Quality Diversity

Generating a repertoire of good solutions

Jørgen Nordmoen, September 30, 2016

UNIVERSITY OF OSLO

Author: Jørgen Nordmoen, Subject: Evolutionary Robotics, Alternate title: Illumination Algorithms

It is important to note that Quality Diversity (QD) is more a new way of thinking searching in Evolutionary Robotics (ER) than it is a concrete algorithm. See section 2.

The "take home message" is that in ER we want to generate a diverse set of solution that are as good as they can be.

Introduction

Why do we need "Quality Diversity"?

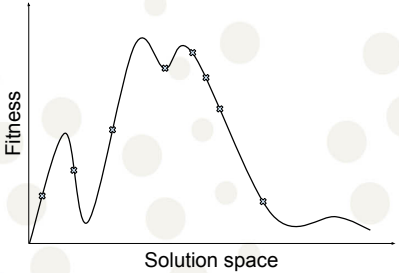
Introduction
Why do we need "Quality Diversity"?

Alternate subtitle: Motivating the need for diversity.
The introduction will motivate why we need "QD". The term will not be explained until later in the presentation, however, I hope that by the end of the introduction the name will become somewhat more self-explanatory, see section 2 for full description.

The basics of Evolutionary Algorithms (EAs)

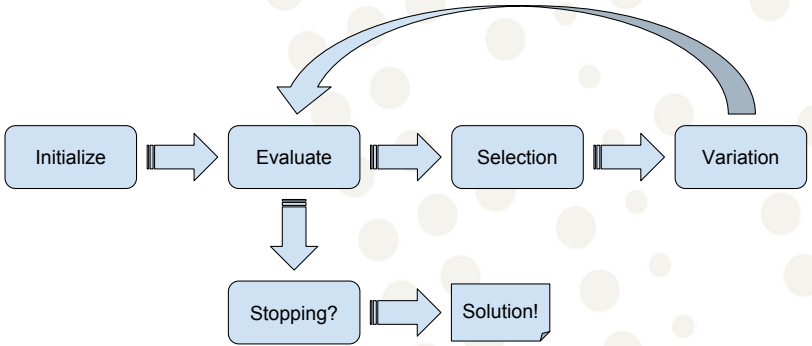
An EA consist of:

- Population of solutions
- Fitness function
- Mutation and recombination operators



The basics of Evolutionary Algorithms (EAs)
An EA consist of:
- Population of solutions
- Fitness function
- Mutation and recombination operators
Fitness
Solution space

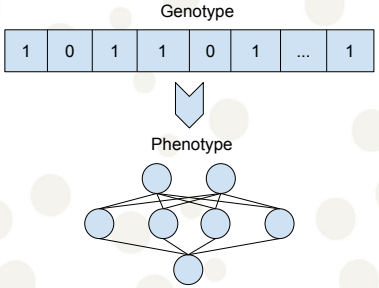
The presentation will start with a brief introduction to EAs.
Figure represents the fitness landscape of an imaginary problem. Along the X-axis we can see possible solutions and the Y-axis is the corresponding fitness, i.e. how good is the solution. In the figure the blue "X"s represent our population.



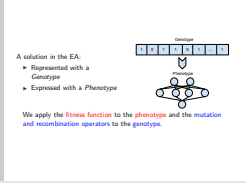
The EA cycle is inspired by natural evolution. We start by initializing a population, often randomly. We then move into a loop where we, evaluate all solutions using our fitness function, select the "best" candidates and create a new population utilizing our variation operators. We also check if a stopping criteria has been reached to determine when to break the cycle. When we break out of the loop the last generation hopefully contains the solution to the problem we started with.

