codigital

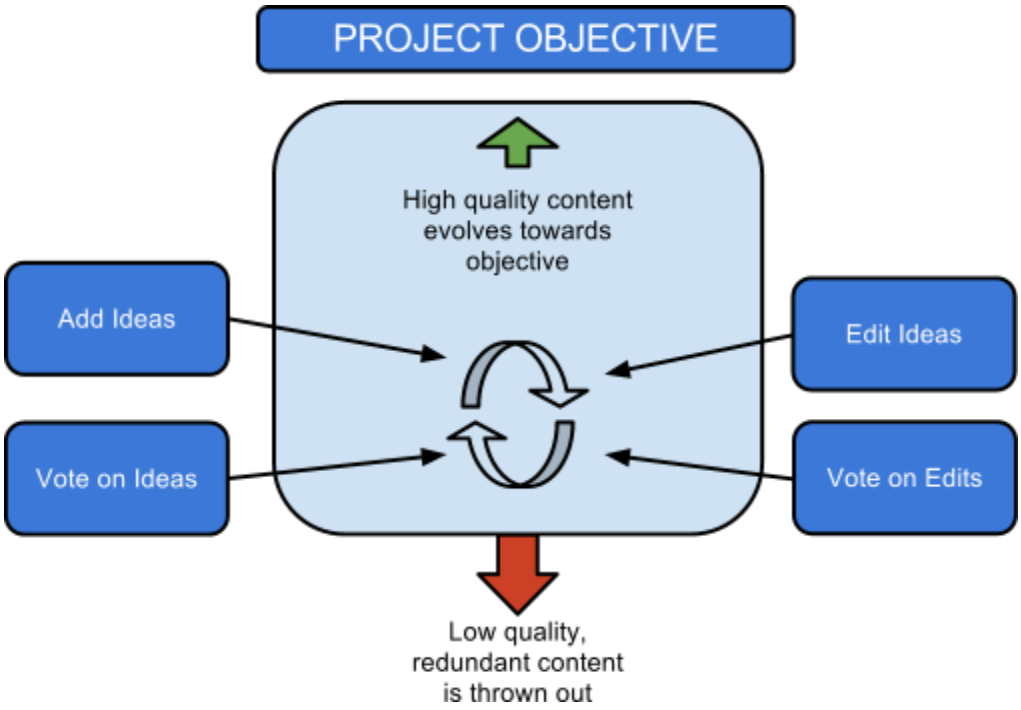
Codigital is a system for large scale collaboration, allowing many views to be evolved and fused into concise collaborative output in real-time. The system enables the entire group's creative ideas to emerge, and opinions to be revealed, in a dynamic, democratic, online process. Collaborators continually process each other's points of view - ideas flow, cross-fertilize and evolve.

- Quality, representative output
- Easy to use
- Fast
- Fun
- Self moderating

The Process

Each project has a specific objective (e.g. **How can we combat global warming?**). All participation in the project is with respect to this common goal.

4 key activities drive the process:
1. Add Ideas; **2. Vote on Ideas**; **3. Edit Ideas**; **4. Vote on Edits**



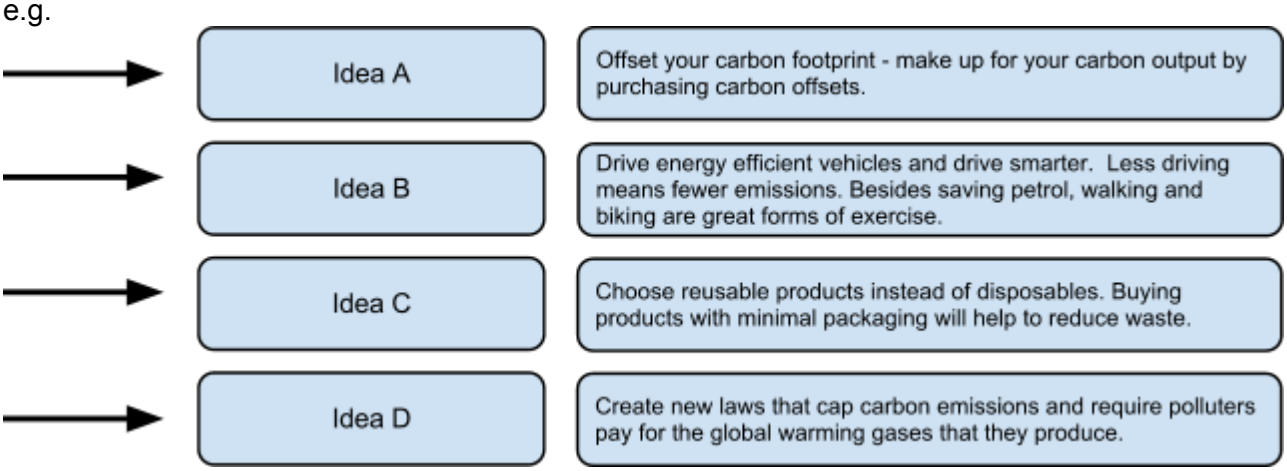
The system:

- Receives new ideas
- Promotes good ideas
- Throws out bad ideas
- Receives new edit suggestions
- Incorporates good edit suggestions
- Throws out bad edit suggestions
- Fuses similar ideas and suggestions
- Evolves thousands of ideas into a manageable set
- Tracks user participation in terms of (i) contribution of content (ii) contribution to the decision making process

Add Ideas

Users add ideas into the system which relate to the project objective.

