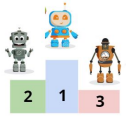
Image source: https://www.dreamstime.com/

Typical competition setting *with an additional portfolio-based analysis*



- **Competition ranking**: based on *average performance across all instances*
- **Portfolio-based analysis**: *provide additional insights into complementary strengths among solvers*

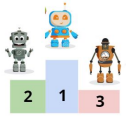
Typical competition setting *with an additional portfolio-based analysis*



- **Competition ranking**: based on *average performance across all instances*
- **Portfolio-based analysis**: *provide additional insights into complementary strengths among solvers*
  - Step 1: finding the smallest portfolio that can achieve the best possible performance

(infinite parallel resources)

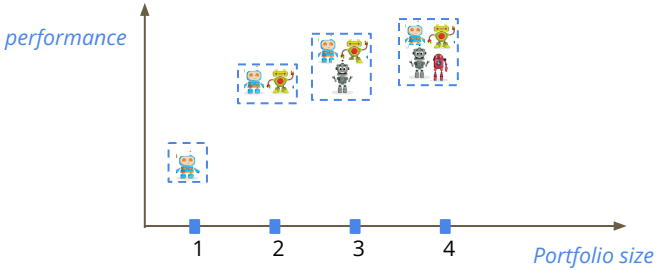
Typical competition setting *with an additional portfolio-based analysis*



- **Competition ranking**: based on *average performance across all instances*
- **Portfolio-based analysis**: *provide additional insights into complementary strengths among solvers*
  - Step 1: finding the smallest portfolio that can achieve the best possible performance

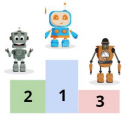
(infinite parallel resources)

  - Step 2: trade-off between portfolio sizes and performance



(limited parallel resources)

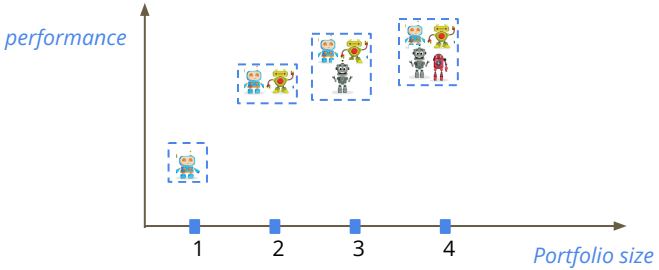
Typical competition setting *with an additional portfolio-based analysis*



- **Competition ranking**: based on *average performance across all instances*
- **Portfolio-based analysis**: *provide additional insights into complementary strengths among solvers*
  - Step 1: finding the smallest portfolio that can achieve the best possible performance

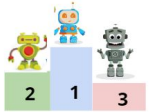
(infinite parallel resources)

  - Step 2: trade-off between portfolio sizes and performance

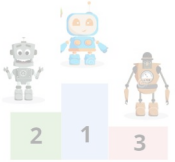


(limited parallel resources)

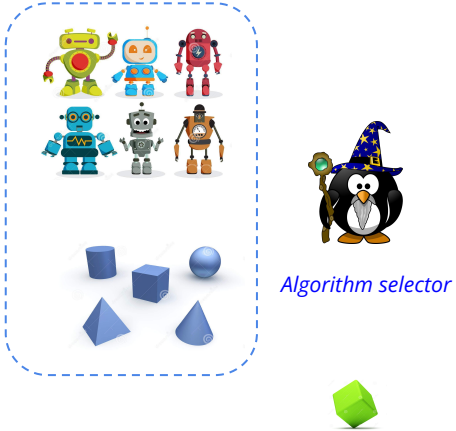
  - Step 3: solver importance from a portfolio viewpoint using Shapley values *(Fréchette et al 2016)*



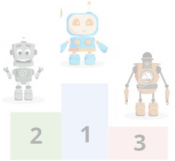
“Cooperative” competition setting



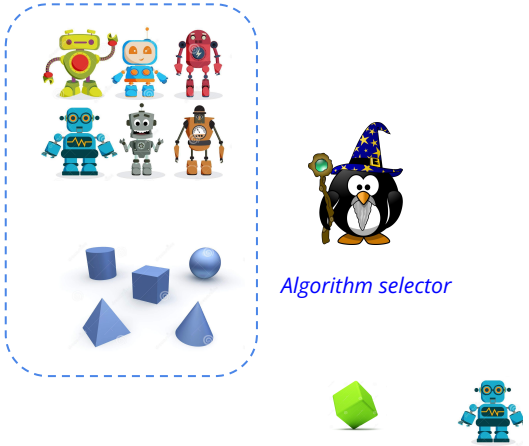
- Sparkle SAT challenge 2018 *(Luo & Hoos, https://ada.liacs.nl/events/sparkle-sat-18/)*
- Sparkle Planning challenge 2019 *(Luo, Vallati & Hoos, https://ada.liacs.nl/events/sparkle-planning-19/)*
- Competition ranking:
  - based on *marginal contribution to performance of an algorithm selector built on a portfolio of all participating solvers.*