A solution in the EA:

- Represented with a *Genotype*
- Expressed with a *Phenotype*

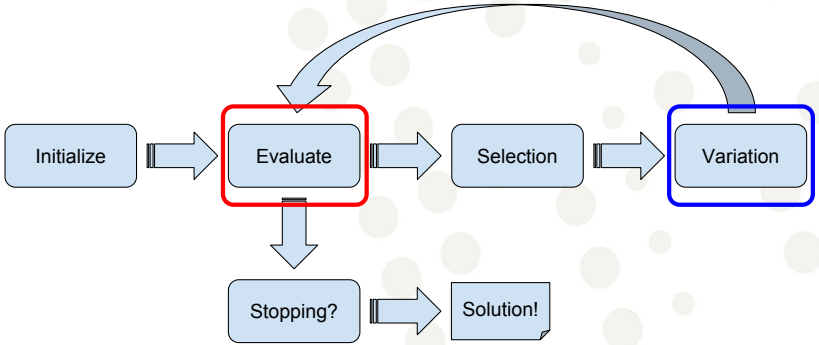


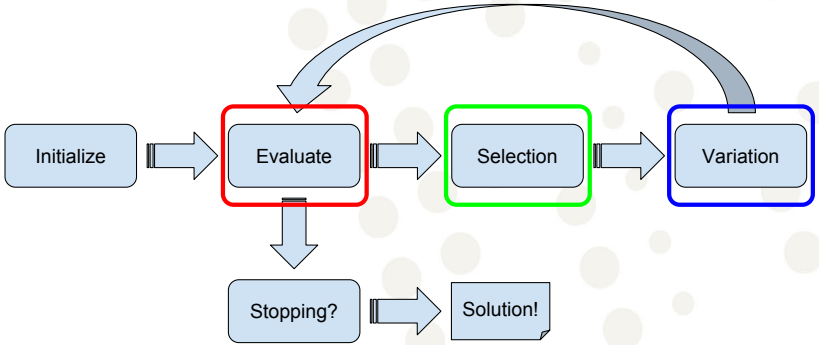
We apply the fitness function to the phenotype and the mutation and recombination operators to the genotype.



The genotype is often a compact representation where we can apply generic variation operators. The phenotype is then the expression of the genotype and is the "real" solution that we are after. We divide the solution into these two different representation to make it easier to apply variation operators during the search. E.g. if we want to create an antenna our phenotype is a 3D structure that hopefully functions as an antenna, but our genotype might be a tree structure of extrusion operations that together can build the structure.

It is important to note that we apply the fitness function to the phenotype and we apply the variation operators to the genotype. We will see how this arrangement can lead to some challenges later in the presentation.


Initialize
Evaluate
Selection
Variation
Stopping?
Solution!



If we look back to the EA cycle we can mark where the fitness evaluation and phenotype occurs, marked in red. And the variation operators and genotype, marked in blue.

In green we can see that selection is a separate step which selects the "best" solutions which is used as the basis for the next generation.

Some observations

- Convergent search (global optimization)
  - Selecting the "best" solutions
  - Homogenizing the "gene pool"

Some observations
- Convergent search (global optimization)
  - Selecting the "best" solutions
  - Homogenizing the "gene pool"

Some observations

- Convergent search (global optimization)
  - Selecting the "best" solutions
  - Homogenizing the "gene pool"
- Fitness and variation operates on different* representations
  - Evaluate phenotype
  - Change genotype

Some observations
- Convergent search (global optimization)
  - Selecting the "best" solutions
  - Homogenizing the "gene pool"
- Fitness and variation operates on different* representations
  - Evaluate phenotype
  - Change genotype

Some observations

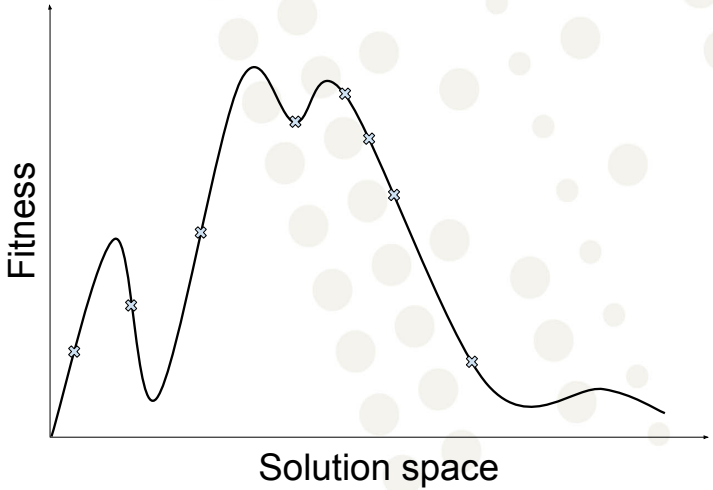
- Convergent search (global optimization)
  - Selecting the "best" solutions
  - Homogenizing the "gene pool"
- Fitness and variation operates on different* representations
  - Evaluate phenotype
  - Change genotype
- Can't separate solutions with the same fitness
  - Selection does not differentiate how the fitness is expressed

