Using Pareto Front for a Consensus Building, Human Based, Genetic Algorithm

Pietro Speroni di Fenizio[1], Chris Anderson[2]

[1] Engenheria Informatica, Coimbra University, Coimbra, Portugal
speroni@dei.uc.pt,
[2] Turnfront,
chris@turnfront.com,

Abstract. We present a decision-making procedure, for a problem where no solution is known a priori. The decision-making procedure is a human powered genetic algorithm that uses human beings to produce variations and evaluate the partial solution proposed. Following [1] we then select the pareto front of the proposed partial solutions, eliminating the dominated ones. We then feed the results back to the human beings, asking them to find alternative proposals that integrate and synthesise the solutions in the pareto front. The algorithm is currently being implemented, and some preliminary results are presented here. Some possible variations on the algorithm, and some of its limitations, are also discussed.

1 Introduction

Decision making is a common challenge in any community, irrespective of its size. Every form of government is essentially a way to solve this problem, but it is also solved (or suffered) in groups that are too small to have any form of official decision-making structure. Often the way to find an agreement is through a democratic voting system. In this way the options are listed, and everybody votes for their favourite option. Theory of voting is generally considered part of game theory, and it has a long history. For a good review we suggest [2]. Mostly the differences between the various voting systems are limited to (1) how many options can each member of the community endorse; (2) if, and how, the options chosen are ranked according to preference, and (3) how the votes of the various members are integrated to select the winning proposal. The general structure is then: (a) the possible options are spelled out (often by some member of the community); (b) everybody votes; (c) through an algorithm the votes are counted; (d) the winning proposal is interpreted as the representative of the community global desire.

Of course unless the decision was voted for unanimously, not everybody will have favoured the winning proposal, and thus some members of the community will be forced to accept a decision they do not favour. If this happens on a recurring basis, involving a stable group overruling a smaller group, it is common to talk about a tyranny of the majority. It is also well known in voting theory (and practice) that requiring unanimity before accepting a decision tends to freeze a community when its size grows too much. How much is "too much", is different from community to community, but often between 10 and 20 members are enough to freeze a community who is trying to achieve unanimity. Is there really no other option?

2 The Unspoken Assumptions

There are a number of assumptions underlying all of these voting systems. We shall try to list them, and propose an alternative decisions system not based on them. Firstly, there are some assumptions being made about the size of the solution-space of possible actions that can be undertaken by a community. It is generally assumed that these possibilities are few (first assumption), are clear (second assumption), and can be easily recognised (third assumption) and acknowledged (fourth assumption). By assuming that there are few possibilities, we also assume that it is possible to list them all (fifth assumption).

If we break free from these assumptions, we can instead suppose that the solution-space is vast, needs to be explored, and no single human being can see all the possible solutions. We can even assume that at the beginning no group of people can see all the possible solutions. Instead of finding an algorithm that selects a solution to implement from a small list of possibilities, we can attempt to find a solution that will be accepted by as many people as possible.

Framing the problem simply as listing the possibilities, and voting for them, ignores an important aspect: mediation. Some, generally few, individuals, when posed with different positions will try to mediate among the various possibilities, trying to find a solution that can be accepted by more people. Not only are those people rare, but also communities need to have one of these people in a particular position of power, to be able to benefit from their ability. So we could say that in a community where everybody can vote, there are generally only few individuals that put forward the possible proposals on which everybody else will vote, and even fewer that are actively working to mediate between the various groups. Those mediators are the ones which are effectively looking for possible solutions that can be endorsed by a wider base.

What we want to suggest here is a system where every member of a community has the opportunity to present proposals, mediate between existing proposals, endorse others proposals, and finally where this happens in a cyclic way in such a way that eventually an optimal solution is eventually reached.

3 The Algorithm

The algorithm that we are proposing in this paper, is a genetic algorithm, that explores the space of possible solution to the question posed. Looking for a solution that can satisfy the maximum number of people; potentially satisfying everybody.

The algorithm starts with a question being posed. It can be posed by one of the users, or it can be posed by an external person. Then every user will be allowed to write a possible solution (called proposal) to the question. At this stage the proposals are secret, and no user is allowed to see the proposals written by the other participants. When everybody has written their proposal, the writing phase ends, and the algorithm moves to the endorsing phase. Now everybody is allowed to read all of the proposals, and endorse the proposals they agree with. It is important in this phase that each user endorse *all* of the proposals that they agrees with. No limit should be set on the number of proposals that a participant can endorse. In the worse case a participant will only endorse their own proposal. If a participant is instead satisfied with all the proposals, they can endorse them all. Once everybody has endorsed the proposals they are happy with, a selection process happens.