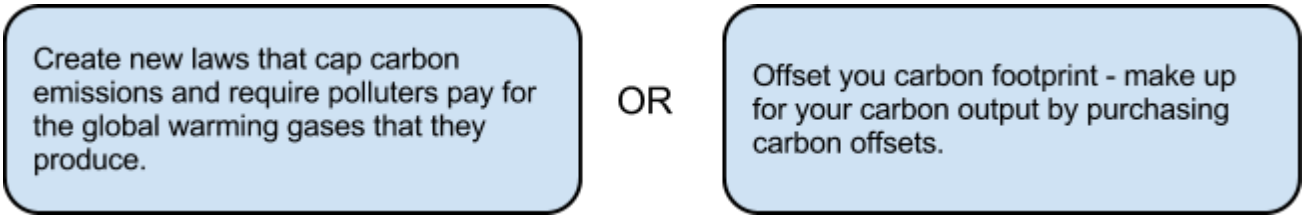
e.g.



Vote on Ideas

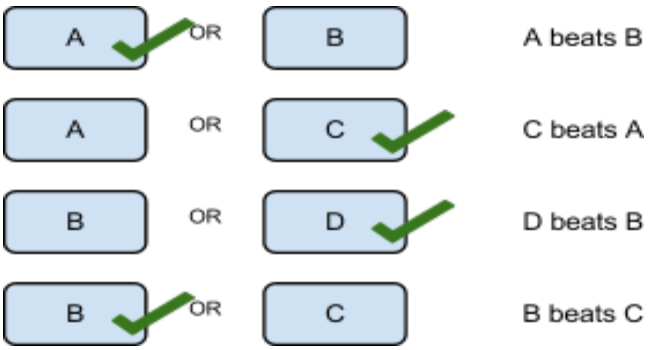
Users are presented with pairs of Ideas to assess with respect to the project objective.

Which of these ideas best fit the project objective?
How do we combat global warming?

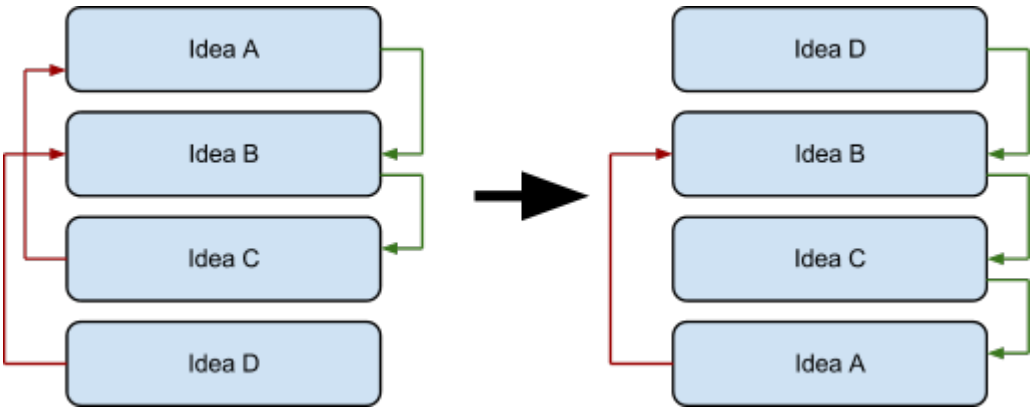


The users click on the idea that, in their opinion, best fits the objective.

The system samples pair choices for each user, ensuring that all ideas are given a proportionate number of votes across a significant number of users.

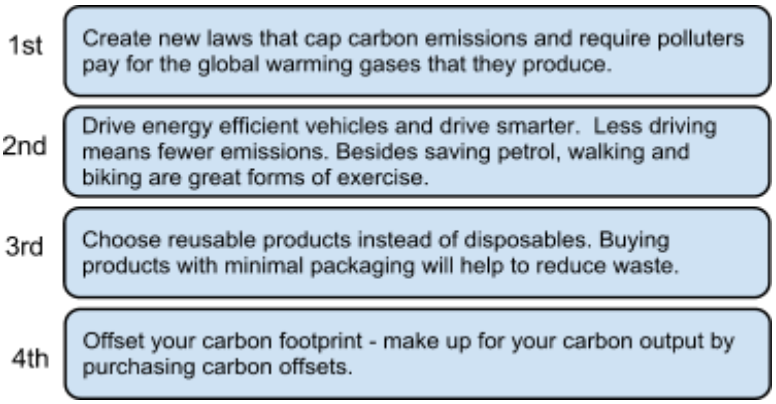


The vote results are collated and processed to generate a new ranking for the ideas.



The ranking algorithm optimizes the ranking with respect to all the votes cast.

