Evolutionary Diversity Optimization

Introduction and Recent Results

Aneta Neumann, Frank Neumann
The University of Adelaide

Evolutionary Algorithms (EAs)

- Evolutionary algorithms are general purpose algorithms.
- follow Darwin's principle (survival of the fittest).
- work with a set of solutions called population.
- parent population produces offspring population by variation operators (mutation, crossover).
- select individuals from the parents and children to create new parent population.
- Iterate the process until a "good solution" has been found.
- EAs are adaptive and often yield good solutions for complex, dynamic and/or stochastic problems

Motivation

- Diversity plays a crucial role in evolutionary computation
- Diversity
  - prevents premature convergence
  - enables successful crossover
  - allows to compute set of Pareto optimal solutions for multi-objective problems

Diversity

- Majority of approaches consider diversity in the objective space.
- Ulrich/Thiele considered diversity in the search space (Tamara Ulrich's PhD thesis "Exploring Structural Diversity in Evolutionary Algorithms", ETH Zurich, 2012).
- Diversity with respect to other properties (features) could be useful in various domains.

Goal:

- Compute a set of good solutions that differ in terms of interesting properties/features.
  - Think of good designs that vary with respect to important properties.
- The goal is to maximize diversity for a set of high quality solutions.
- This is different from the standard use of diversity in evolutionary computation where diversity is used to avoid premature convergence.

Evolutionary Diversity Optimisation (EDO)

Here, we aim for a set of solutions (P) for a given optimisation problem that all have acceptable quality but differ in terms of some structural properties:

$D(P) \rightarrow Max$

st:

$c(p_i) \leq c_{max} \qquad \forall\, p_i \in P$

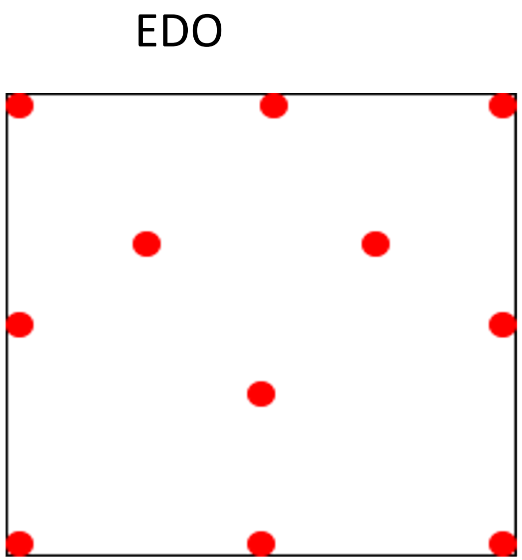
Advantages:

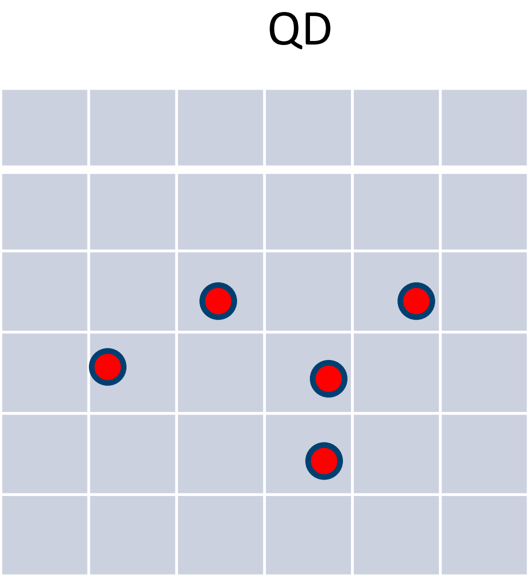
- Enables decision-makers to choose from different alternatives.
- Provides us with valuable information on the solution space.
- Aides when the problem slightly changes.

Quality Diversity

- Quality Diversity (QD) approaches aim to compute the best possible solution in sections of a given behavioral/feature space.
- This provides a different way of solving a problem compared to classical approaches.
- It gives high quality solutions with respect to different features combinations and the use of QD often also leads to an overall better performing solution.
- QD has been mainly used in the area of robotics, games, etc.
- Some recent studies on using it for combinatorial optimization problems.

EDO vs QD





- EDO works with fixed population size and imposes a diversity measure. It aims to maximize diversity among sets of solutions meeting a given quality criterion.
- QD (map elites versions) works with increasing population size and partitions feature/behavioral space into boxes and improve the quality of the solution in each box.

Application Areas

Optimisation:

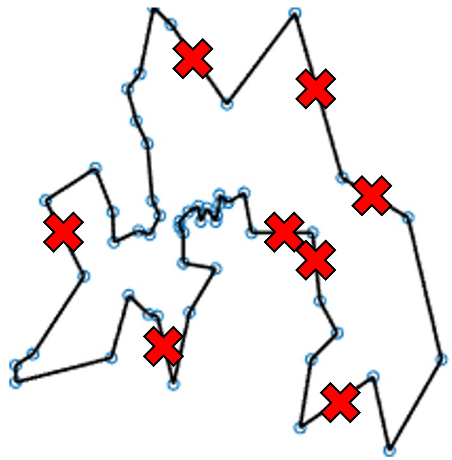
- Present set of diverse high quality solutions (instead of single one) to enable discussion for further refinement.
- See how good solutions distribute with respect to components and/or important features of solutions.

Automated machine learning (AutoML):

- Understand algorithm performance with respect to important features through diverse problem instances.
- Construct diverse sets of problem instances for algorithm selection and configuration.

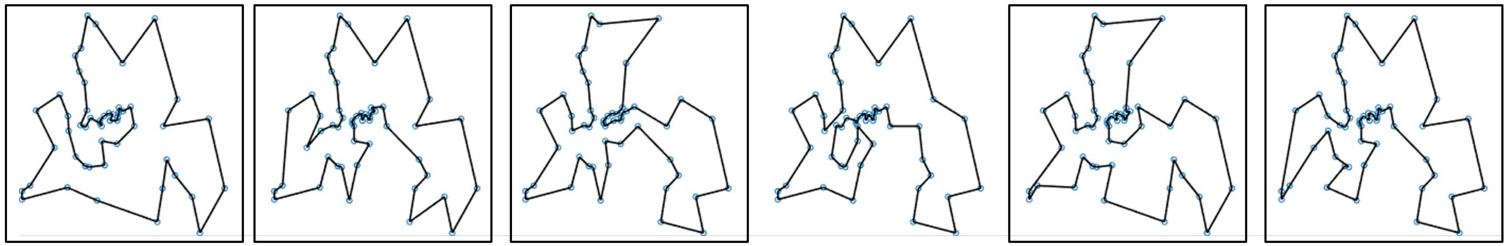
Diverse Sets of Solutions in Optimization

berlin52.tsp



Optimal tour

1.05-approx.



Diversity optimization for the detection and concealment of spatially defined communication networks.

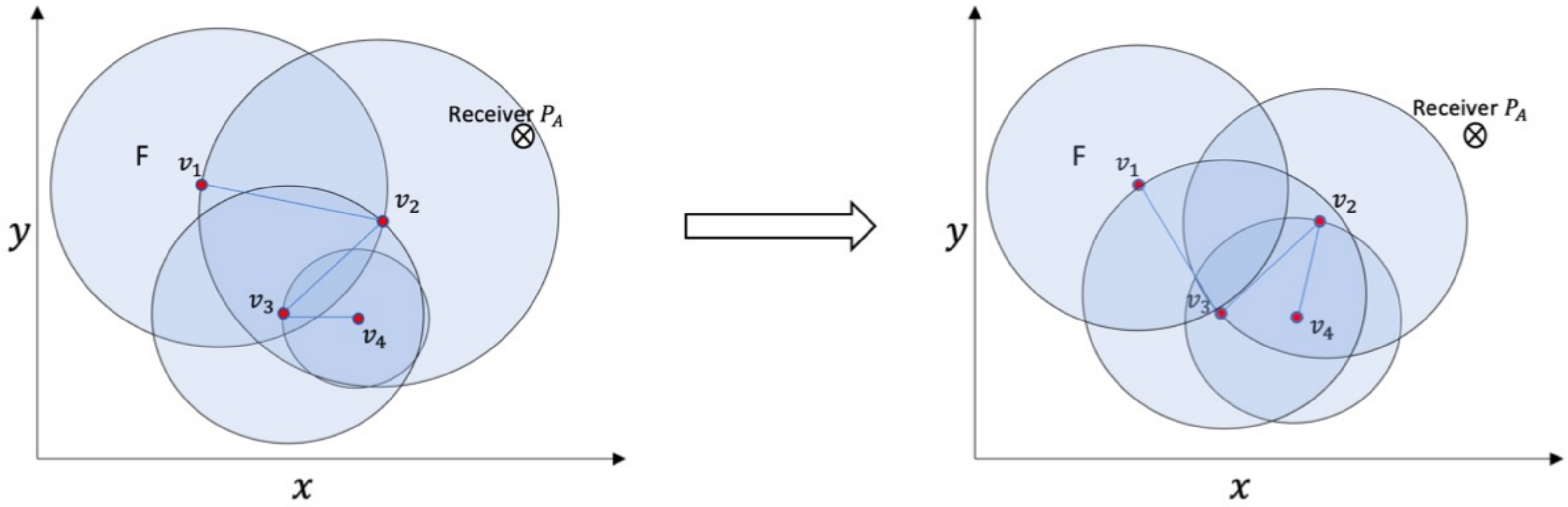
(A. Neumann, S. Goulder, X. Yan, G. Sherman, B. Campbell, M. Guo, F. Neumann, GECCO 2023)

Problem in Communication Networks

- We consider the problem of constructing a wireless communication network for a given set of entities.