We define a proposal A as dominating a proposal B, if the set of participants that endorse proposal A strictly contains the set of participants that endorse proposal B. Of course if A dominates B, and B dominates C, then A dominates C.

To select the winner proposals we eliminate all the proposal that are being dominated by any other proposal. What remains is a Pareto Front of the proposals. Note that each participant will be present in at least one proposal. As such the Pareto Front can be said to represent every person that has participated, so far. As the selection ends, we say that also a generation (or a turn) has passed.

If the selection process produced a single proposal, this must necessarily be endorsed by everybody. We then decide that the question has been answered, and an unanimously acceptable solution has been found. If the selection process did not produce a single proposal, the process continues with a new generation.

Now at this new generation the participants are presented with the question that was posed (the same as before), and the pareto front of the proposals that won the past round. All the other proposals from the past generation have been eliminated, and will be ignored. The participants are now invited to write new proposals, taking the previous Pareto Front as an inspiration. They should try to find possible synthesis among them. Although they are encouraged to integrate the proposals, each participant is allowed to write anything they want. They can introduce new solutions, re-propose past solutions that did not make it to the Pareto Front, and rephrase past proposals. If a participant feels that he has nothing to contribute, and that his view is fully represented by the proposals that made it to the Pareto Front, he is also allowed not to contribute at all at this stage. Once the participants (who wanted to write) have written their proposals, again we move to the endorsing phase, and then to the selection stage, and so on.

The process continues through writing, and endorsing phase, until the system has either converged to a single unanimous answer, or the system does not seem to produce any more variations, and generations after generations the Pareto Front is always the same.

The system is very simple, it is a genetic algorithm that uses human beings to produce the 'genetic' variation. It also uses human beings to evaluate the generated partial solutions. And then then limits itself to the pareto front of the proposals.

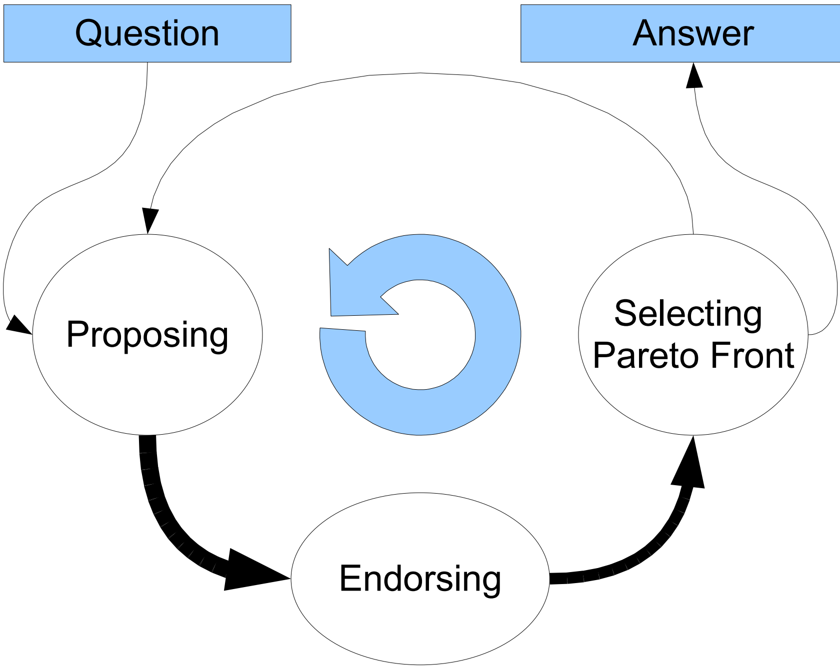

Fig. 1. A schematic representation of the genetic algorithm

We have already discussed in the introduction, how we moved from framing the problem as finding the best solution among a given set, to find the best solution, in an open context. Once we have framed the problem as a search problem, using a genetic algorithm is a natural choice.

What we are looking for is a solution which is expressed as a text describing a set of actions. If we were trying to produce such a solution automatically we would incur three different problems. Each unsolved so far. First of all there is no automatic way to translate a text in a set of actions. Solving this problem would be equivalent to solve the general problem of automatically finding the semantic meaning of a text. Although a lot of research has been done so far, the results generally apply only for very specific cases. The second problem is to select and filter the actions that are meaningful and possible, from the ones that simply do not make any sense. And this is a second unsolved problem, which would instead require a computer program to have an understanding of the physical world, and its laws. And finally the genetic algorithm would have to be able to evaluate which solutions are better than others. None of those problems have been solved so far. So instead we decided to let human beings produce the possible solutions, and again human beings evaluate them. Selecting which to pick for the next generation is in itself not a straightforward task. There are three requirements that need to be satisfied: (1) we want the algorithm to converge in few solutions, that eventually (through various generations) give rise to a single answer; (2) we want the set of solutions to represent the whole panorama of possibilities, and (3) we want every participant to feel that the solutions they endorse are part of this panorama.