In this example, the resulting order might be:

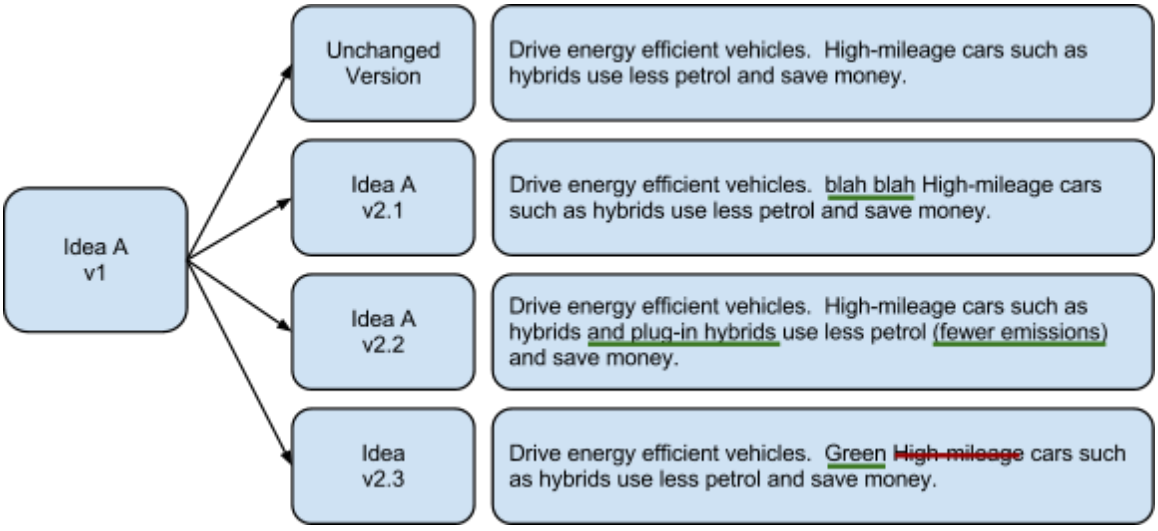


Edit Ideas

Users edit each other's ideas.

All suggested edits are 'pending' in the next generation of the content until one of the competing versions wins and is promoted to be the live version.

In the example below, we see that Idea A has had 3 edits submitted. These 3 versions, and an unchanged copy of the original, constitute the possible versions for the next generation of the idea.



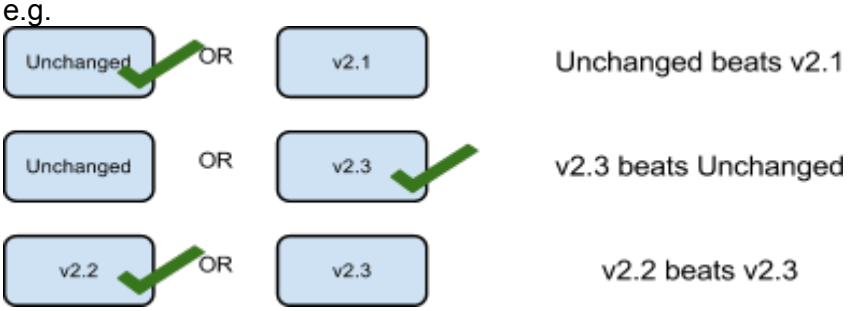
The size of the edit is restricted to 30% as measured by the difference algorithm. This is to ensure that ideas develop progressively.

Vote on Edits

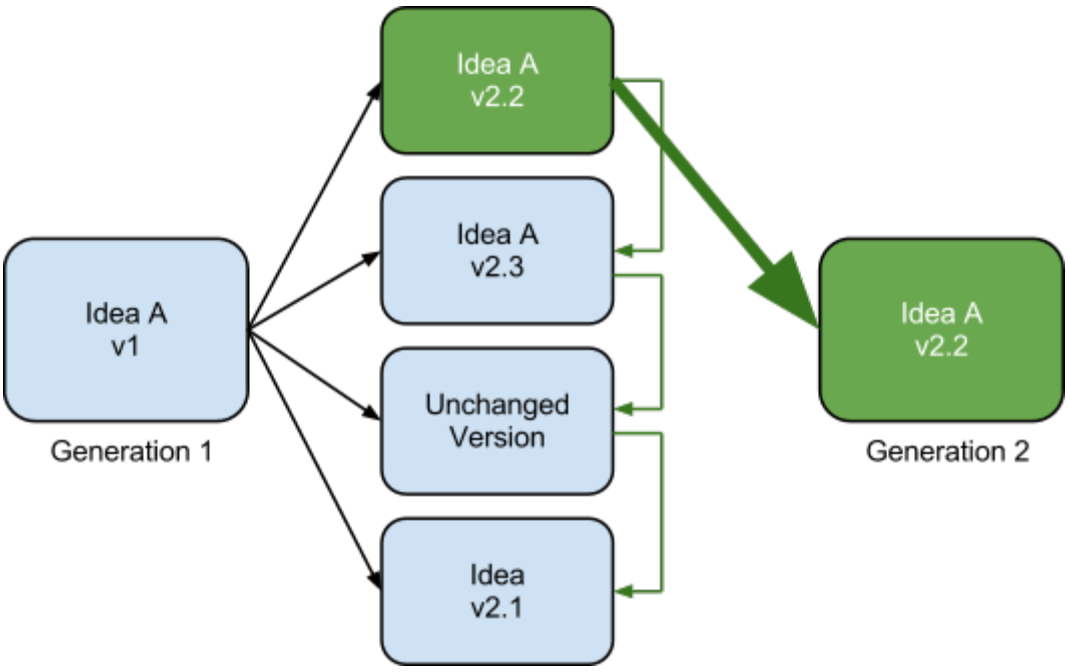
Users are presented with pairs of versions of Ideas to assess with respect to the project objective.