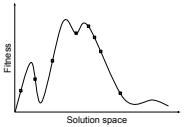
Some observations
- Convergent search (global optimization)
  - Selecting the "best" solutions
  - Homogenizing the "gene pool"
- Fitness and variation operates on different* representations
  - Evaluate phenotype
  - Change genotype
- Can't separate solutions with the same fitness
  - Selection does not differentiate how the fitness is expressed

The standard EA is a convergent search, meaning it will try to converge on the best solution. The best solution is decided by the fitness function and the selection process always* tries to choose the better solutions for the next generation, this is called "selection pressure". In addition to the selection pressure we have the variation operators that will try to modify the selected solutions so that they become better. There are many ways to do this, but many variation operators can also lead to a more homogenized population.

As noted earlier, fitness and variation works on different representations of the problem. This is important because a change in genome does not necessarily mean a change in either phenotype or fitness.

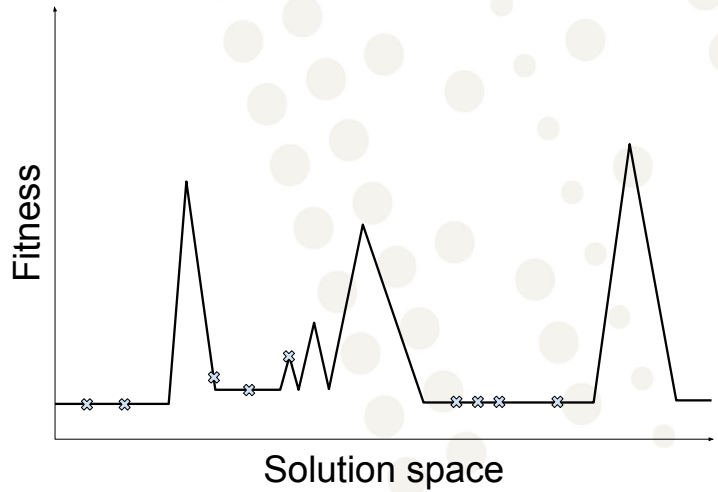
The final thing to note is that solutions with the same fitness is not distinguished further by an EA. If such situations occur during an EA run it is not guaranteed that either or both solutions are kept. Even though the might represent two different strategies for solving the problem.

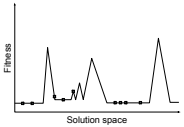




To illustrate the previous slides further we can go back to the fitness landscape presented earlier. If we look at the second highest peak we can see that it currently has the best solution. So ideally we would like all solution to move closer in order for one of the solutions on the left to climb the tallest peak and find the global optimum. However, if the problem is difficult enough what is more likely to occur is that all solutions **converge** on the second tallest peak and we are stuck in a local optimum.

In the next slide we will see how deception in the fitness landscape affects the search, but it is worth noting that the genotype, phenotype disconnect could be visible here too. In this figure the disconnect would mean that small changes to a solution might move it away from the peaks.