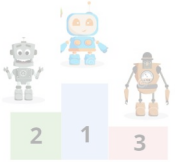
“Cooperative” competition setting



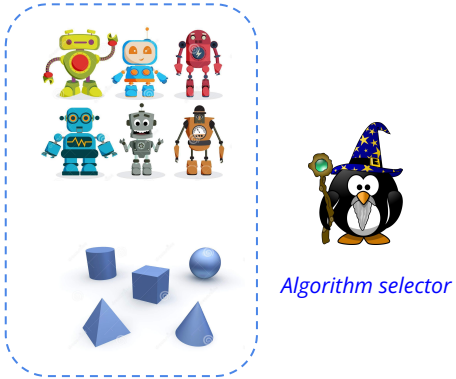
- Sparkle SAT challenge 2018 *(Luo & Hoos, https://ada.liacs.nl/events/sparkle-sat-18/)*
- Sparkle Planning challenge 2019 *(Luo, Vallati & Hoos, https://ada.liacs.nl/events/sparkle-planning-19/)*
- Competition ranking:
  - based on *marginal contribution to performance of an algorithm selector built on a portfolio of all participating solvers.*



“Cooperative” competition setting



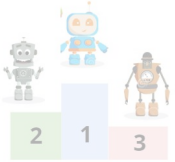
- Sparkle SAT challenge 2018 *(Luo & Hoos, https://ada.liacs.nl/events/sparkle-sat-18/)*
- Sparkle Planning challenge 2019 *(Luo, Vallati & Hoos, https://ada.liacs.nl/events/sparkle-planning-19/)*
- Competition ranking:
  - based on *marginal contribution to performance of an algorithm selector built on a portfolio of all participating solvers.*



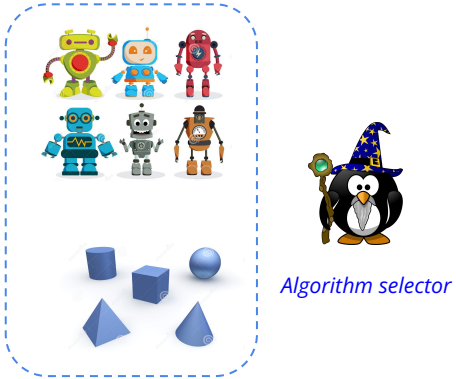
Algorithm selector

Marginal contribution of 

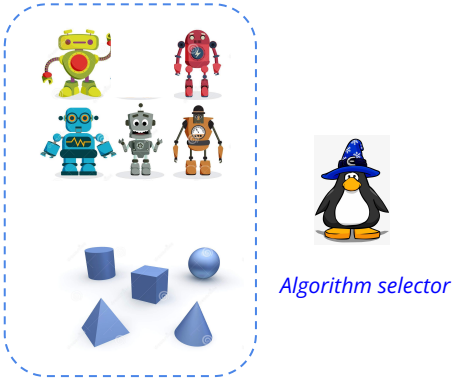
“Cooperative” competition setting



- Sparkle SAT challenge 2018 *(Luo & Hoos, https://ada.liacs.nl/events/sparkle-sat-18/)*
- Sparkle Planning challenge 2019 *(Luo, Vallati & Hoos, https://ada.liacs.nl/events/sparkle-planning-19/)*
- Competition ranking:
  - based on *marginal contribution to performance of an algorithm selector built on a portfolio of all participating solvers.*



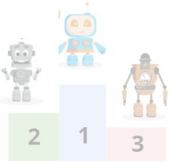
Algorithm selector



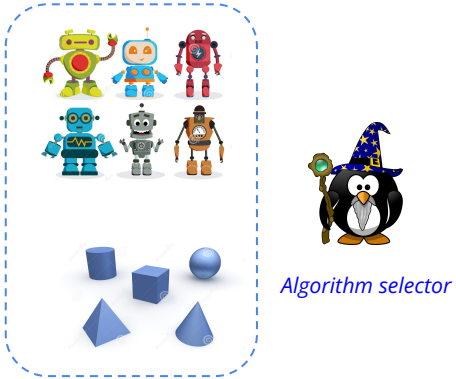
Algorithm selector

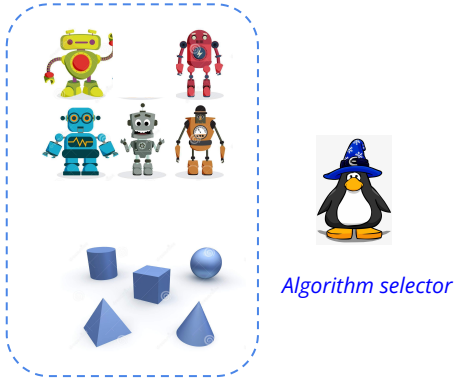
Marginal contribution of 

“Cooperative” competition setting



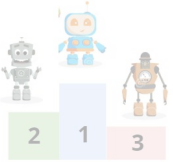
- Sparkle SAT challenge 2018 *(Luo & Hoos, https://ada.liacs.nl/events/sparkle-sat-18/)*
- Sparkle Planning challenge 2019 *(Luo, Vallati & Hoos, https://ada.liacs.nl/events/sparkle-planning-19/)*
- Competition ranking:
  - based on *marginal contribution to performance of an algorithm selector built on a portfolio of all participating solvers.*