The users click on the version that, in their opinion, best fits the objective.

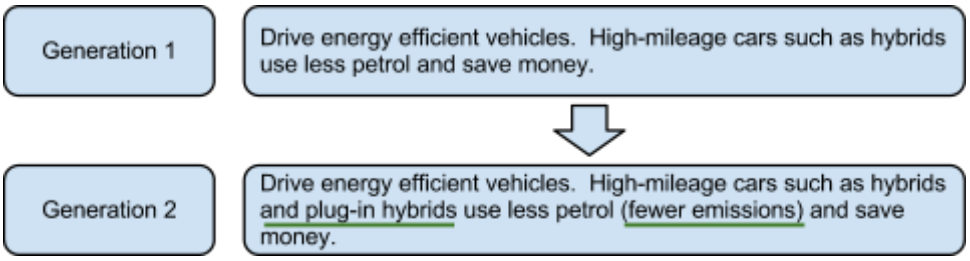
e.g.



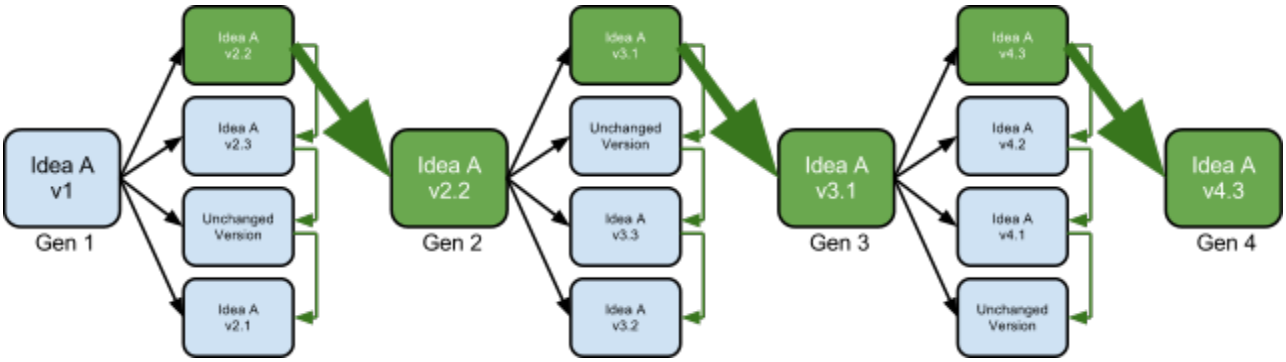
As the votes come in, the versions are ranked accordingly. Once certain vote thresholds are reached, a winner is chosen and is promoted to be the new live version.

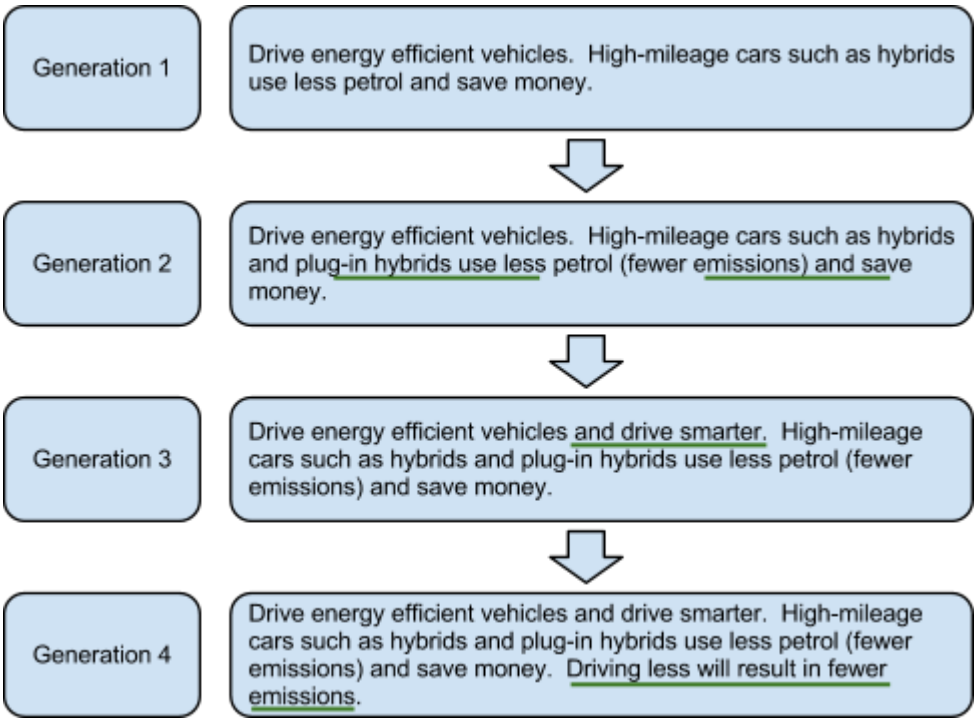


Here we see that the suggested version (v2.2) has won and has been promoted as the new live version for Idea A in generation 2. The other edits are discarded.