- Goals:
- Enable communication between entities but minimize the area covered by the senders' transmissions while also avoiding adversaries that may observe the communication.
- Compute diverse sets of high-quality solutions to provide a variety of structural different options to decision makers.

Adversarial Minimum Area Spanning Forest (AMASF) Problem



Opitmization problem:
Find spanning forest with minimal number of connected components that minimizes the area.

Diversity optimization problem:
Compute a set of high quality solutions that has edges distributed as equally as possible.

EDO for AMASF

Area of MSF

Algorithm 1 $(\mu + \lambda)$ EA for Diversity Optimization

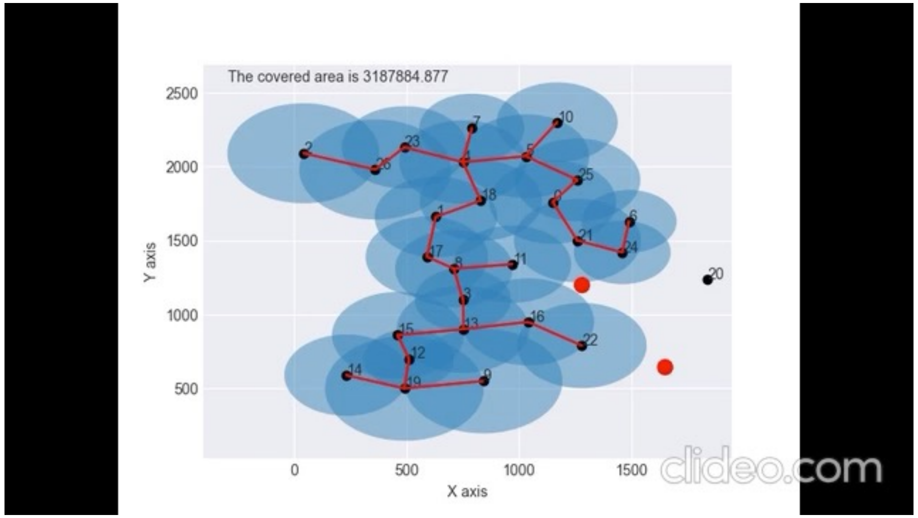
Initialize the population P with μ spanning forests such that $|A(F)| \leq (1+\alpha) \cdot |A(MSF)|$ for all $F \in P$.
while termination criterion is not reached **do**
  let $C \subseteq P$ where $|C| = \lambda$.
  for $F \in C$ **do**
    produce an offspring F' of F by applying mutation.
    if $|A(F')| \leq (1+\alpha) \cdot |A(MSF)|$ **then**
      add F' to P.
    end if
  end for
  $F^* = \arg\min_{F \in P} |A(F)|$
  while $|P| > \mu$ **do**
    let $\hat{F} = \arg\max_{F \in P \setminus \{F^*\}} D(P \setminus \{F\})$.
    remove $\hat{F}$ from P.
  end while
end while
return P.