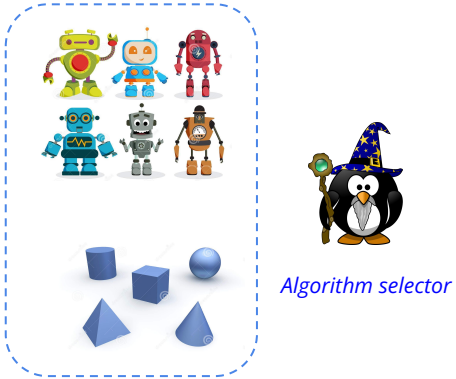


Marginal contribution of  **: performance of**  **– performance of** 

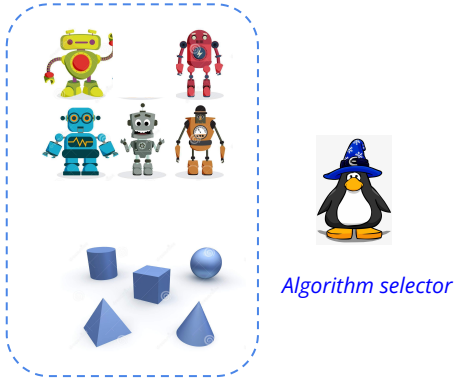
“Cooperative” competition setting



- Sparkle SAT challenge 2018 *(Luo & Hoos, https://ada.liacs.nl/events/sparkle-sat-18/)*
- Sparkle Planning challenge 2019 *(Luo, Vallati & Hoos, https://ada.liacs.nl/events/sparkle-planning-19/)*
- Competition ranking:
  - based on *marginal contribution to performance of an algorithm selector built on a portfolio of all participating solvers.*

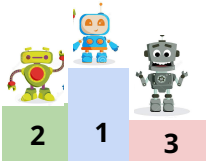


Algorithm selector



Algorithm selector

Marginal contribution of  **:** **performance of**  **–** **performance of** 



MiniZinc Challenges: https://www.minizinc.org/challenge.html

- an annual competition series (2008-present) for benchmarking constraint solving technologies
- *various solving paradigms*: CP, SAT, SMT, MIP & hybrid
- 100 instances each year (20 problems, 5 instances/problem)

MiniZinc Challenges: https://www.minizinc.org/challenge.html

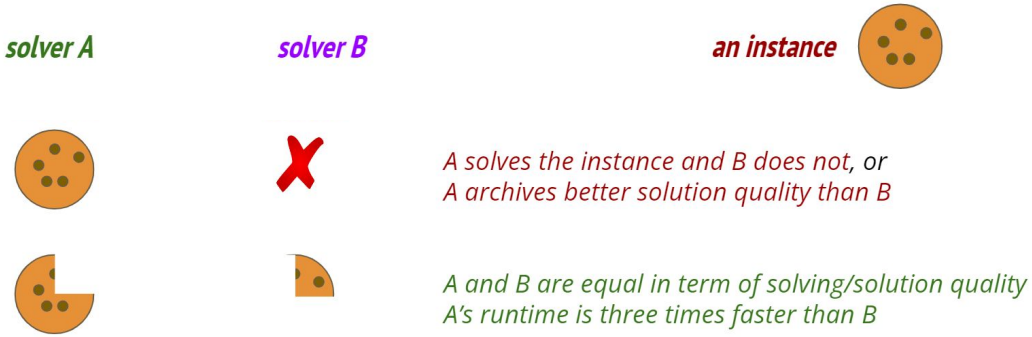
- an annual competition series (2008-present) for benchmarking constraint solving technologies
- *various solving paradigms*: CP, SAT, SMT, MIP & hybrid
- 100 instances each year (20 problems, 5 instances/problem)

- **Competition ranking**
  - MiniZinc scoring method
    - measure relative performance for *a pair of solvers*
    - takes into account both *running time* and *solution quality*

MiniZinc Challenges: https://www.minizinc.org/challenge.html

- an annual competition series (2008-present) for benchmarking constraint solving technologies
- *various solving paradigms*: CP, SAT, SMT, MIP & hybrid
- 100 instances each year (20 problems, 5 instances/problem)

- **Competition ranking**
  - MiniZinc scoring method
    - measure relative performance for *a pair of solvers*
    - takes into account both *running time* and *solution quality*



MiniZinc Challenges: https://www.minizinc.org/challenge.html

- an annual competition series (2008-present) for benchmarking constraint solving technologies
- *various solving paradigms*: CP, SAT, SMT, MIP & hybrid
- 100 instances each year (20 problems, 5 instances/problem)

- **Competition ranking**
  - MiniZinc scoring method
    - measure relative performance for *a pair of solvers*
    - takes into account both *running time* and *solution quality*
  - Borda counting system: produces *a single score for each solver* across all instances
    - For *every* pair of solvers, calculate MiniZinc scores on each instance.
    - *Overall score of each solver*: average MiniZinc scores across all instances.

MiniZinc Challenges: https://www.minizinc.org/challenge.html

- an annual competition series (2008-present) for benchmarking constraint solving technologies
- *various solving paradigms*: CP, SAT, SMT, MIP & hybrid
- 100 instances each year (20 problems, 5 instances/problem)

- **Competition ranking**
  - MiniZinc scoring method
    - measure relative performance for *a pair of solvers*
    - takes into account both *running time* and *solution quality*
  - Borda counting system: produces *a single score for each solver* across all instances
    - For *every* pair of solvers, calculate MiniZinc scores on each instance.
    - *Overall score of each solver*: average MiniZinc scores across all instances.