This process is repeated, allowing the content to evolve smoothly over time.





Putting it all together

Voting on Ideas drives the continual ranking of ideas against each other

Voting on Edits incorporates the best edit suggestions into existing ideas, driving evolution

Germs of good ideas are developed incrementally to become more competitive - many collaborators claim a stake in a well developed idea

As ideas evolve they are reassessed against new and existing ideas, - only the very best ideas survive

The Result

- **Concise, evolved content** that is representative of the group's views
- **Distributed ownership** - the group have a well specified shareholding in the result
- **Analytics** - reports on the content evolution and user participation

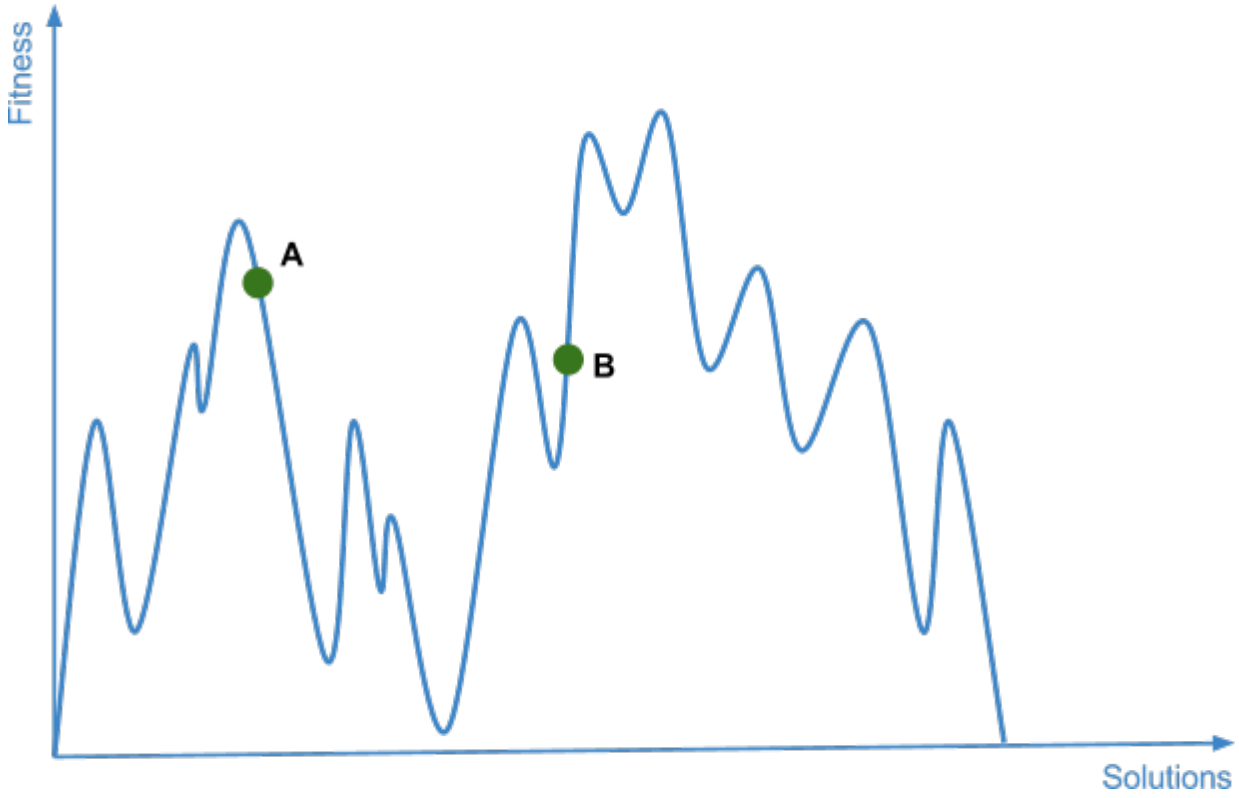
APPENDIX

The process embodies an evolutionary process

- Exploits the power of cumulative selection
- Constantly improving content
- Optimizes the search through the fitness landscape

Fitness Landscape

We assume that every possible idea or proposed solution to this problem, however composed, falls somewhere on the fitness landscape as illustrated below:



In this illustration, solution A is better than solution B.