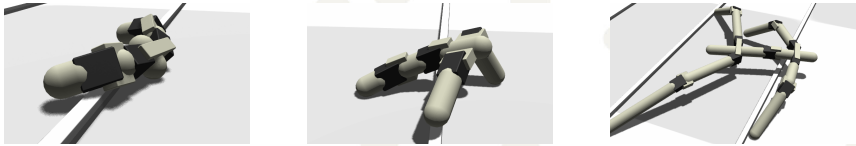




This fitness landscape tries to illustrate a "deceptive" problem. What this means is that the problem is difficult and the fitness landscape contains mainly steep peaks that are difficult to randomly discover. As we saw in the previous slide, in a normal EA we would like our problems to be of such a nature that better solutions should be focused on. However, if the problem is deceptive it is unlikely that the current good solutions are close to the global optimum and to discover the global peaks one might need several generations of mutations. In the best case these mutations does not change the fitness and this can randomly happen, but what often end up happening is that the mutation lowers the fitness and so the probability of several mutations nearly never happens. The diversity of the population will therefore converge and so it is unlikely that we find the global optimum.

In previous slides it was postulated that EAs have taken inspiration from natural evolution, yet natural evolution is not a global optimizer. In nature most animals occupy a niche and does not compete on a global scale. This allows for a diverse set of expression of the same basic concepts. This ideal is interesting because it allows us as problem solvers to select which expression we like the most, not necessarily because the chosen expression is the best, but because it is interesting in it own way.

An additional benefit of diversity might be that we can create some interesting combinations.



From Samuelsen and Glette[1]

From Samuelsen and Glette[1]

To illustrate this point further we can look at these three different expressions of morphology from Eivind Samuelsen. Again, if we simply optimize for speed we would chose the faster of the three, yet each expression is a novel and interesting way to solve the problem of how to create a robot.

Background

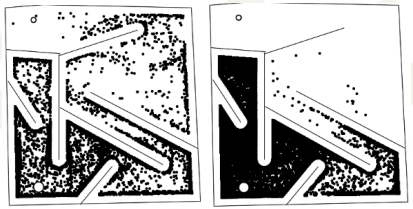
What is "Quality Diversity"?

Background
What is "Quality Diversity"?

The following slides will describe the history behind QD and what we mean by QD.
We will begin with a brief history of different algorithms and then discuss the intention behind QD.
It is important to stress that QD is more a way to think about ER than a concrete implementation.

Novelty Search[2]

- Introduction of divergent search
- Deceptive problems are hard for fitness based EA
- Searching for "interesting" solutions better than "good" solutions
- Archive of previous solutions

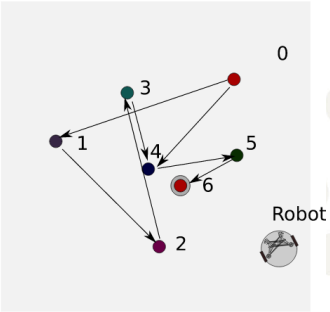


Novelty Search[2]
- Introduction of divergent search
- Deceptive problems are hard for fitness based EA
- Searching for "interesting" solutions better than "good" solutions
- Archive of previous solutions

QD started with Novelty search. Novelty search introduced the notion that fitness might not be the best guide for an EA. Their results showed that it is easy for fitness to get stuck in local minima, but using novelty can solve the problem.
Novelty search was implemented by rewarding solutions that end up at a different place than other solutions. This pressures the search to explore new locations which can avoid local minima.
The images on the slide show the "hard" maze with novelty on the left and fitness on the right. The agents are differential drive robots that start in the large circle and want to find the small circle. The search stopped either when a solution has found the goal or when enough iterations where performed. From the graphics we can see that novelty search manages to find the goal and is much more spread out. Fitness is not able to find the goal and is mostly trapped in a local minima.

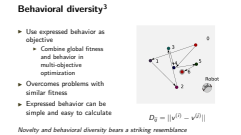
Behavioral diversity[3]

- Use expressed behavior as objective
  - Combine global fitness and behavior in multi-objective optimization
- Overcomes problems with similar fitness
- Expressed behavior can be simple and easy to calculate