In the literature there are different threads of research that studied this problem. If we look at this as a voting problem, there is the whole voting theory that has been developed. On the other hand, if we look at this as a genetic algorithm problem, there is also a huge literature on the subject.

As a voting problem the assumption has always been that each voter had an equally valid point of view, and so it was natural and fair to just sum them. But basically each voter represents a different dimension, on which each of the proposals would be evaluated. If we sum the votes, we are evaluating a bunch of points in a unidimensional space. And we just need to decide where to stop. How many proposals should be allowed to the next generation. At this point the proposals are also playing a transitive game. If proposal A, wins over proposal B (i.e. A has more votes than B), which wins over proposal C, then A wins over C. But we are effectively throwing away a lot of information. In particular we are ignoring who has voted for what proposal. So the result is suboptimal, in the sense that many participants are not really considered in the result.

If instead we consider this information, the problem moves from a transitive game to an intransitive game; now a proposal A can be better than a proposal B (according to Joe), which can be better than the proposal C (according to Suzanne), which can again be better than the proposal A (according to James). In this case no proposal is then inherently better, and we are faced with what is called an intransitive cycle. In the field of genetic algorithm Bucci and Pollack [1] have shown that it is possible to escape those intransitive cycles by using Pareto Fronts.

Not only does this avoid the potentially dangerous place of intransitive cycles, back to a transitive games; now a proposal A dominates a proposal B if and only if A is bigger or equal in all dimensions to B (no participant prefers B to A), and there is at least a dimension in which A is strictly bigger than B (there is one person that prefers A to B). So if A dominates B, and B dominates C then A will dominate C. This effectively permits us to drop all the proposals that are being dominated by another one. What remains is what is called a Pareto Front. While we are doing this we are not losing the representation of any participant: if proposal A dominates B, and we drop B, all the participant that have endorsed B will have endorsed also A. So by ignoring B we are not ignoring any participant input. They are all present in A. Now assuming that each person has endorsed at least one proposal (which is a safe assumption, since at the very least they would have endorsed the proposal written by them), then each person will be present in the Pareto Front.

This solution also solved the problem of how many proposals to take-on to the next generation. We need to take all and only the proposals of the Pareto Front. If we took less, the solution would not be inclusive, and we would run the risk of not representing all the participants. If we took more, we would be keeping a solution which is unnecessarily redundant.

4 Limits, Problems and Variations

While the basic algorithm is quite simple, there are a number of possible variations that should be tested. Some of those variations are presented here.

Anonymity of the proposals. Should proposals be anonymous, or should the people who are endorsing the proposals be allowed to know who has written a particular proposal. If the proposals are anonymous the participants are forced to read the whole proposal, before endorsing it. Will they be able to understand it? This depends on the topic, and on the group. The philosophy behind this choice requires each proposal to stand on its own merit, and not be endorsed thanks to the popularity of the person writing it. On the other hand, if the proposals are not anonymous, people who are interested in the topic, but lack the technical knowledge to understand the subtle elements of it, could still participate in the vote. In a system where the key element is a comparison between sets of voters, and not the actual counting of the endorsers, the anonymous choice seems to be a natural one.

Anonymity of the endorsements. After the endorsing phase, the successful Pareto Front of the proposals are fed back to the participants, asking them to write new proposals. When this is done the name of the participants that have endorse each proposal can also be made public, or not. By making those information public it permits to the participants to understand which are the major proposals, also it works as a light social control system, to avoid participants abusing the system in an antisocial way. In this system each user has significant power. If a participant writes a proposal, and then endorses only his own proposal, he can be sure that his proposal will be present in the pareto front. This behavior is possible, and even socially acceptable when a person is honestly in disagreement with all the proposals that are being presented. But it can be abused by using it as a way to protest. By letting everybody see who has voted for what, this kind of antisocial behavior is exposed, and generally ceases. On the other hand an anonymous system would permit everybody to endorse what they truly believe in. So in this case both of the possibilities make sense, and both should be tested.