- **Non-participant solvers**
  - Do not enter the competitions, but are used for computing the Borda scores.

MiniZinc Challenge 2021 Results

Entrants

The entrants for this year (with their descriptions, when provided):

- Choco 4 (description). A Java FD solver.
- flatzingo (description).
- iZplus (description).
- JaCoP (description). A Java FD solver.
- Mistral-2.0 (description).
- OR-Tools (description).
- OscaR/CBLS (description). A constraint-based local search solver written in Scala.
- PicatSAT (description).
- SICStus Prolog (description). A Prolog development environment with a FD constraint programming module.
- Yuck (description). A local search solver written in Scala.

participants

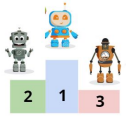
In addition, the challenge organisers entered the following FlatZinc and MiniZinc implementations:

- Chuffed (description). A C++ FD solver using lazy clause generation.
- Geas (description). A C++ lazy clause generation solver with an OCaml FlatZinc frontend.
- Gecode (description). A C++ FD solver.
- MZN/Cbc (description). Translates to MILP, uses Cbc version 2.10.5.
- MZN/CPLEX (description). Translates to MILP, uses IBM ILOG CPLEX Optimizer version 20.10.
- MZN/Gurobi. Translates to MILP, uses Gurobi version 9.1.2.
- sunny-cp¯ (description). A variant of sunny-cp only using the 2020 portfolio CPLEX, Gecode, JaCoP, iZplus, OR-Tools, Picat, SICStus Prolog, Yuck.
- sunny-cp (description). A multi-threaded CP portfolio solver using a 2020 portfolio of CP and MIP solvers incl. Chuffed, Gecode.

non-participants

Taken from:
https://www.minizinc.org/challenge2021/results2021.html

Typical competition setting *with an additional portfolio-based analysis*



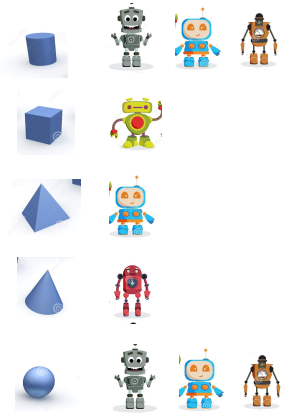
- **Competition ranking**: based on *average performance across all instances*
- **Portfolio-based analysis**: *provide additional insights into complementary strengths among solvers*
  - Step 1: **finding the smallest portfolio that can achieve the best possible performance**

  - Step 2: trade-off between portfolio sizes and performance

  - Step 3: solver importance from a portfolio viewpoint using Shapley values

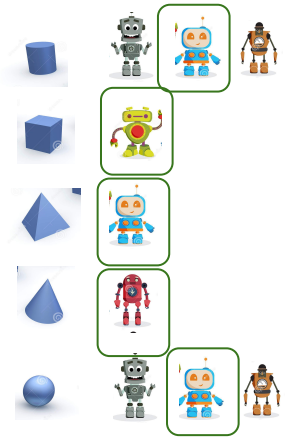
Step 1: finding the smallest portfolio that can achieve the best possible performance

A minimum set cover problem

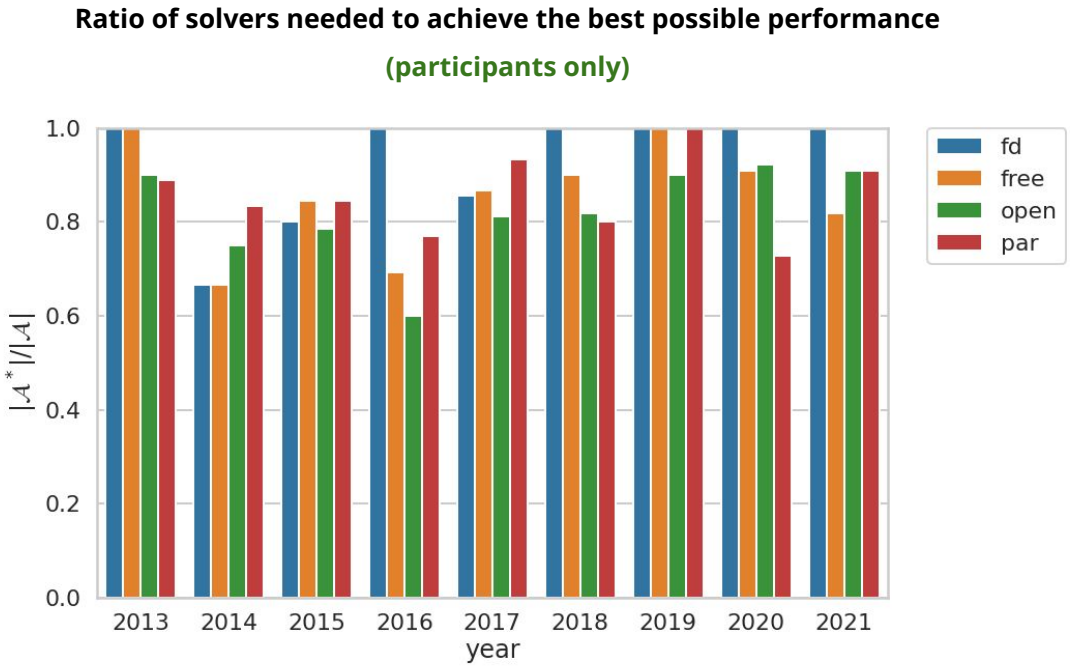


Step 1: finding the smallest portfolio that can achieve the best possible performance