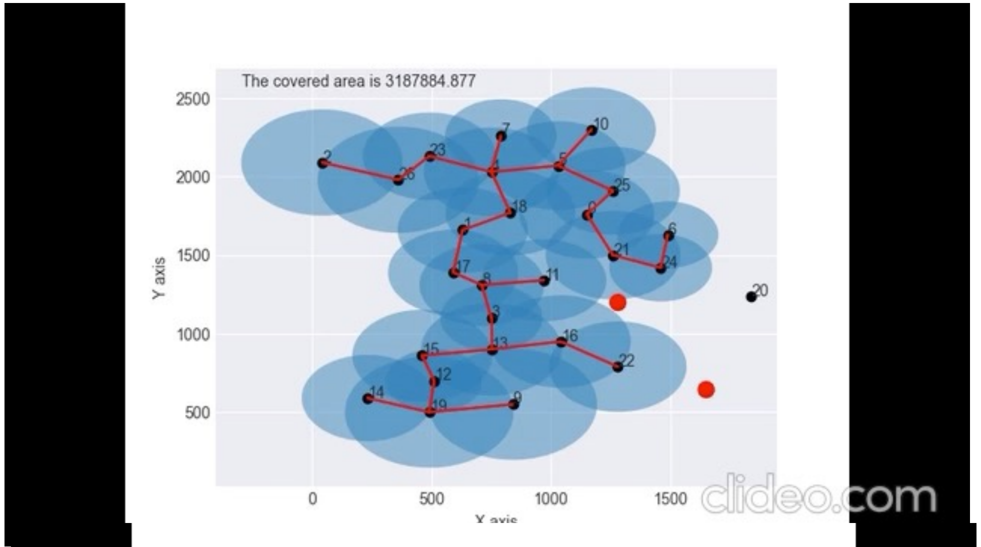
1 Initialization

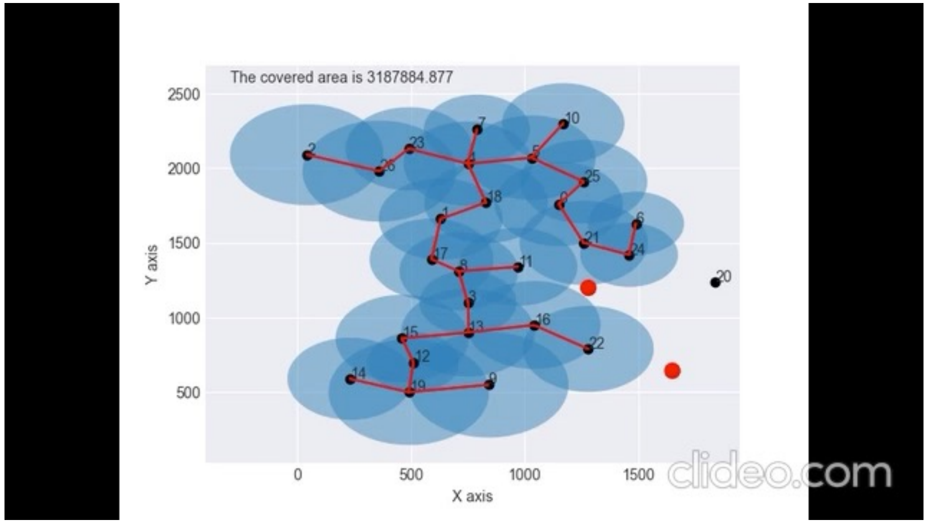
2 Offspring population

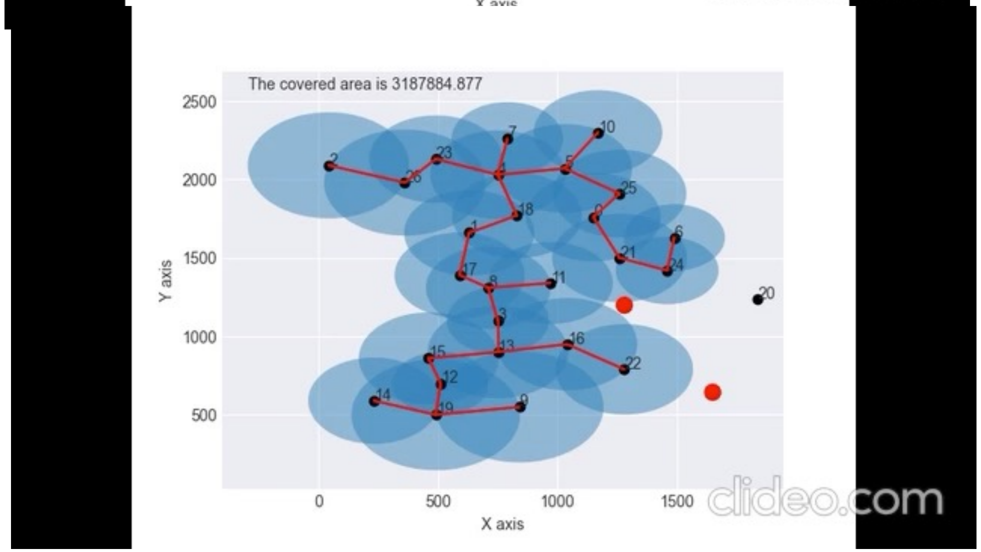
3 Diversity Based selection

Visualization

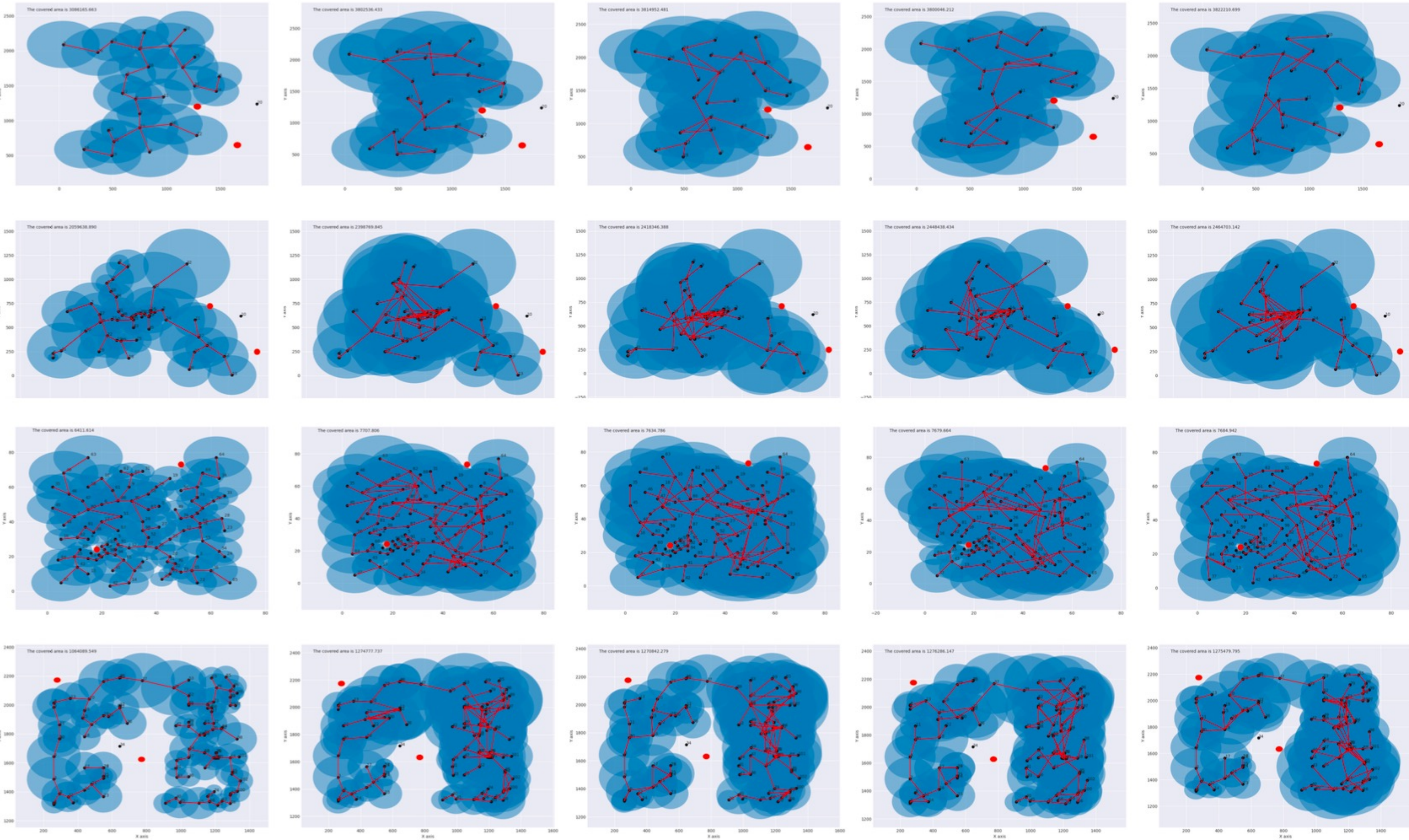








EDO for Communication Networks



Evolutionary diversity optimisation in constructing satisfying assignments.

(A. Nikfarjam, R. Rothenberger, F. Neumann, T. Friedrich, GECCO 2023)

SAT

- SAT include determining the existence of an assignment satisfying a Boolean formula. Consider $X = \{x_1, \dots, x_n\}$, a Boolean formula in CNF can be:

$$
\begin{aligned}
&(x_4 \vee \neg x_{12} \vee x_2) \\
&\wedge \\
&(\neg x_{n-2} \vee \neg x_8 \vee x_1) \\
&\wedge \\
&\dots \\
&\wedge \\
&(x_5 \vee \neg x_n \vee x_{11})
\end{aligned}
$$

Algorithms

Here, we use a SAT solver (minisat, Niklas Eén and Niklas Sörensson (2003)) to solve a formula.

Question: how can we make the solver to generate a diverse set of assignments?

By adding new clauses, we can force a SAT solver to generate distinctive assignments.

Algorithms:

1. Basic
2. Bitflip EA
3. EDO algorithm

Basic algorithm

- Consider the following assignment:

F	**T**	**T**	**F**	**F**	**T**	**T**

- We can forbid it by adding the following clause:

$$(x_1 \vee \neg x_2 \vee \neg x_3 \vee x_4 \vee x_5 \vee \neg x_6 \vee \neg x_6)$$

Bitflip algorithm

- Consider the following assignment:

F	T	T	F	F	T	T

- We can fix the negated variables by adding the following clauses:

$$(\neg x_3)$$
$$\wedge$$
$$(x_5)$$

- Then, we iteratively generate an offspring using a SAT solver, and remove an individual with the least contribution to the diversity.

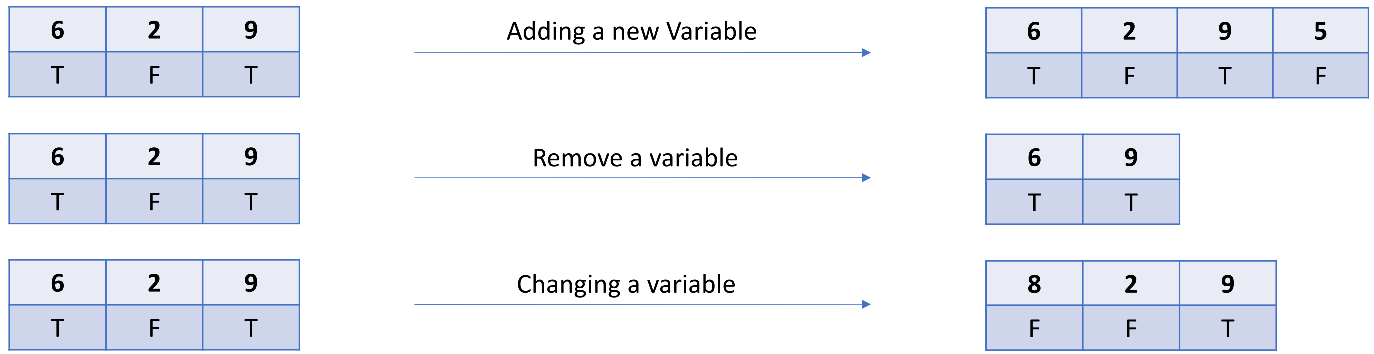
EDO algorithm

- Consider the following assignment:

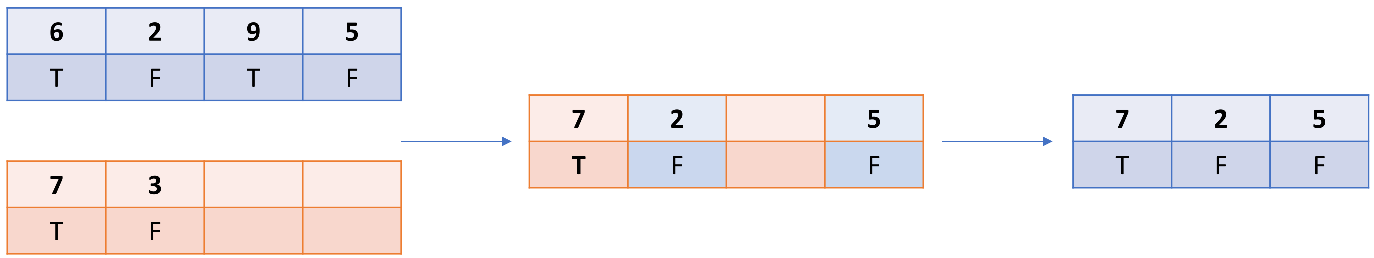
3	5
T	F

F	T	T	F	F	T	T

Operators



(a) Mutation



Add empty cells to the parent with less variables to have an equal size.

Independently select each variable and its values from the parents.

Remove the empty cells.