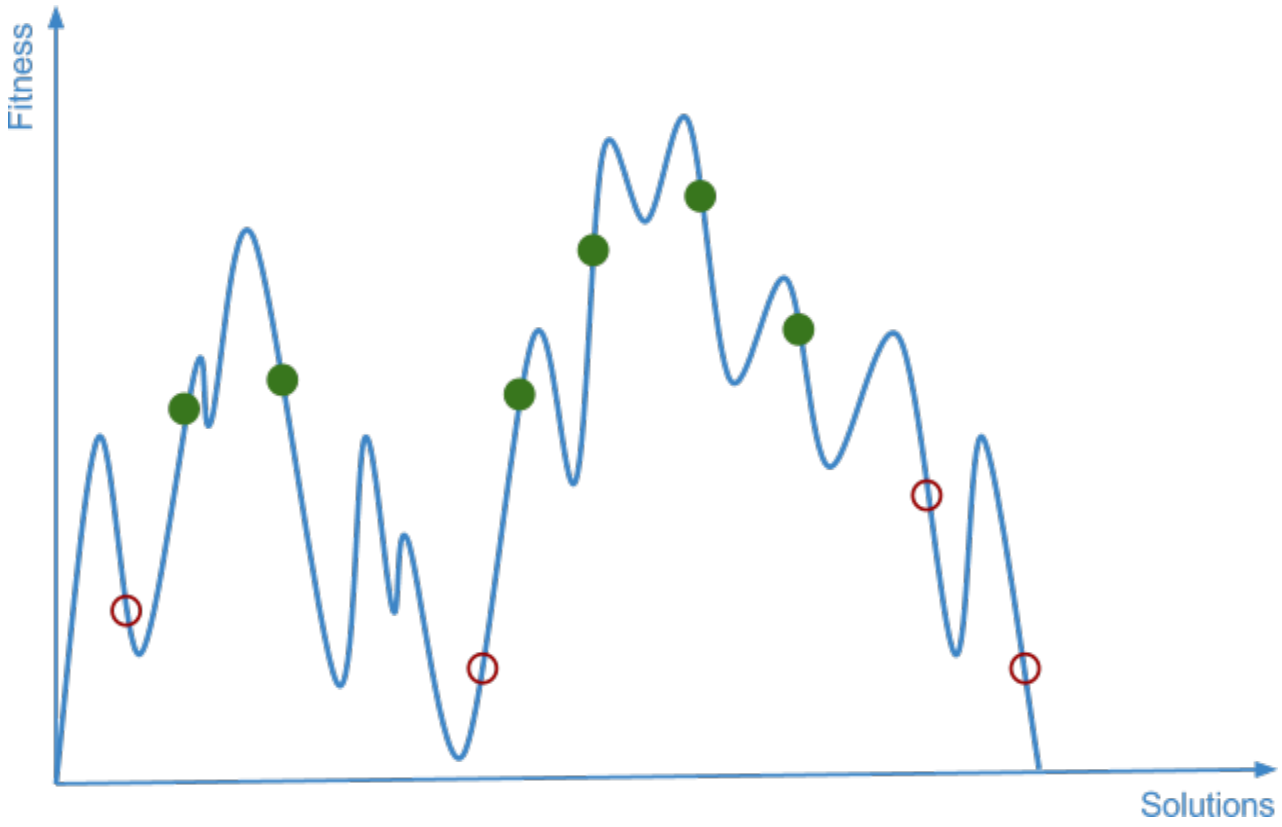
The peaks represent local maxima, and can be thought of as the optimal composition of that particular solution.

In this case, though A is better than B, there are solutions near B that are significantly better than A. You could think of A being a well formed bad idea, and B being a badly formed good

idea.

Submit and rank

By allowing participants to submit many ideas and to rank each other's ideas, we start to identify the better solutions on the fitness landscape:



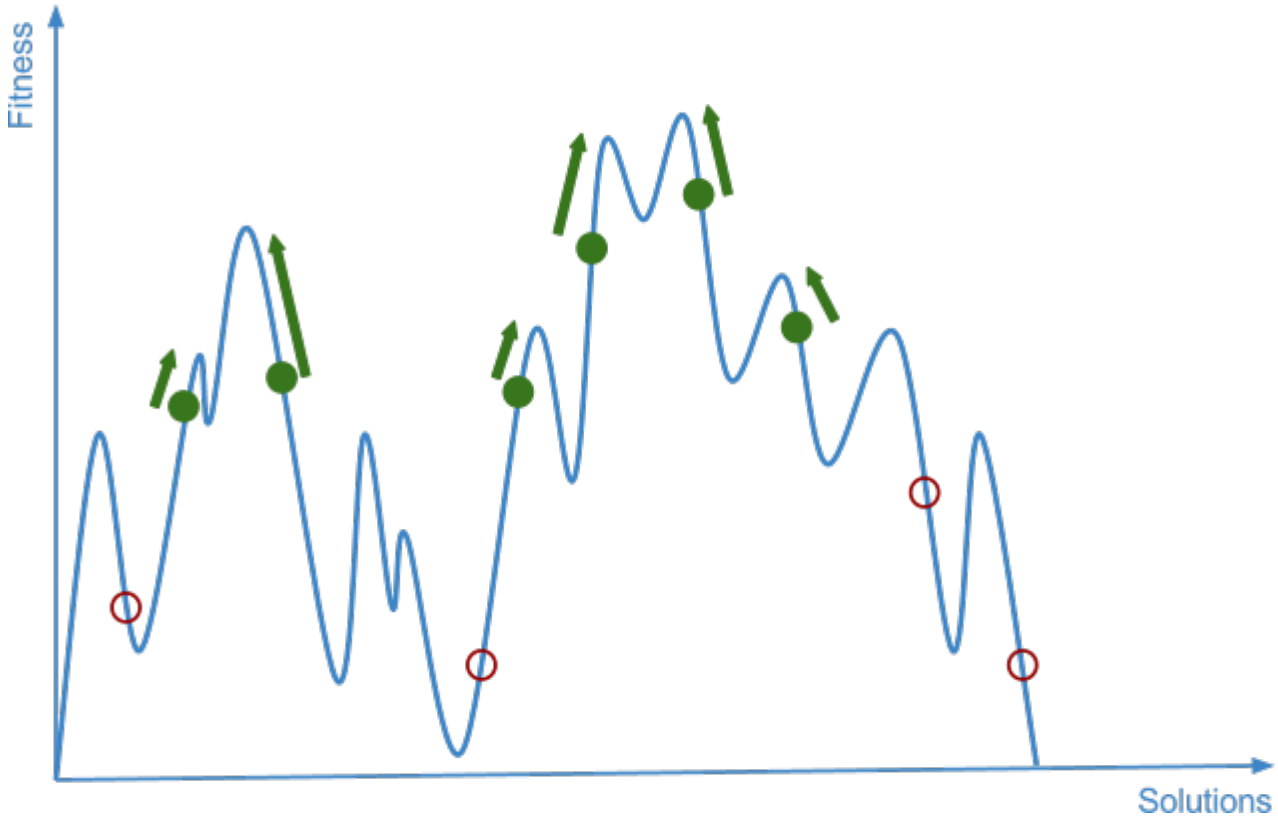
In this illustration, the better solutions are identified by solid green circles.