A minimum set cover problem



Step 1: finding the smallest portfolio that can achieve the best possible performance

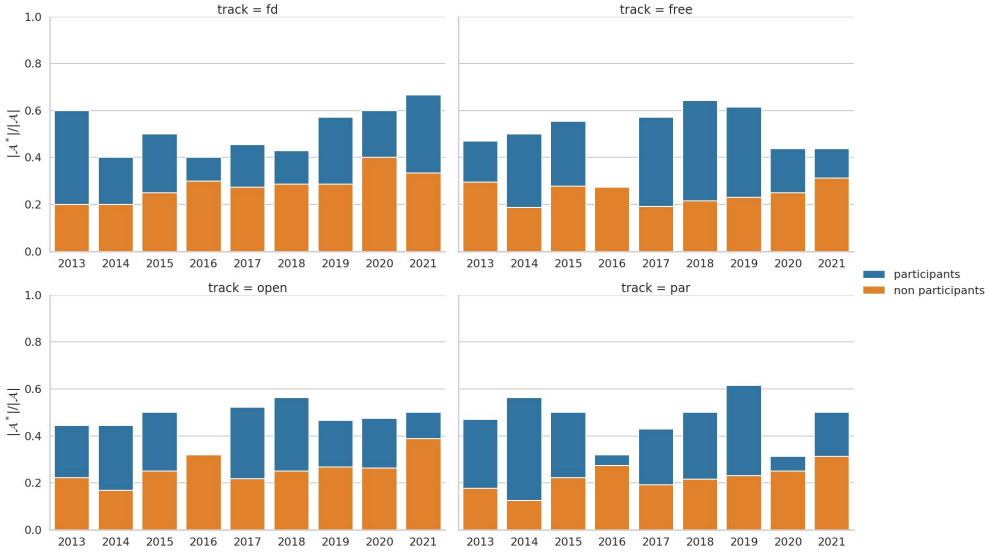


Performance of participant solvers is often highly complementary

Step 1: finding the smallest portfolio that can achieve the best possible performance

Ratio of solvers needed to achieve the best possible performance

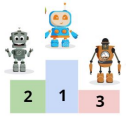
(non-participants included)



Many solvers are completely dominated by others.

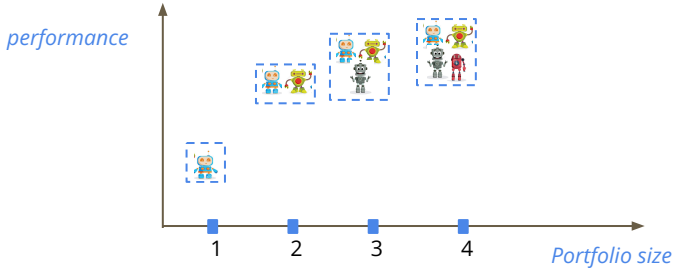
However, in most cases, participants and non-participants are well complementary to each other.

Typical competition setting *with an additional portfolio-based analysis*

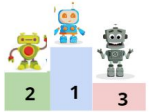


- **Competition ranking**: based on *average performance across all instances*
- **Portfolio-based analysis**: *provide additional insights into complementary strengths among solvers*
  - Step 1: finding the smallest portfolio that can achieve the best possible performance

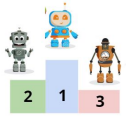
  - **Step 2: trade-off between portfolio sizes and performance**



  - Step 3: solver importance from a portfolio viewpoint using Shapley values

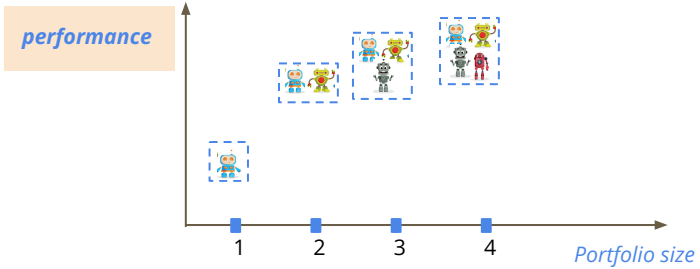


Typical competition setting *with an additional portfolio-based analysis*

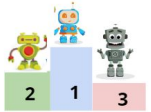


- **Competition ranking**: based on *average performance across all instances*
- **Portfolio-based analysis**: *provide additional insights into complementary strengths among solvers*
  - Step 1: finding the smallest portfolio that can achieve the best possible performance

  - **Step 2: trade-off between portfolio sizes and performance**



  - Step 3: solver importance from a portfolio viewpoint using Shapley values



Measuring performance of a portfolio

- The Virtual Best Solver (VBS) of a portfolio: for each instance, take the best performing solver.
- The Oracle (*O*): the VBS of a portfolio that include *all participant & non-participant solvers*.
- The Participant-Oracle (O_{par}): the VBS of a portfolio that include *all participant solvers*.