(b) Crossover

Results

Power law distribution instances

	Basic 1			Bit-flip 2			EDO 3 Mutation			EDO 3 Crossover+Mutation		
m	H_1	H_2	Stat (1)	H_1	H_2	Stat (2)	H_1	H_2	Stat (3)	H_1	H_2	Stat (4)
210	0.055	0.016	$2^-3^-4^-$	0.753	0.839	$1^+3^-4^-$	0.962	0.959	$1^+2^+4^*$	0.953	0.955	$1^+2^+3^*$
220	0.052	0.011	$2^-3^-4^-$	0.721	0.818	$1^+3^-4^-$	0.945	0.938	$1^+2^+4^*$	0.932	0.933	$1^+2^+3^*$
230	0.055	0.019	$2^-3^-4^-$	0.738	0.823	$1^+3^-4^-$	0.937	0.932	$1^+2^+4^*$	0.925	0.925	$1^+2^+3^*$
240	0.046	0.007	$2^-3^-4^-$	0.731	0.808	$1^+3^-4^-$	0.933	0.927	$1^+2^+4^*$	0.921	0.924	$1^+2^+3^*$
250	0.171	0.135	$2^-3^-4^-$	0.774	0.851	$1^+3^-4^-$	0.928	0.918	$1^+2^+4^*$	0.911	0.915	$1^+2^+3^*$
260	0.114	0.075	$2^-3^-4^-$	0.765	0.832	$1^+3^-4^-$	0.925	0.909	$1^+2^+4^*$	0.914	0.904	$1^+2^+3^*$
270	0.089	0.061	$2^-3^-4^-$	0.757	0.823	$1^+3^-4^-$	0.911	0.893	$1^+2^+4^*$	0.896	0.886	$1^+2^+3^*$
280	0.172	0.143	$2^-3^-4^-$	0.76	0.828	$1^+3^-4^-$	0.907	0.897	$1^+2^+4^*$	0.886	0.885	$1^+2^+3^*$
290	0.14	0.083	$2^-3^-4^-$	0.826	0.842	$1^+3^-4^-$	0.912	0.878	$1^+2^+4^+$	0.9	0.874	$1^+2^+3^-$
300	0.272	0.235	$2^-3^-4^-$	0.825	0.825	$1^+3^-4^-$	0.902	0.856	$1^+2^+4^*$	0.895	0.857	$1^+2^+3^*$
310	0.191	0.156	$2^-3^-4^-$	0.776	0.777	$1^+3^-4^*$	0.862	0.814	$1^+2^+4^*$	0.844	0.806	$1^+2^*3^*$
320	0.099	0.051	$2^-3^-4^-$	0.478	0.424	$1^+3^-4^*$	0.611	0.489	$1^+2^+4^*$	0.591	0.478	$1^+2^*3^*$
330	0.169	0.135	$2^-3^-4^-$	0.544	0.503	$1^+3^-4^*$	0.666	0.56	$1^+2^+4^*$	0.643	0.547	$1^+2^*3^*$
340	0.182	0.129	$2^-3^-4^-$	0.627	0.562	$1^+3^-4^-$	0.73	0.611	$1^+2^+4^*$	0.717	0.603	$1^+2^+3^*$
350	0.157	0.113	$2^-3^-4^-$	0.534	0.496	$1^+3^-4^*$	0.61	0.532	$1^+2^+4^*$	0.605	0.531	$1^+2^*3^*$
360	0.089	0.047	$2^-3^-4^-$	0.531	0.501	$1^+3^-4^*$	0.606	0.537	$1^+2^+4^*$	0.6	0.535	$1^+2^*3^*$
370	0.156	0.11	$2^-3^-4^-$	0.425	0.339	$1^+3^-4^-$	0.535	0.394	$1^+2^+4^*$	0.529	0.392	$1^+2^+3^*$
380	0.161	0.121	$2^-3^-4^-$	0.437	0.344	$1^+3^-4^*$	0.498	0.375	$1^+2^+4^*$	0.491	0.372	$1^+2^*3^*$

Results

Uniform distribution instances

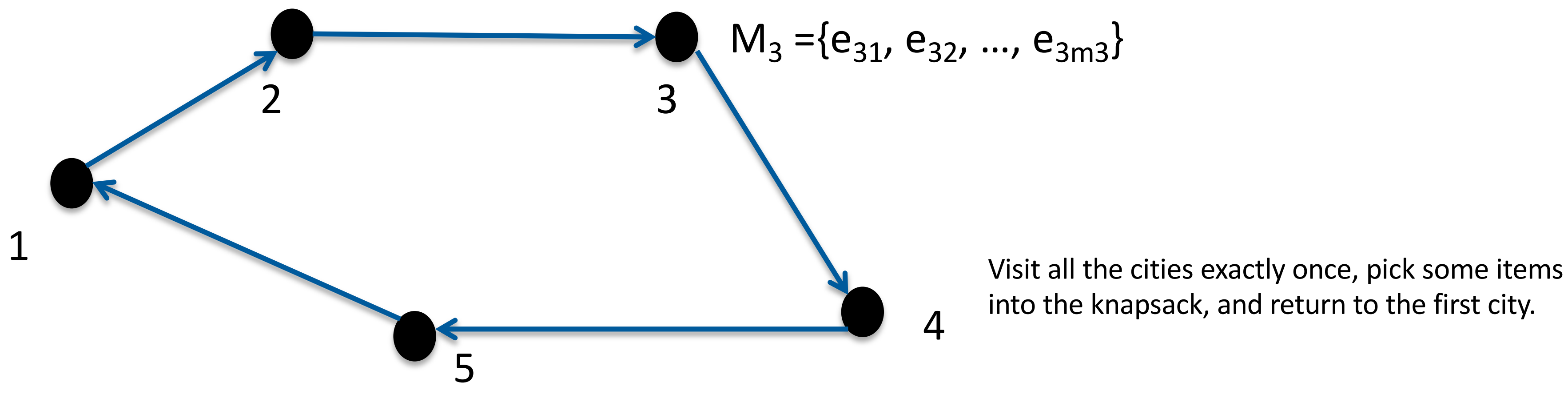
	Basic 1			Bit-flip 2			EDO 3 Mutation			EDO 3 Crossover+Mutation		
m	H_1	H_2	Stat (1)	H_1	H_2	Stat (2)	H_1	H_2	Stat (3)	H_1	H_2	Stat (4)
270	0.295	0.28	$2^-3^-4^-$	0.859	0.889	$1^+3^-4^-$	0.942	0.947	$1^+2^+4^*$	0.94	0.948	$1^+2^+3^*$
280	0.241	0.217	$2^-3^-4^-$	0.867	0.879	$1^+3^-4^-$	0.944	0.943	$1^+2^+4^*$	0.944	0.946	$1^+2^+3^*$
290	0.202	0.186	$2^-3^-4^-$	0.834	0.848	$1^+3^-4^-$	0.937	0.938	$1^+2^+4^*$	0.939	0.941	$1^+2^+3^*$
300	0.183	0.175	$2^-3^-4^-$	0.877	0.888	$1^+3^-4^-$	0.943	0.943	$1^+2^+4^*$	0.946	0.946	$1^+2^+3^*$
310	0.09	0.078	$2^-3^-4^-$	0.875	0.893	$1^+3^-4^-$	0.943	0.946	$1^+2^+4^*$	0.945	0.948	$1^+2^+3^*$
320	0.062	0.051	$2^-3^-4^-$	0.884	0.894	$1^+3^-4^-$	0.936	0.939	$1^+2^+4^*$	0.937	0.94	$1^+2^+3^*$
330	0.157	0.137	$2^-3^-4^-$	0.885	0.895	$1^+3^-4^-$	0.927	0.927	$1^+2^+4^*$	0.932	0.934	$1^+2^+3^*$
340	0.135	0.117	$2^-3^-4^-$	0.898	0.905	$1^+3^-4^-$	0.928	0.927	$1^+2^+4^*$	0.933	0.933	$1^+2^+3^*$
350	0.073	0.062	$2^-3^-4^-$	0.895	0.903	$1^+3^-4^-$	0.916	0.918	$1^+2^+4^*$	0.918	0.92	$1^+2^+3^*$
360	0.08	0.067	$2^-3^-4^-$	0.866	0.875	$1^+3^-4^-$	0.893	0.896	$1^+2^+4^*$	0.898	0.903	$1^+2^+3^*$
370	0.084	0.07	$2^-3^-4^-$	0.851	0.862	$1^+3^-4^-$	0.884	0.886	$1^+2^+4^*$	0.891	0.895	$1^+2^+3^*$
380	0.058	0.042	$2^-3^-4^-$	0.846	0.855	$1^+3^-4^-$	0.876	0.879	$1^+2^+4^*$	0.877	0.88	$1^+2^+3^*$
390	0.178	0.178	$2^-3^-4^-$	0.822	0.822	$1^+3^*4^-$	0.832	0.829	$1^+2^*4^*$	0.835	0.832	$1^+2^+3^*$
400	0.226	0.215	$2^-3^-4^-$	0.637	0.622	$1^+3^-4^-$	0.648	0.63	$1^+2^+4^*$	0.647	0.629	$1^+2^+3^*$
410	0.105	0.098	$2^-3^-4^-$	0.674	0.669	$1^+3^-4^-$	0.693	0.685	$1^+2^+4^*$	0.693	0.684	$1^+2^+3^*$
420	0.125	0.118	$2^-3^-4^-$	0.603	0.592	$1^+3^*4^-$	0.612	0.599	$1^+2^*4^*$	0.613	0.6	$1^+2^+3^*$
430	0.153	0.146	$2^-3^-4^-$	0.311	0.299	$1^+3^*4^-$	0.326	0.309	$1^+2^*4^*$	0.326	0.309	$1^+2^+3^*$
440	0.059	0.047	$2^-3^-4^-$	0.352	0.335	$1^+3^-4^-$	0.366	0.346	$1^+2^+4^*$	0.366	0.347	$1^+2^+3^*$

A Quality Diversity Approach for the Traveling Thief Problem

(A. Nikfarjam, A. Neumann, F. Neumann, GECCO 2022 (Best Paper Nominee), ACM TELO)

Traveling Thief Problem (TTP)

- TTP combines TSP and KP into a multi-component problem
- Given n cities i, 1 ≤ i ≤ n, distances d_{ij} between them, and for each city i a set of items M_i (each item has a profit and weight), find a tour (Π) and a packing plan (P) such that the overall benefit is **maximal**.



- Vehicle (thief) travels along the chosen tour and weight of already packed items slows down the vehicle (thief).