$$D_{ij} = ||v^{(i)} - v^{(j)}||$$

Novelty and behavioral diversity bears a striking resemblance



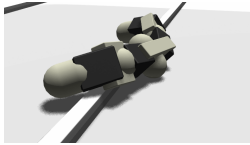
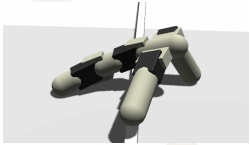
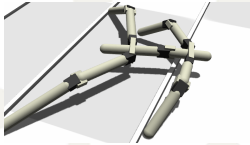
This next paper is not exactly QD, but it introduced a key aspect of later QD algorithms, behavior characteristics. Behavior characteristics is a description of how the behavior of the solution and not simply it fitness in solving the given task.
In the paper the authors combined fitness and behavioral characteristics in a multi-objective algorithm, NSGAII, to solve a difficult navigation task. The task was for a small differential drive robot to turn on lights in a given order, as depicted on the image.
Their results showed that by adding the behavior characteristics the EA was able to overcome problems with equal fitness to eventually solve the problem. They also showed that behavior characteristics can be easy to calculate and for their experiments simply used a vector of booleans to indicate which switches the robot had activated.
Fitness function was number of steps to activate the last light.

Novelty Search with Local Competition[4]

- Diversity on its own is not enough
- Global quality is still convergent
- Let niches evolve and have competition only inside niches!
  - Local competition allows each niche to become proficient
  - Novelty measured against whole population and archive

Novelty Search with Local Competition[4]
- Diversity on its own is not enough
- Global quality is still convergent
- Let niches evolve and have competition only inside niches!
  - Local competition allows each niche to become proficient
  - Novelty measured against whole population and archive

Novelty Search with Local Competition[4]

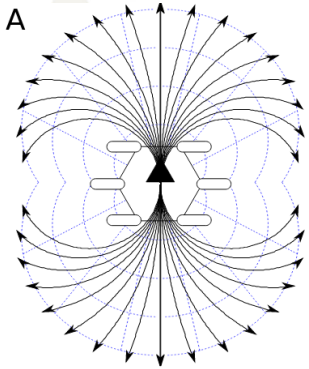
Novelty Search with Local Competition[4]

The next piece of the puzzle was Novelty Search with Local Competition. Novelty search before showed that searching for fitness alone is not enough, yet we still need some notion of quality. With the introduction of local competition the authors kept the best part about novelty search, that the search is allowed to diverge, but they also introduced fitness by having similarly expressed solution compete with each other. This algorithm differs from previous by allowing solutions to only compete with similar solutions, but diverge. Their inspiration came from niches in the real-world. If a newly created solution is so different from previous solutions then it should not compete because it has created its own niche to occupy.

The solution was created by using NSGAII with local fitness competition and novelty. In the paper a comparison is also made to global competition and novelty. Global competition ultimately crates better solutions, but it does not explore as much of the search space as with local competition.

Behavioral Repertoire Learning in Robotics[5]

- Treat the novelty archive as a repertoire
- New solutions that are not novel, but better replaces older solutions
- Select desired behavior from repertoire as needed



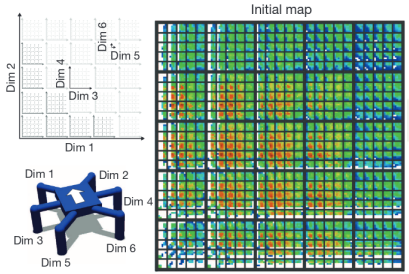
Behavioral Repertoire Learning in Robotics[5]
- Treat the novelty archive as a repertoire
- New solutions that are not novel, but better replaces older solutions
- Select desired behavior from repertoire as needed

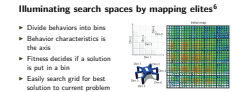
Another QD algorithm is behavioral repertoire learning. This approach, much like Novelty Search uses an archive to store previous seen solutions. However, by using fitness to replace the solutions the algorithm not only have a final generation, but also an archive of good solutions. The authors showed that by creating this repertoire they were able to select solutions that fit the current problem description.

The key insight is that one should accept solutions if it is able to solve the problem in a novel way or is strictly better than something that has been seen before. By replacing solutions in the archive the authors were able to take advantage of the initial random babbling that might produce solutions going in the "wrong" direction, but can be stepping stones for future better solutions.