Measuring performance of a portfolio

- The Virtual Best Solver (VBS) of a portfolio: for each instance, take the best performing solver.
- The Oracle (*O*): the VBS of a portfolio that include *all participant & non-participant solvers*.
- The Participant-Oracle (O_{par}): the VBS of a portfolio that include *all participant solvers*.
- Performance of a *portfolio A w.r.t the Oracle O*:

$$\mathcal{P}_{\mathcal{O}}(\mathcal{A}) = \frac{score(\mathcal{VBS}(\mathcal{A}))}{score(\mathcal{O})}$$

total MiniZinc scores of the pair of VBS(A) and O across all instances

$$\mathcal{P}_{\mathcal{O}}(\mathcal{A}) \leq 1$$

Measuring performance of a portfolio

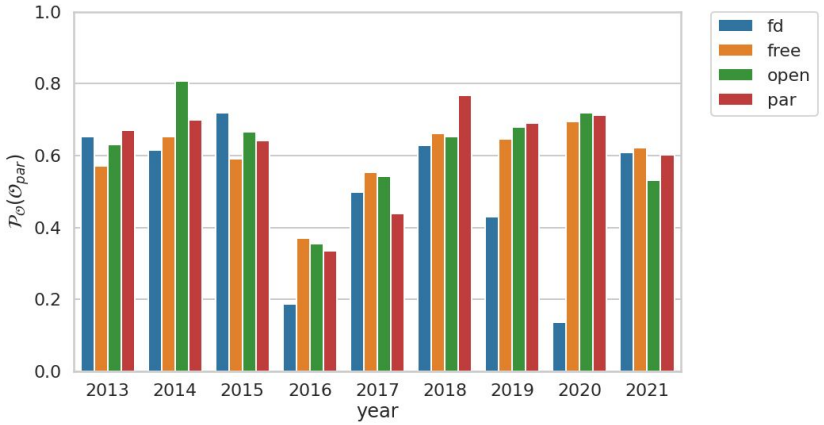
- The Virtual Best Solver (VBS) of a portfolio: for each instance, take the best performing solver.
- The Oracle (*O*): the VBS of a portfolio that include *all participant & non-participant solvers*.
- The Participant-Oracle (O_{par}): the VBS of a portfolio that include *all participant solvers*.
- Performance of a *portfolio A w.r.t the Oracle O*:

$$\mathcal{P}_{\mathcal{O}}(\mathcal{A}) = \frac{score(\mathcal{VBS}(\mathcal{A}))}{score(\mathcal{O})}$$

total MiniZinc scores of the pair of VBS(A) and O across all instances

$$\mathcal{P}_{\mathcal{O}}(\mathcal{A}) \leq 1$$

Performance of the Participant-Oracle w.r.t the Oracle



Step 2: trade-off between portfolio sizes and portfolio performance

(link)

Step 2: trade-off between portfolio sizes and portfolio performance

Best subset of solvers per portfolio size (participants only)

$\mathcal{P}_{\mathcal{O}_{par}}(\mathcal{A})$	$\mathcal{A}$
	year: 2019, track: free
36.1%	or-tools,
55.6%	or-tools, picatsat
67.4%	or-tools, picatsat, sicstus
79.2%	or-tools, picatsat, sicstus, yuck
91.5%	or-tools, picatsat, sicstus, yuck, izplus
96.2%	or-tools, picatsat, sicstus, yuck, izplus, jacop
98.2%	or-tools, picatsat, sicstus, yuck, izplus, jacop, concrete
99.5%	or-tools, picatsat, sicstus, yuck, izplus, jacop, concrete, oscarcbls
100%	or-tools, picatsat, sicstus, yuck, izplus, jacop, concrete, oscarcbls, choco
	year: 2020, track: free
59.7%	or-tools,
71.7%	or-tools, flatzingo
81.0%	or-tools, flatzingo, sicstus
90.2%	or-tools, flatzingo, sicstus, mistral
94.0%	or-tools, flatzingo, sicstus, mistral, oscarcbls
96.9%	or-tools, flatzingo, sicstus, mistral, oscarcbls, picatsat
98.2%	or-tools, flatzingo, sicstus, mistral, oscarcbls, picatsat, choco
99.4%	or-tools, flatzingo, sicstus, mistral, oscarcbls, picatsat, choco, jacop
99.8%	or-tools, flatzingo, sicstus, mistral, oscarcbls, picatsat, choco, jacop, optimathsat-int
100%	or-tools, flatzingo, sicstus, mistral, oscarcbls, picatsat, choco, jacop, optimathsat-int, yuck
	year: 2021, track: free
49.8%	or-tools-cp-sat,
62.0%	or-tools-cp-sat, yuck
75.6%	or-tools-cp-sat, yuck, picatsat
82.9%	or-tools-cp-sat, yuck, picatsat, choco-4-10-7
88.5%	or-tools-cp-sat, yuck, picatsat, choco-4-10-7, jacop
92.1%	or-tools-cp-sat, yuck, picatsat, choco-4-10-7, jacop, coin-or-cbc
95.6%	or-tools-cp-sat, yuck, picatsat, choco-4-10-7, jacop, coin-or-cbc, izplus
97.6%	or-tools-cp-sat, yuck, picatsat, choco-4-10-7, jacop, coin-or-cbc, izplus, mistral-2.0
100%	or-tools-cp-sat, yuck, picatsat, choco-4-10-7, jacop, coin-or-cbc, izplus, mistral-2.0, flatzingo