Traveling Thief Problem (TTP)

Renting rate is paid for the knapsack per time unit

Travel from city i to i+1 in π

- Fitness is given by

$$Z([\Pi, P]) = \sum_{i=1}^{n}\sum_{k=1}^{m_i} p_{ik}y_{ik} - R\left(\frac{d_{x_n x_1}}{\upsilon_{max} - \nu W_{x_n}} + \sum_{i=1}^{n-1}\frac{d_{x_i x_{i+1}}}{\upsilon_{max} - \nu W_{x_i}}\right)$$

traveling speed

$$\nu = \frac{\upsilon_{max} - \upsilon_{min}}{W}$$

cumulative weight of the items

profits

Travel from last city to first city

- We aim to compute a diverse set of high-quality solutions differing in TSP and KP score.
- Behavioral descriptor presents the length of the tour (f) and the value of items collected (g),

4

Heuristics

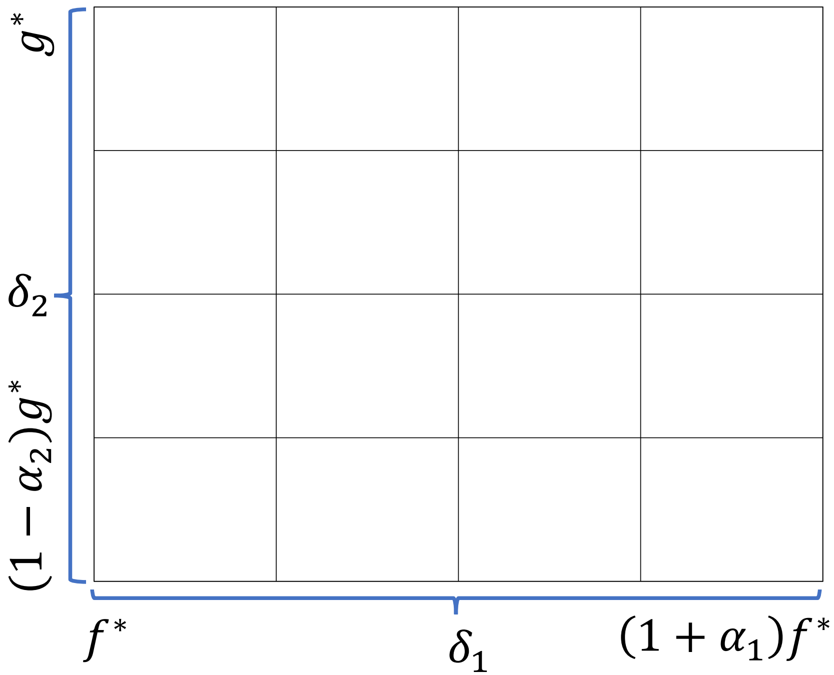
- TTP consists of two parts:
  - Choose TSP tour
  - Choose packing plan

Bi-level Map-Elites-based Evolutionary Algorithm

- For TSP tour often popular TSP heuristics are used:
  - EAX
  - 2-OPT
- Packing plan:
  - Dynamic Programming
  - (1+1) EA / bit flip mutations

Map for Quality Diversity TTP Approach

- Map for traveling thief problem
- f* is the cost of optimal TSP tour
- g* is the optimal profit for the knapsack problem



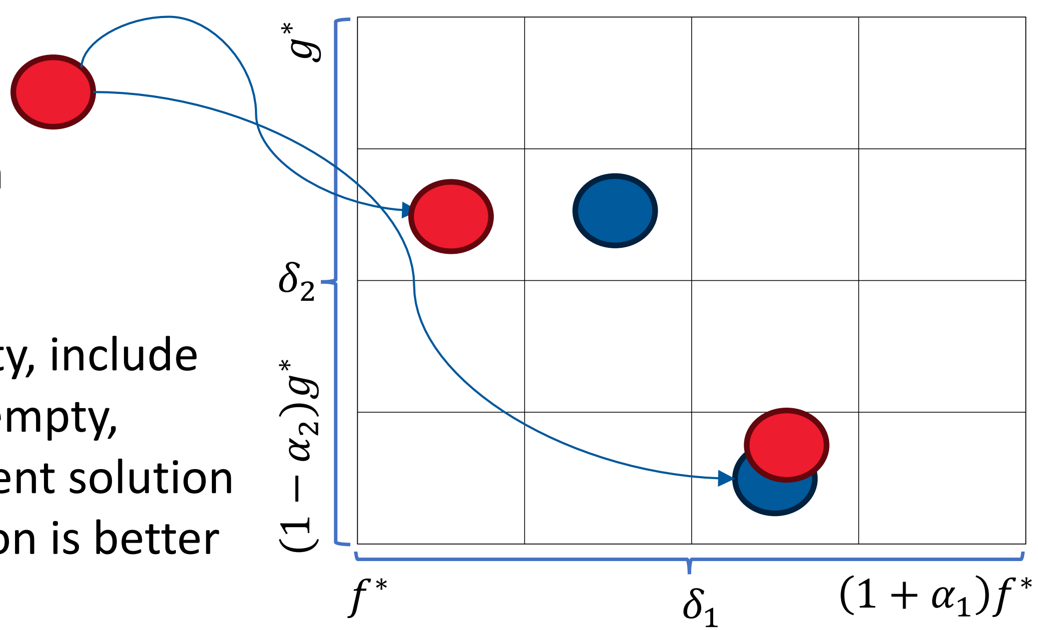
The representation of an empty map

Visualising the distribution of high-quality TTP solutions

MAP Elites for TTP

New solution
Map to cell

If cell is empty, include
If cell is not empty,
Replace current solution
If new solution is better



Choice for investigated TTP instance:
$\alpha_1 = 0.05, \alpha_2 = 0.2$

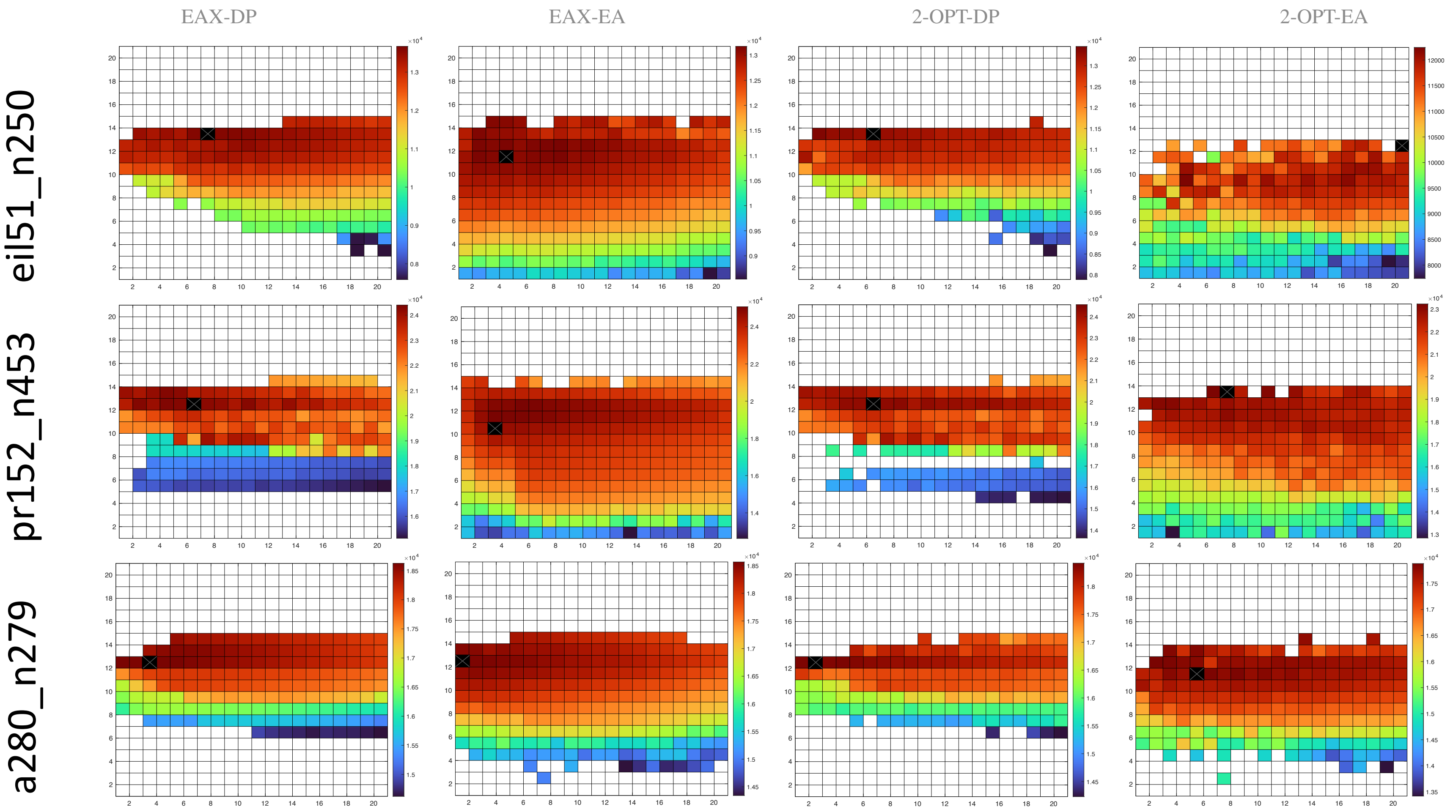
Algorithm 1 The MAP-Elites-Based Evolutionary Algorithm

```
Find the optimal/near-optimal values of the TSP and the KP by algorithms in Nagata and Kobayashi [2013], Toth [1980], respectively.
Generate an empty map and populate it with the initialising procedure.
while termination criterion is not met do
    Generate an offspring and calculate the TSP and the KP scores.
    if The TSP and the KP scores are within α1%, and α2% gaps to the optimal values of BD. then
        Find the corresponding cell to the TSP and the KP scores.
        if The cell is empty then
            Store the offspring in the cell.
        else
            Compare the offspring and the individual occupying the cell and store the best individual in terms of TTP score in the cell.
```