Illuminating search spaces by mapping elites[6]

- Divide behaviors into bins
- Behavior characteristics is the axis
- Fitness decides if a solution is put in a bin
- Easily search grid for best solution to current problem





The final contribution to QD is the MAP-Elites algorithm. Conceptually simple the algorithm divides the behavior space into discrete bins and tries to find the best solution in each bin. Much like behavioral repertoire learning it generates a final "map" of solutions and like novelty search and local competition it only compares fitness with solutions that are similar in behavior.

Summary

- Divergent search
- Quality among "similar" expressions
- "Behavior" is the important metric

"... the goal of this new type of search, called *quality diversity* (QD), is to find a maximally diverse collection of individuals (..) in which each member is as high performing as possible."[7]

Summary
- Divergent search
- Quality among "similar" expressions
- "Behavior" is the important metric

"... the goal of this new type of search, called *quality diversity* (QD), is to find a maximally diverse collection of individuals (..) in which each member is as high performing as possible."[7]

Current research

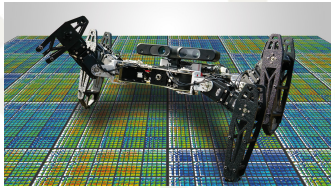
Current research and challenges with Quality Diversity

Current research
Current research and challenges with Quality Diversity

The following slides will illustrate the challenges with QD and what is being done to address these issues.

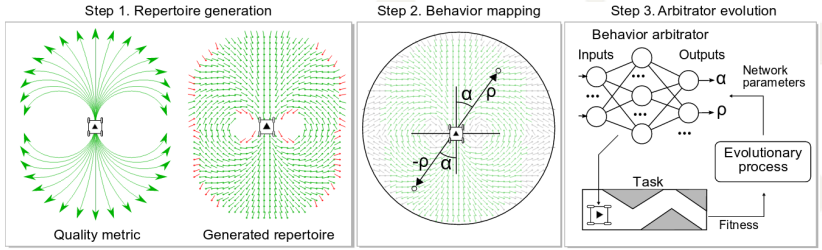
Examples of real-world experiments

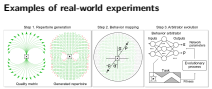
- Robots that can adapt like animals[8]
- Evolving a Behavioral Repertoire for a Walking Robot[9]
- EvoRBC: Evolutionary Repertoire-based Control for Robots with Arbitrary Locomotion Complexity[10]



Examples of real-world experiments
- Robots that can adapt like animals[8]
- Evolving a Behavioral Repertoire for a Walking Robot[9]
- EvoRBC: Evolutionary Repertoire-based Control for Robots with Arbitrary Locomotion Complexity[10]

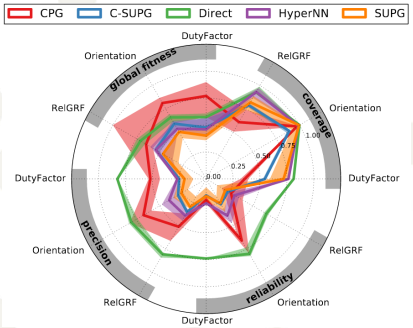
Examples of real-world experiments





Encoding matters[11]

- Direct better than generative encoding (?)
- Generative encoding usually changes significantly
- Small influence on real-world tests!



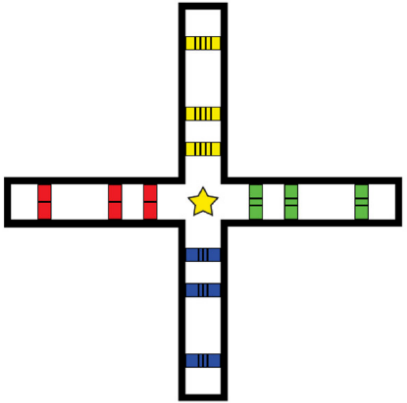
Alignment of behavioral characteristics (BC)

- Alignment is how well the BC is matched with fitness
- Pugh et al.[7] showed that
  - Choice of alignment is important in deceptive problems
  - Using multiple alignments can be used, but might not generate better results
- Auerbach et al.[12] showed that
  - BC drives the search in different directions
  - Novelty Search and MAP-Elites needs different types of alignment