Step 2: trade-off between portfolio sizes and portfolio performance

Best subset of solvers per portfolio size (participants only)

$\mathcal{P}_{\mathcal{O}_{par}}(\mathcal{A})$	$\mathcal{A}$
	year: 2019, track: free
36.1%	or-tools,
55.6%	or-tools, picatsat
67.4%	or-tools, picatsat, sicstus
79.2%	or-tools, picatsat, sicstus, yuck
91.5%	or-tools, picatsat, sicstus, yuck, izplus
96.2%	or-tools, picatsat, sicstus, yuck, izplus, jacop
98.2%	or-tools, picatsat, sicstus, yuck, izplus, jacop, concrete
99.5%	or-tools, picatsat, sicstus, yuck, izplus, jacop, concrete, oscarcbls
100%	or-tools, picatsat, sicstus, yuck, izplus, jacop, concrete, oscarcbls, choco
	year: 2020, track: free
59.7%	or-tools,
71.7%	or-tools, flatzingo
81.0%	or-tools, flatzingo, sicstus
90.2%	or-tools, flatzingo, sicstus, mistral
94.0%	or-tools, flatzingo, sicstus, mistral, oscarcbls
96.9%	or-tools, flatzingo, sicstus, mistral, oscarcbls, picatsat
98.2%	or-tools, flatzingo, sicstus, mistral, oscarcbls, picatsat, choco
99.4%	or-tools, flatzingo, sicstus, mistral, oscarcbls, picatsat, choco, jacop
99.8%	or-tools, flatzingo, sicstus, mistral, oscarcbls, picatsat, choco, jacop, optimathsat-int
100%	or-tools, flatzingo, sicstus, mistral, oscarcbls, picatsat, choco, jacop, optimathsat-int, yuck
	year: 2021, track: free
49.8%	or-tools-cp-sat,
62.0%	or-tools-cp-sat, yuck
75.6%	or-tools-cp-sat, yuck, picatsat
82.9%	or-tools-cp-sat, yuck, picatsat, choco-4-10-7
88.5%	or-tools-cp-sat, yuck, picatsat, choco-4-10-7, jacop
92.1%	or-tools-cp-sat, yuck, picatsat, choco-4-10-7, jacop, coin-or-cbc
95.6%	or-tools-cp-sat, yuck, picatsat, choco-4-10-7, jacop, coin-or-cbc, izplus
97.6%	or-tools-cp-sat, yuck, picatsat, choco-4-10-7, jacop, coin-or-cbc, izplus, mistral-2.0
100%	or-tools-cp-sat, yuck, picatsat, choco-4-10-7, jacop, coin-or-cbc, izplus, mistral-2.0, flatzingo

OR-Tools is a very strong solver

Step 2: trade-off between portfolio sizes and portfolio performance

Best subset of solvers per portfolio size (participants only)

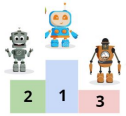
$\mathcal{P}_{\mathcal{O}_{par}}(\mathcal{A})$	$\mathcal{A}$
	year: 2019, track: free
36.1%	or-tools,
55.6%	or-tools, picatsat
67.4%	or-tools, picatsat, sicstus
79.2%	or-tools, picatsat, sicstus, yuck
91.5%	or-tools, picatsat, sicstus, yuck, izplus
96.2%	or-tools, picatsat, sicstus, yuck, izplus, jacop
98.2%	or-tools, picatsat, sicstus, yuck, izplus, jacop, concrete
99.5%	or-tools, picatsat, sicstus, yuck, izplus, jacop, concrete, oscarcbls
100%	or-tools, picatsat, sicstus, yuck, izplus, jacop, concrete, oscarcbls, choco
	year: 2020, track: free
59.7%	or-tools,
71.7%	or-tools, flatzingo
81.0%	or-tools, flatzingo, sicstus
90.2%	or-tools, flatzingo, sicstus, mistral
94.0%	or-tools, flatzingo, sicstus, mistral, oscarcbls
96.9%	or-tools, flatzingo, sicstus, mistral, oscarcbls, picatsat
98.2%	or-tools, flatzingo, sicstus, mistral, oscarcbls, picatsat, choco
99.4%	or-tools, flatzingo, sicstus, mistral, oscarcbls, picatsat, choco, jacop
99.8%	or-tools, flatzingo, sicstus, mistral, oscarcbls, picatsat, choco, jacop, optimathsat-int
100%	or-tools, flatzingo, sicstus, mistral, oscarcbls, picatsat, choco, jacop, optimathsat-int, yuck
	year: 2021, track: free
49.8%	or-tools-cp-sat,
62.0%	or-tools-cp-sat, yuck
75.6%	or-tools-cp-sat, yuck, picatsat
82.9%	or-tools-cp-sat, yuck, picatsat, choco-4-10-7
88.5%	or-tools-cp-sat, yuck, picatsat, choco-4-10-7, jacop
92.1%	or-tools-cp-sat, yuck, picatsat, choco-4-10-7, jacop, coin-or-cbc
95.6%	or-tools-cp-sat, yuck, picatsat, choco-4-10-7, jacop, coin-or-cbc, izplus
97.6%	or-tools-cp-sat, yuck, picatsat, choco-4-10-7, jacop, coin-or-cbc, izplus, mistral-2.0
100%	or-tools-cp-sat, yuck, picatsat, choco-4-10-7, jacop, coin-or-cbc, izplus, mistral-2.0, flatzingo

4th place in competition ranking

second-to-last in competition ranking

Solvers that look weak in a traditional competition ranking may actually be very well complementary to the winner.