This process of allowing participants to submit and rank each other's ideas is an effective way of exploring the fitness landscape and identifying the better solutions.

However, each submission relies on the individual expertise and knowledge of that participant and we might need many submissions before we happen to land on the very best solutions.

Editing each other's solutions

By allowing participants to edit each other's ideas, we can attempt to build on existing solutions and find the very best solutions, i.e. the peaks on the fitness landscape. In order to do this, we borrow principles from genetic algorithms, and evolve existing solutions through controlled change.

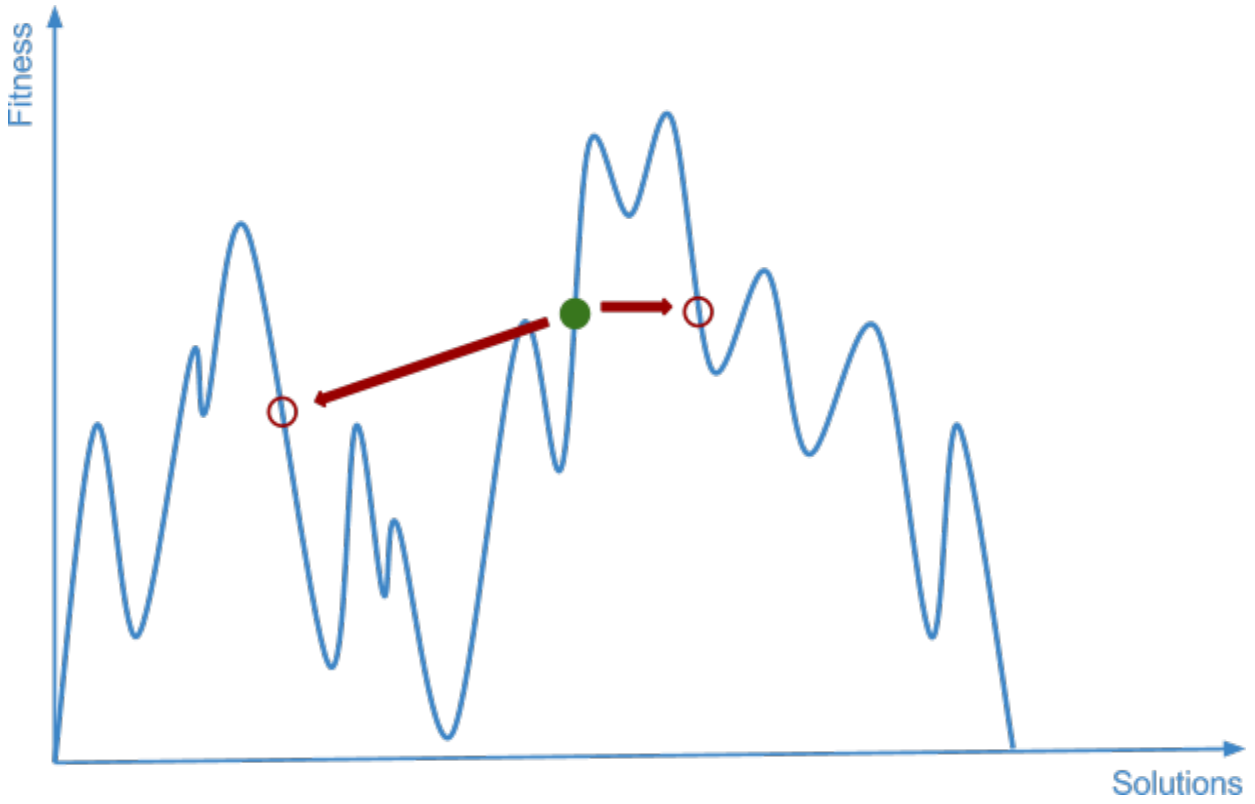


Restricting change

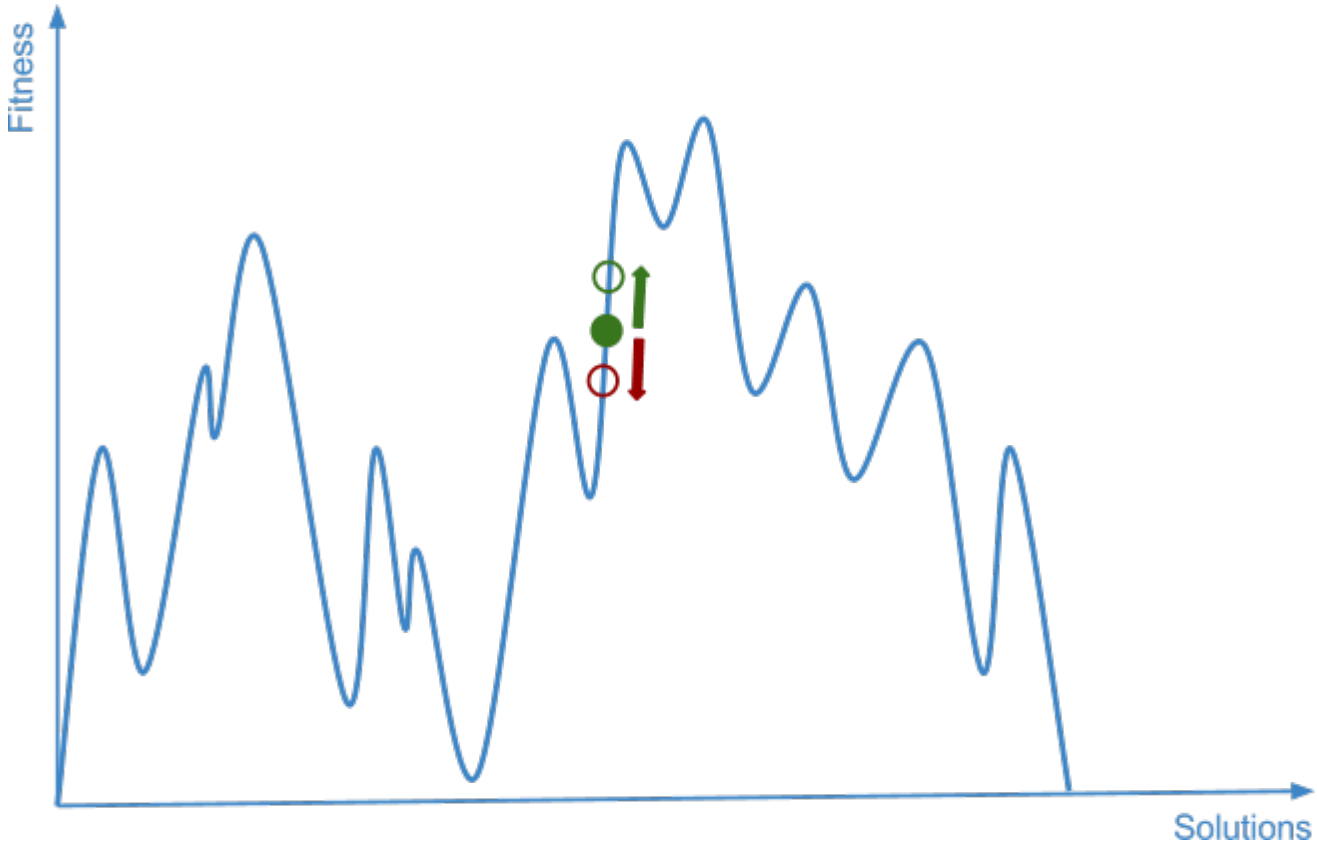
Mathematical genetic algorithms apply a specific mutation rate on the genome to control the efficiency of the evolution. Too much mutation and the changes cause haphazard jumps across the fitness landscape. Too small and the evolution stagnates.

Too much variation

If the edits are unrestricted or are restricted inconsistently, it is possible that an edit will jump too far on the fitness landscape, and we are no better off than receiving a completely new submission:


Fitness
Solutions

Controlling and optimizing the process



We need to ensure that the edit is an adjacent point on the fitness landscape. In so doing, the edit will either be a little bit better or a little bit worse. By receiving enough edits controlled in this way, we can increase the chances that we climb the hill of the landscape and reach the peak of any given solution.

An implicit genotype

In the Codigital process, there is no genetic specification of the content (genotype). There is only the phenotype, i.e. the expressed form of the content. The changes are applied directly to the phenotype. In order to control our search over the fitness landscape, we must ensure that competing changes are of comparable size in terms of their information distance.

Our technology analyzes the edit and derives the amount of information that has been added or taken away. This can be thought of as the consideration of an implicit genotype. I.e. if this solution were coded genetically, how much genetic change would have been needed to display this amount of change in the phenotype.