Resulting Maps for different Approaches on strongly correlated TTP Instances

Cells are coloured based on the average TTP scores

$\alpha_1 = 0.05, \alpha_2 = 0.2$



Illuminate the distribution of the solutions over the space of f and g

1+1 EA populates larger part of the map

The distribution of TTP solutions of the four competitors over the behaviour space

Results for EAX based QD Algorithms

Table 3. Performance of the MAP-Elites-based Approach in Terms of the TTP Score

In.	EAX-EA (1) Average	Stat	Best	CPU time	2-OPT-EA (2) Average	Stat	Best	CPU time	Best-known value
19	32625.3	2^{+}	33092	835	29687.5	1^{-}	30065.8	728	32993.1
20	18975.9	2^{+}	19188.4	708	17622.2	1^{-}	17803.8	685	19379.7
21	35175.8	2^{+}	35512.2	696	33456.6	1^{-}	34455.2	674	35015.2
22	642.3	2^{+}	1137.5	1625	-4337	1^{-}	-2693.5	1696	893.4
23	51988.6	2^{+}	52651.8	1624	48382.8	1^{-}	49830.4	1685	51303.4
24	29201.5	2^{+}	32072.9	1627	25214.6	1^{-}	25618	1620	28304
25	104549.9	2^{+}	105434.5	7937	95300.4	1^{-}	96468.9	7975	105908.1
26	71829.9	2^{+}	73152.8	6914	67954.6	1^{-}	69060.6	7285	72308.7
27	107975.3	2^{+}	109395.1	6848	104852.4	1^{-}	106735	7091	108236.1
28	258901.5	2^{+}	260839.7	38669	238212.3	1^{-}	240916	45390	263040.2
29	129168.4	2^{+}	131072	36670	122606.8	1^{-}	123626.8	39520	131486.2
30	230888.9	2^{+}	237097.6	32136	225694.2	1^{-}	227466.4	31796	233343

Quality diversity approaches for time-use optimisation to improve health outcomes.
(A. Nikfarjam, T. Stanford, A. Neumann, D. Dumuid, F. Neumann,
GECCO 2024 (Best Paper Award RWA track))

Motivation

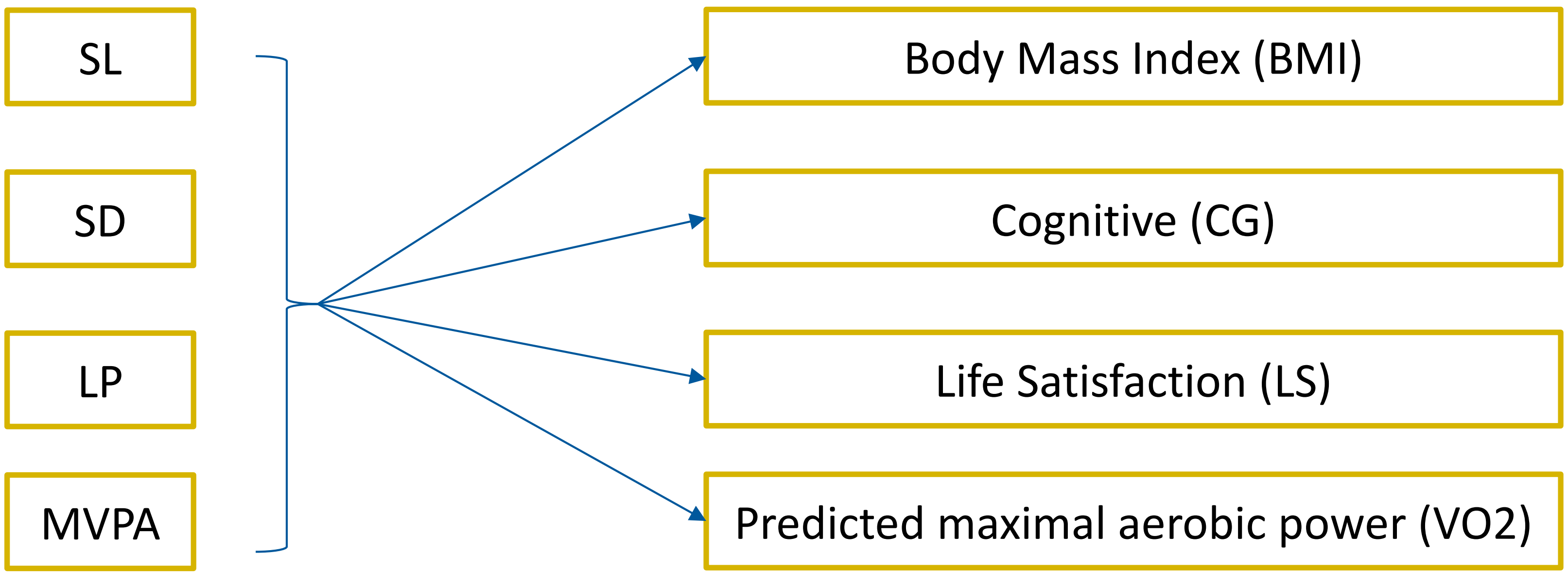
- Time is limited and how we spend our time has a strong impact on health and wellbeing.
- Daily activities such as sleeping, sitting, walking/running have a strong impact on health outcomes.
- Providing tools that allow people to structure their day to achieve health outcomes can be highly beneficial to improve the health of the population.
- We provide a tool that shows improvements in health outcomes based on daily activities (for children).

Technical Level:

- We provide quality diversity approaches to provide a picture on the impact of activities on health outcomes.
- We also show the interactions between different goals when allocating times for daily activities.

Activities and Health Outcomes

- Sleeping (SL)
- Sedentary Activities (SD)
- Light Physical Activities (LP)
- Moderate-to-vigorous Physical Activities (MVPA)



Activity Optimizer

Activity Optimiser

Select the objective function:

Academic

Sleep minutes:

480 610 710

480 510 540 570 600 630 660 690 710

sedentary minutes:

390 550 750

390 430 470 510 550 590 630 670 710 750

light physical activities:

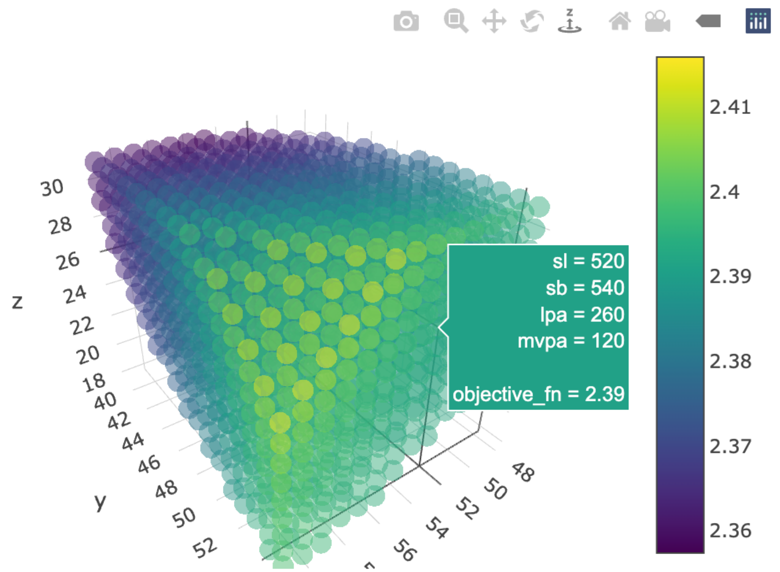
150 310 410

150 180 210 240 270 300 330 360 390

MVPA:

10 420 1,440

10 160 310 460 610 760 910 1,210 1,440

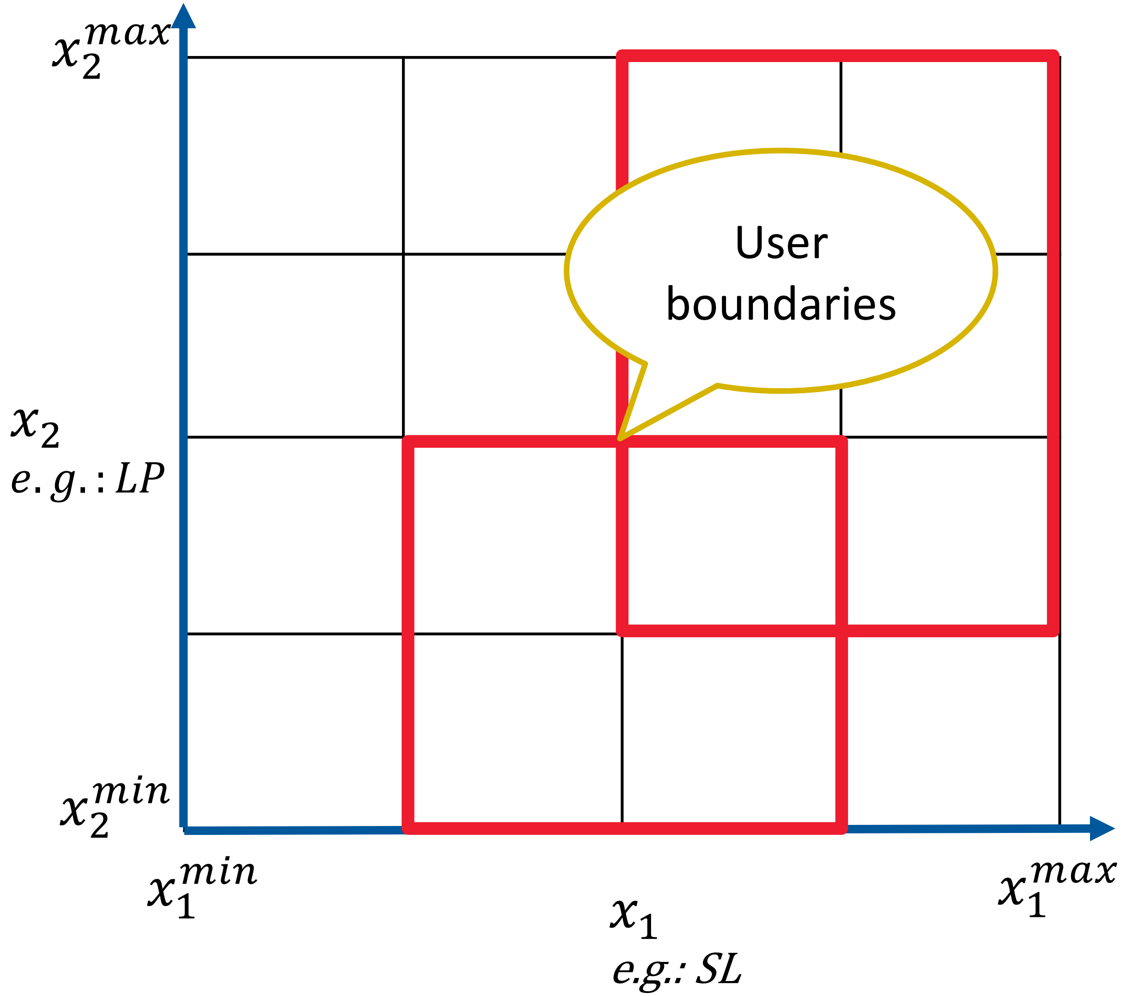



```
The optimal objective is:
2.41589173
The Sleep time:
610
The Sedentary time:
550
The La time:
260
The MVPA time:
20
```

https://arena2024.shinyapps.io/ActivityOptimiser/

Behavioral Space based on Decision Variables

- Setting up a behavioral space partitioned into cells.
- Find best-performing solution for each cell.
- User can set their boundaries.
- Report the best solution within the selected area.

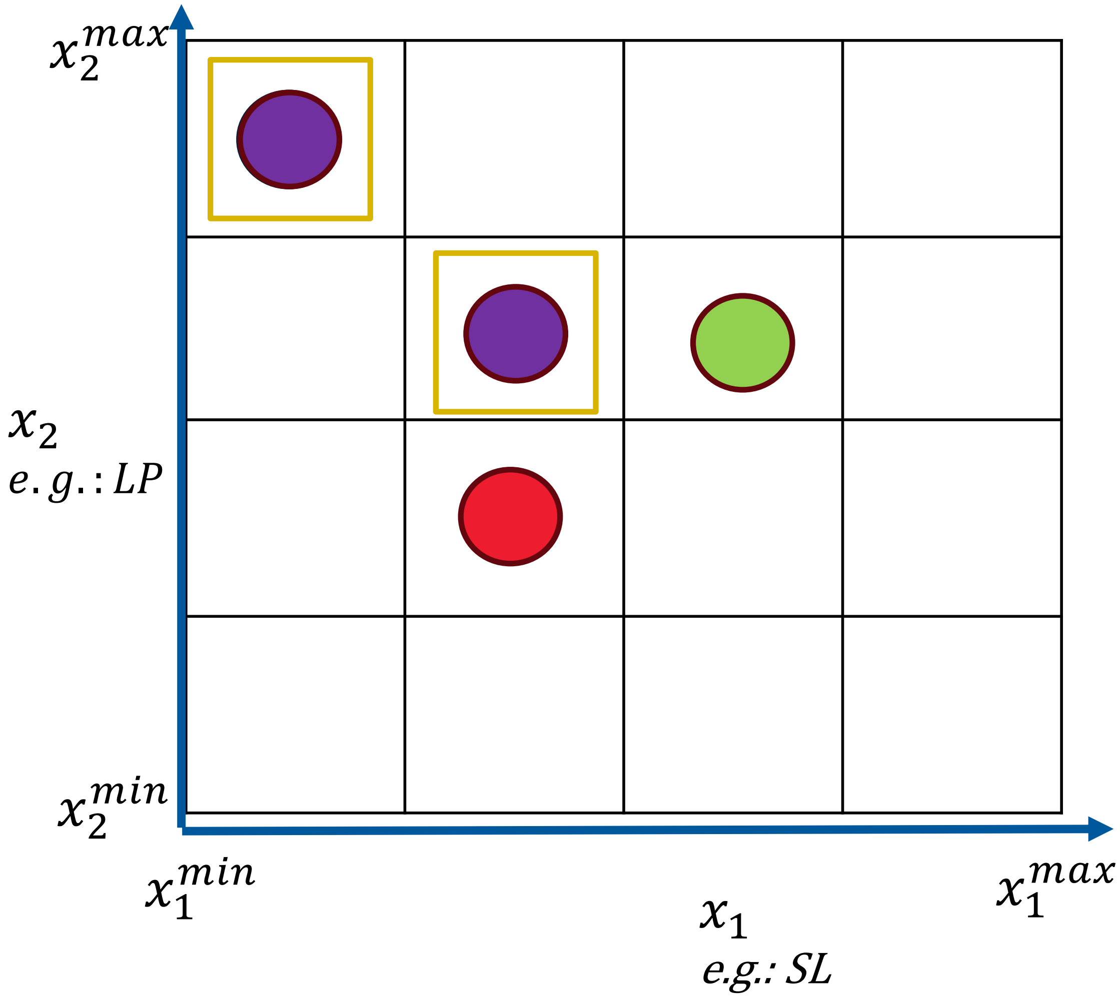


Map-Elites based EA

Algorithm 2 The EA algorithm

Require: An initial MAP.

while No termination criteria are met **do**
  Randomly select 2 solutions to serve as the parents.
  Generate an offspring x by the crossover.
  Apply the mutation operator on the offspring x.
  if x is feasible **then**
    Find the cell within the MAP where x belongs.
    if The cell is empty **then**
      Store the solution in the empty cell.
    else if x has a higher quality compared to the solution already occupying the cell **then**
      Replace the old solution with x.
    end if
  end if
end while



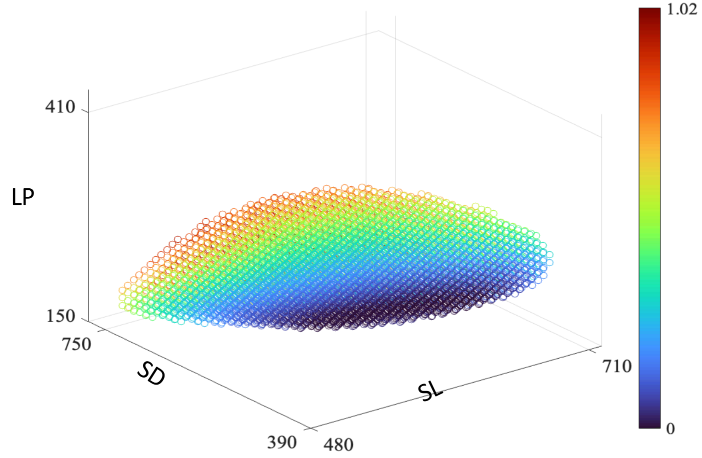
Generate a map with solutions obtained by EA
Repeat these steps:

Choose 2 solutions randomly and generate a new solution by crossover and mutation.

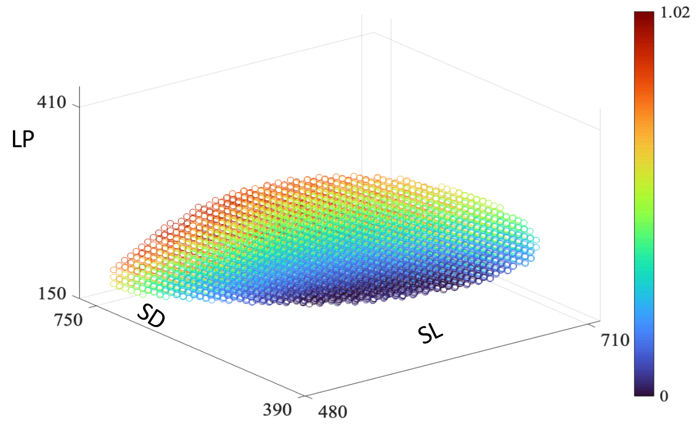
If it is occupied, keep the solution with higher quality.

Example Results for BMI

- We only show results for areas where we have enough data to do a reliable prediction.
- Figures shows distribution of solutions in the behavioral space according to activities sleep, sedentary, light and moderate-to-vigorous physical activities.
- The two figures are almost identical, showing the decent performance of the EA.