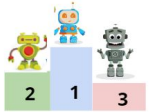
Typical competition setting *with an additional portfolio-based analysis*



- **Competition ranking**: based on *average performance across all instances*
- **Portfolio-based analysis**: *provide additional insights into complementary strengths among solvers*
  - Step 1: finding the smallest portfolio that can achieve the best possible performance

  - Step 2: trade-off between portfolio sizes and performance

  - **Step 3: solver importance from a portfolio viewpoint using Shapley values**



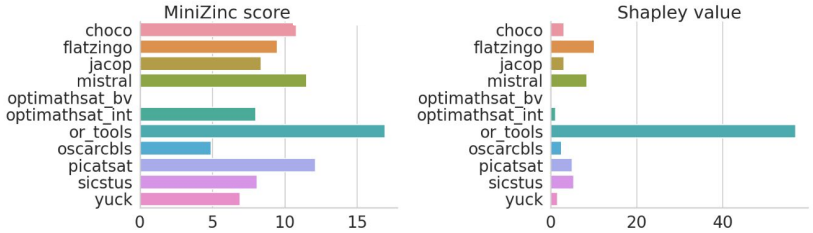
Step 3: solver importance from a portfolio viewpoint using Shapley values

- Shapley values: a concept in coalitional game theory
- Fréchette, A., Kotthoff, L., Michalak, T., Rahwan, T., Hoos, H. and Leyton-Brown, K. *Using the shapley value to analyze algorithm portfolios*. In *Proceedings of the AAAI Conference on Artificial Intelligence, 2016*
- Shapley values of a solver S in a portfolio A: ***total*** *marginal contribution* of S on ***all subsets*** of A *(using the VBS)*

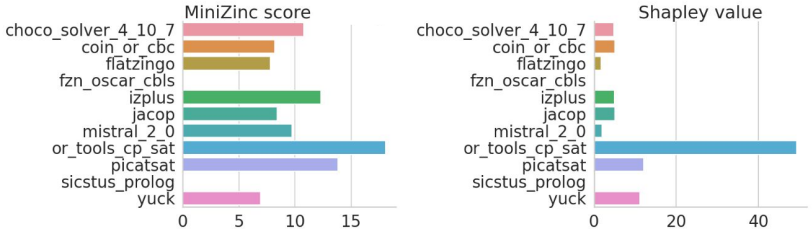
Step 3: solver importance from a portfolio viewpoint using Shapley values

- Shapley values: a concept in coalitional game theory
- Fréchette, A., Kotthoff, L., Michalak, T., Rahwan, T., Hoos, H. and Leyton-Brown, K. *Using the shapley value to analyze algorithm portfolios*. In *Proceedings of the AAAI Conference on Artificial Intelligence, 2016*
- Shapley values of a solver S in a portfolio A: ***total** marginal contribution* of S on ***all subsets*** of A *(using the VBS)*

2020 - free



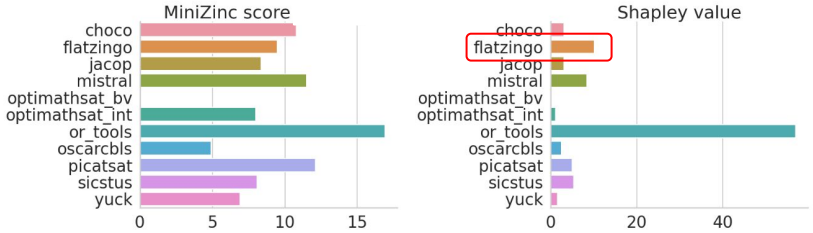
2021 - free



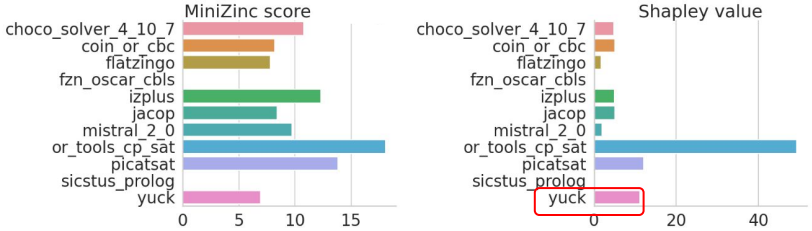
Step 3: solver importance from a portfolio viewpoint using Shapley values

- Shapley values: a concept in coalitional game theory
- Fréchette, A., Kotthoff, L., Michalak, T., Rahwan, T., Hoos, H. and Leyton-Brown, K. *Using the shapley value to analyze algorithm portfolios*. In *Proceedings of the AAAI Conference on Artificial Intelligence, 2016*
- Shapley values of a solver S in a portfolio A: ***total*** *marginal contribution* of S on ***all subsets*** of A *(using the VBS)*

2020 - free



2021 - free



Summary

- Traditional ranking method in competition settings is a good way to measure performance of solvers, but it does not necessarily reveal the full potential of a solver.
- An additional portfolio-based analysis can provide further insights on the complementary strengths of solvers
    - Code and data are available at: *https://github.com/ndangtt/portfolio-based-analysis*