Alignment of behavioral characteristics (BC)
- Alignment is how well the BC is matched with fitness
- Pugh et al.[7] showed that
  - Choice of alignment is important in deceptive problems
  - Using multiple alignments can be used, but might not generate better results
- Auerbach et al.[12] showed that
  - BC drives the search in different directions
  - Novelty Search and MAP-Elites needs different types of alignment

Evolving generalists[13]

- Quality Diversity algorithms creates specialists
- Need to encourage each solution to acquire new skills



Evolving generalists[13]
- Quality Diversity algorithms creates specialists
- Need to encourage each solution to acquire new skills

Open questions

- Isn't this just brute-force?
- How is this different from multi-objective optimization?
- Isn't this just moving all knowledge into behavior characteristics?
- Is Quality Diversity just a result that we can throw more compute power at the problem?
  - Can't we solve the same problem with regular EAs with larger populations?
  - Aren't the problems just so artificial that "Novelty" must win?

Open questions

- Isn't this just brute-force?
- How is this different from multi-objective optimization?
- Isn't this just moving all knowledge into behavior characteristics?
- Is Quality Diversity just a result that we can throw more compute power at the problem?
  - Can't we solve the same problem with regular EAs with larger populations?
  - Aren't the problems just so artificial that "Novelty" must win?

"Take home message"

- Quality Diversity is new way to think about searching
  - Diversity can create stepping stones to future solutions
  - Quality is important to direct the search

"Take home message"
- Quality Diversity is new way to think about searching
  - Diversity can create stepping stones to future solutions
  - Quality is important to direct the search

Bibliography

[1] Eivind Samuelsen and Kyrre Glette. Some distance measures for morphological diversification in generative evolutionary robotics. In *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, pages 721–728. ACM, 2014.

[2] Joel Lehman and Kenneth O. Stanley. Exploiting open-endedness to solve problems through the search for novelty. In *ALIFE*, pages 329–336, 2008.

[3] Jean-Baptiste Mouret and Stéphane Doncieux. Overcoming the bootstrap problem in evolutionary robotics using behavioral diversity. In *2009 IEEE Congress on Evolutionary Computation*, pages 1161–1168. IEEE, 2009.

[4] Joel Lehman and Kenneth O. Stanley. Evolving a diversity of virtual creatures through novelty search and local competition. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 211–218. ACM, 2011.

[5] Antoine Cully and Jean-Baptiste Mouret. Behavioral repertoire learning in robotics. In *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, pages 175–182. ACM, 2013.

[6] Jean-Baptiste Mouret and Jeff Clune. Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909*, 2015.

[7] Justin K. Pugh, Lisa B. Soros, and Kenneth O. Stanley. Quality diversity: A new frontier for evolutionary computation. *Frontiers in Robotics and AI*, 3:40, 2016.

[8] Antoine Cully, Jeff Clune, Danesh Tarapore, and Jean-Baptiste Mouret. Robots that can adapt like animals. *Nature*, 521(7553):503–507, 2015.

[9] Antoine Cully and J-B Mouret. Evolving a behavioral repertoire for a walking robot. *Evolutionary computation*, 24(1):59–88, 2016.

[10] Miguel Duarte, Jorge Gomes, Sancho Moura Oliveira, and Anders Lyhne Christensen. Evorbc: Evolutionary repertoire-based control for robots with arbitrary locomotion complexity. In *Proceedings of the 18th annual conference on Genetic and evolutionary computation*. ACM, 2016.

[11] Danesh Tarapore, Jeff Clune, Antoine Cully, and Jean-Baptiste Mouret. How do different encodings influence the performance of the MAP-Elites algorithm? In *Genetic and Evolutionary Computation Conference*, 2016.

[12] Joshua E. Auerbach, Giovanni Iacca, and Dario Floreano. Gaining insight into quality diversity. In *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion*, pages 1061–1064. ACM, 2016.

[13] Christopher Stanton and Jeff Clune. Curiosity search: Producing generalists by encouraging individuals to continually explore and acquire skills throughout their lifetime. *PloS one*, 11(9):e0162235, 2016.

Bibliography