(a) BMI-brute force



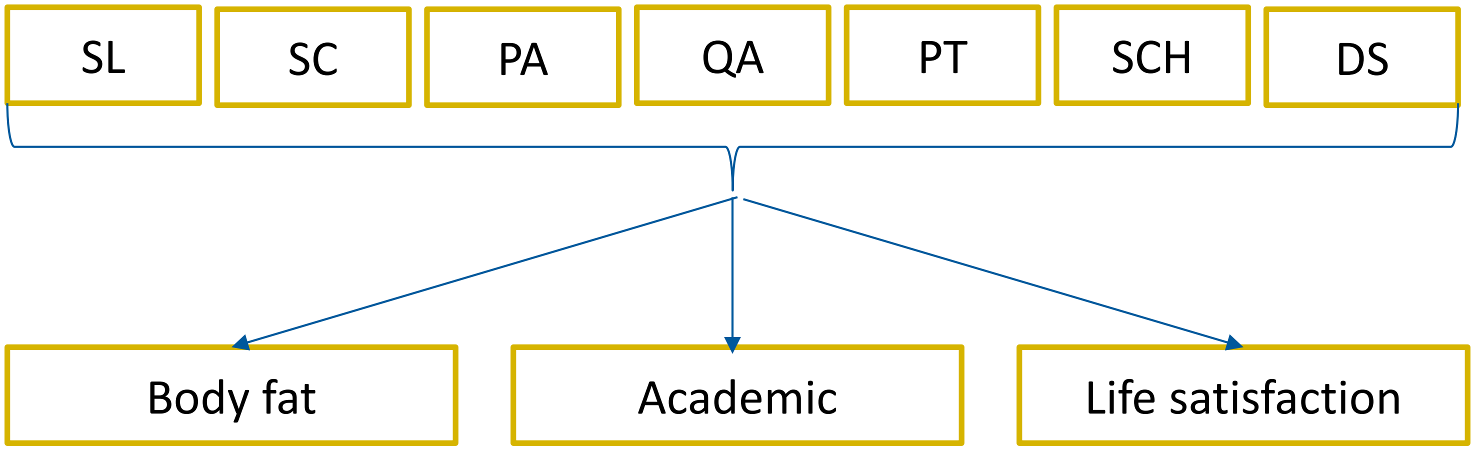
(e) BMI-EA

Higher dimensional Problems

- We have already seen that the EA results match the exact brute force results for 4-dimensional problems.
- We now consider problems with 7 input variables where we are not able to carry out the brute force approach.
- We want to show how the MAP elites approaches performs for theses problems.
- We show projections onto 2-dimensional subspace to display our results.

7D Problem

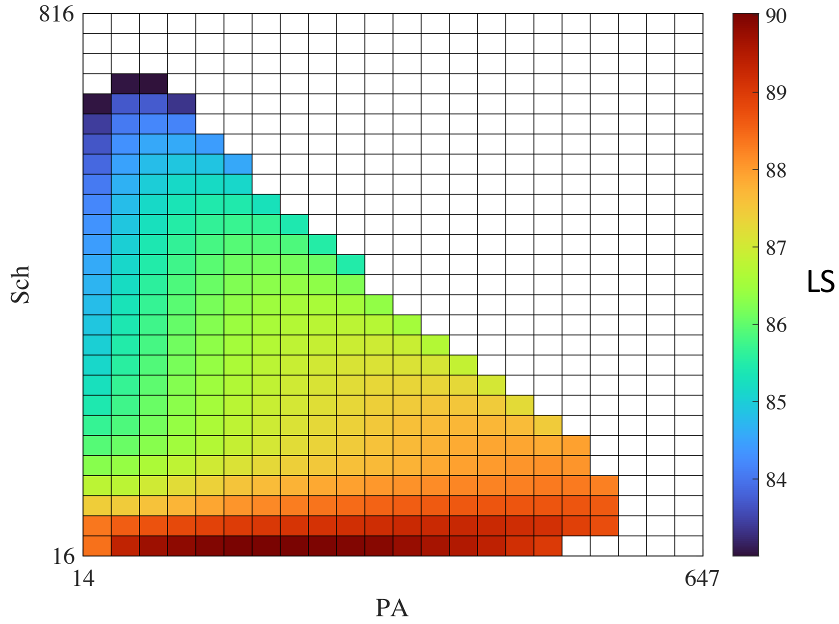
- Sleep (SL)
- Screen time (SC)
- Physical activities (PA)
- Quiet time (QA)
- Passive transport (PT)
- School-related (SCH)
- Domestic and self-care time (DS)



Example Results of Variable-based BS 7D (Life Satisfaction)

Distribution of high-quality LS solutions in projected ($PA \times SCH$) behavioral space.

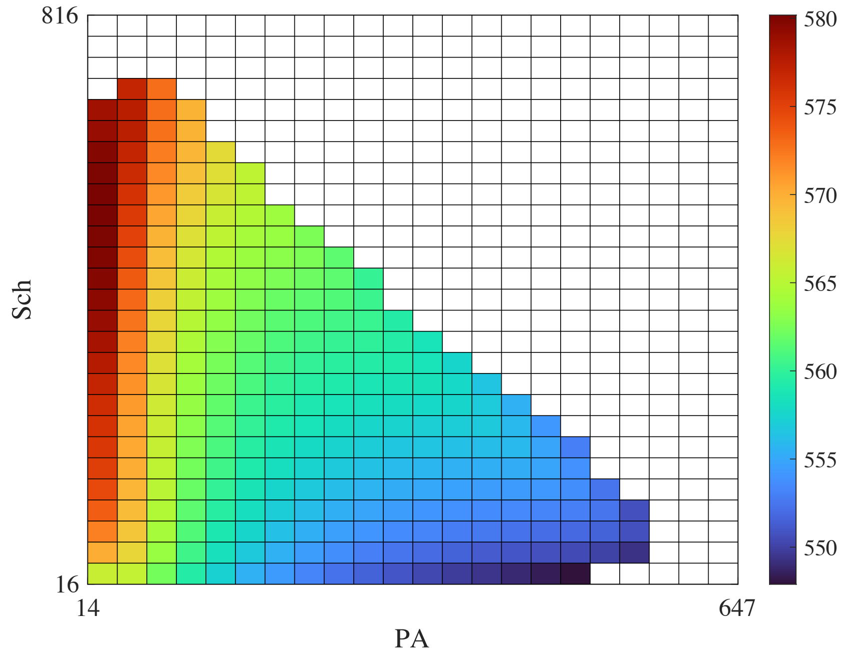
- The lower school time, the higher LS.
- The best LS solutions correspond to moderate allocations of PA.



Example Results of Variable-based BS 7D (Academic)

Distribution of high-quality solutions for academic performance in projected (PA×Sch) behavioral space.

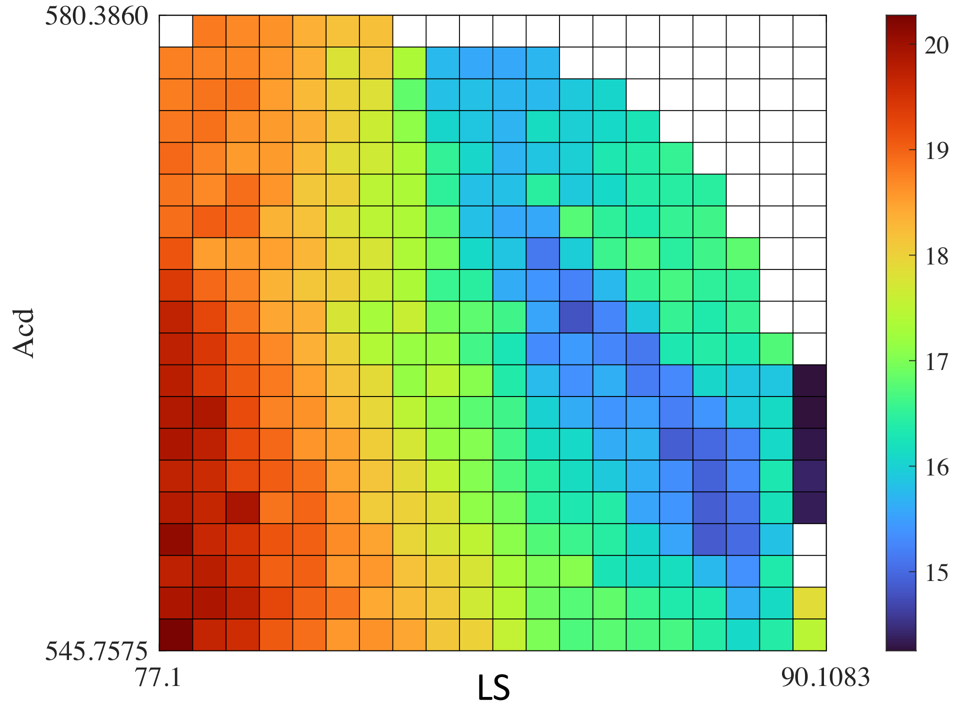
- Academic performance clearly increases with school time and reduces with the amount of physical activity.



Example Results of Objective-based BS 7D (Body Fat)

Distribution of minimal body fat solutions in projected (LS×Acd) behavioral space.

- Healthy body fat percentages (green) correspond to good life satisfaction health scores.
- High body fat scores correspond to low life satisfaction health scores (and low academic performance)



Conclusions

- Evolutionary diversity optimization and quality diversity approaches have gained increasing attention in evolutionary computation.
- They provide highly quality solutions with different structural and behavioral properties for a wide range of different problems.

Current interest:

- Theoretical understanding and analysis of these approaches.
- Applications of EDO and QD to combinatorial optimization problems and interesting real-world applications.

Thank you!