Who is allowed to write, who is allowed to endorse? In our description we assumed that everybody who was allowed to write a proposal was allowed to endorse them. This does not necessarily need to be so. If the participants who are allowed to propose are a subset of the participants who are allowed to endorse, we have a situation who is similar to a modern democracy, where few people define the options for everybody else. We are not particularly interested in this situation, as it has already been tested enough in modern democracies. If instead the participants who are allowed to suggest proposals are a supra-set of the set of the endorsers, then we have a situation in which a community is discussing an issue, and external people are allowed to insert new ideas. This situation has rarely been tested. Another, different, possibility is a situation where no one who has proposed something is allowed to participate in the endorsing phase. If this is done by splitting the group into two subgroups at the beginning, and keeping every proposal anonymous, this last option might produce interesting results.

Changing the Participants During the Process. So far we have assumed that the same participants that have written a proposal one generation, will also write it on the next generation. And somehow this would probably be an optimal situation, because, since the people participating are also the ones evaluating the results, this defines a static fitness landscape on which the genetic algorithm can climb. Unfortunately this is not always possible. Since this decision system requires multiple voting generations, and in general a protracted interaction between the users and the system, it is possible (and even common) that the participants in a generation (or even between phases, inside a generation) might change. If the community is big enough this is not necessarily a problem, provided there are enough participants to represent the various possible ideas, the system can keep on finding an answer. If the community is small, changing the participants half way seems to produce the most unreliable results. As it would be running a genetic algorithm where the fitness function changes from one generation to the next. Unfortunately most of the tests that we could do so far have suffered from this problem.

Real Questions versus False Questions. Although the work is still preliminary, we already noticed an interesting pattern. We tried the algorithm several times, on various questions. Every time the question was a real question, among real participants, which were going to have their life changed by the result, the algorithm seemed to work better. It would act in a more predictable way, it would converge more rapidly. When more than one result was in the Pareto Front, the participants would try harder to synthesize an acceptable compromise. When instead the question was irrelevant, the answer were random, the endorsing was random, and the algorithm did not seem to converge easily (if it would converge at all). All this seem to suggest that the algorithm is indeed exploring a space of possibilities. And when the question is a real question, there is a definite fitness space to be explored. With peaks, valleys, and neutral ridges. When instead the question is irrelevant (to the participants), the algorithm is unable to find any real synthesis because no real synthesis is there to be found. This suggests that future work should be done on participants that are really involved with the results of the procedure.

5 Partial Results and Conclusions

At the moment we only did preliminaries studies on the subject. We tried it out among six participants with pen and paper. We then implemented the algorithm on a website (http://pareto.ironfire.org), and invited some testers to try it out. All the results are promising, but not consistent enough to make a statistical case. We will thus only relay them, as an anticipation of some future work. On the pen and paper example, the question posed was: "We are going for one month together, in vacation, this summer. What shall we do?". This test only lasted two generations. On the first generation the answer were: "go camping"; "help my grandmother with her garden" (from participant 'D.'); "join a construction site and build a house"; "go biking in east Europe"; "go to thailand"; "go to Canada". After the evaluation process the Pareto Front only included two proposals left: "go to Canada" and "help my grandmother with her garden". Then the participants were invited to write new proposals. Five out of six proposals suggested to "go to Canada with D.'s Grandmother, and ...". The sixth proposed to pay a gardener for D.'s Grandmother, and then go to Canada. We notice here an interesting result. The proposals seem to get more complex as the generation passes. As if the algorithm started by exploring the space of possibilities, in a more general way, and then become more precise in successive generations. Everybody is effectively trying to mediate between the elements. It was also interesting that each person tried to reinsert what they really cared for, in the next proposal. For example the participant that first suggested to "go biking in east Europe", on the second generation suggested to "go to Canada with D.'s grandmother, and go biking, after leaving D's grandmother in a camping site." A decision making algorithm was presented to permit to a community to investigate and discover the most widely endorsed proposal that answers a given question. A number of possible variations were discussed and an example of the partial results was presented. The next phase of the investigation will consist in testing the algorithm with bigger communities, for longer time, as well as testing the effect of the possible variations.

6 Acknowledgment

We would like to thank for the interesting discussions: Giovani Spagnolo, Manuel Barkau, Christopher Ritter. We also thank all the people who participated so far in our tests.

References

1. Bucci, A., Pollack, J.B.: A mathematical framework for the study of coevolution. In: Foundations of Genetic Algorithms 7, Morgan Kaufmann (2003) 221–235
2. Taylor, A.D., Pacelli, A.M.: Mathematics and politics: Strategy, voting, power and proof. springer book (Mar 2008) 1–377