AUDENS
Revista estudiantil d'anàlisi interdisciplinària

Núm. 4
2 0 2 1

When Accidents Become Design Choices: Navigation Systems, Rat-Running, and AI Safety

Antoni Lorente Martínez[1]

Abstract: There are many artificial intelligence algorithms that work particularly well when trying to find the best solution to a given problem from the set of all possible solutions. However, such an unprecedented ability to solve optimisation problems only stresses the need to carefully pick out the right goal to be optimised. In this regard, and taking route-planning services as a guiding example, I claim that the current problem definition for route-planning algorithms prompts disruptive driving practices such as intelligent rat-running which create, in turn, global problems by intending to optimise local ones. In order to avoid this, I defend that the design approach to such algorithms should aim for hybrid search strategies that constrain the local benefit to the global costs of a given solution, in order to set the grounds for a safer AI in the future.

Keywords: Optimisation Problem; Search Strategies; AI Safety; Navigation Systems;

Introduction

To drive a black cab in London, one needs to pass "the knowledge", a thorough oral test in which the candidate must be able to connect by heart, turn after turn, two points selected by the examiners from the map of London city, instantiating — almost

[1] antoni.lorente_martinez@kcl.ac.uk, Universitat Autònoma de Barcelona & King's College London.

nostalgically — an old paradigm of route-planning and road navigation. Common humans, once they sit before the driving wheel, may know the route to some places by heart, or even more suitable alternatives depending on the traffic experienced in past occasions. However, navigation systems such as Google Maps exempt us from having to plan or learn a vast set of routes and their variations by providing us with an interactive route-planner, which can be filtered to some extent in order to accommodate our preferences.

To solve the problem of getting from point A to point B, we no longer seem to need to possess the knowledge — either in its most impressive or in its more humble form — of where do bottlenecks form, which are the least congested routes at the rush hour, or which streets are suitable alternatives if we encounter an accident. Mobile devices and algorithms produce, aggregate, and reproduce information, triggering a change: from possessed knowledge to accessible information.

Connecting two points in a map in the most effective way belongs to a particular set of problems commonly known as optimisation problems: i.e. finding the best solution to a given problem from the set of all feasible solutions. As the medieval proverb praises, "all roads lead to Rome", but if you are subjected to some constraints — time, for example —, you may want to find the fastest path that will get you there: you will need to select, from the set of all possible options, the fastest one. And even though we humans face this kind of problems on a daily basis, when it comes to optimisation artificial intelligence seem to be unbeatable. This is, indeed, what enables the transition described above from knowledge to information: the fact that powerful algorithms can solve hard problems or make sense of big amounts of data which would otherwise be incomplete, even intractable, to us.

Yet the main concern when it comes to optimisation problems is not to find the best answer, but actually to raise the appropriate question. In this regard, in the following discussion I evaluate some of the challenges and consequences of picking an optimisation strategy by means of an example: search algorithms in navigation systems.

To do so, I begin in Section 1 by providing a brief overview of optimisation problems regarding route-planning algorithms. Then, in Section 2, I discuss three possible strategies to solve such problems — individual, collective and hybrid —, which I present graphically in order to better understand how these three approaches relate to each other and to two further constraints: practicality and legality. To finish, I also discuss the feasibility of each group of strategies according to how traffic is conceived. Finally, in Section 3, I insist on the relevance of the example of navigation systems, from which valuable insights can be drawn in order to ensure the future of safe developments in AI. Along these lines, I stress the need to rule out antisocial behaviours in current low-impact applications in order to prevent more dire consequences in future developments.

§1. Behind the scenes of route-plannig algorithms

Search algorithms are a subset of computational tools that find — or at least aim for — the most efficient connection between two nodes in a graph. In short, a graph is a set of nodes (or states) connected by paths (or actions). To build search algorithms each path in the graph has an associated cost allocated in turn by a path cost function. Thus, a given problem may have multiple solutions, yet all solutions must be action sequences that take from the initial state to a final state (or *root* and *goal state* respectively). From the set of all such possible solutions, the optimal one will be the one that has the lowest cost. (Russell and Norvig, 2010:68-9)

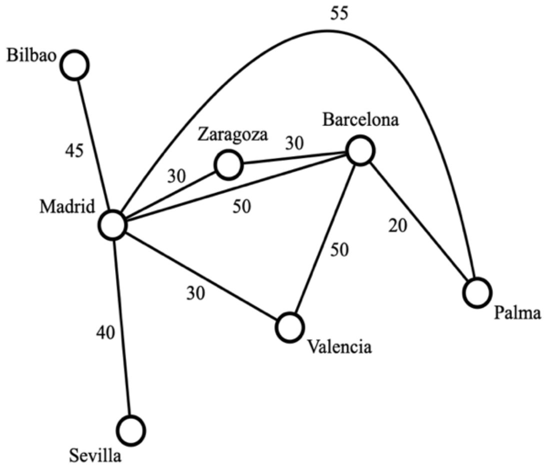

Figure 1. A simplified weighted graph between some Spanish cities. Source: the author

A search algorithm can be fed either the root state, or both the root and goal states. In the first case, the algorithm provides what is commonly known as the shortest-path tree — a tree-like map connecting the source with every other node in the graph in the most efficient way. However, if the input consists of both the root and the goal states, the algorithm outputs the optimal path between such nodes. In any case, a search algorithm needs a search strategy, i.e. some criteria to guide which node is to be expanded next.

According to the search strategy, the algorithm will perform differently depending on various features such as the branching factor (or the maximum number of successor nodes), the depth of the shallowest goal (or how many nodes must the algorithm go through before finding the first goal) or even the amount of memory needed to perform the computations. (Ibid., 80)

In connection therewith, a simplified model of any route-planning service can be thought of as a search algorithm that knows both the root and goal state, and that defines a path cost function according to constraints that are already set, such as open roads or one-way streets, together with those defined by the user like a preference for routes that avoid tolls or motorways, for example. If problem-specific knowledge like this, and not just the definition of the problem is available, algorithms rely on what is known as informed search strategies. Such information boosts the programme to find a solution to the problem in a more efficient way. (Ibid., 92)

Some candidate algorithms for modelling route-planning services are Dijkstra's Algorithm, which is usually taken as the benchmark for comparison, some variation of it, heuristic approaches such as A* algorithm, or combinations of speedup techniques to improve the performance of the strategy. (Sanders and Schultes, 2007:24-8) The technical differences behind these strategies, however, are only tangentially relevant to the philosophical question that motivates this exercise. But the idea that search algorithms try to minimise the cost of connecting two nodes in a graph will need to stick around until the end. In the concrete example of navigation systems, the graph represents a network of locations connected by roads. Thus, the algorithm is tasked with finding the least costly path between the user's starting point and the destination. Once this is done, the journey can start.

§2. CHOOSING A STRATEGY

Suppose you want to cook a particular dish. A quick search on the internet may provide you with multiple entries: hundreds, maybe thousands. Some versions of the recipe will prime time over taste; others will focus on the benefits of slow-cooking some of its parts. Yet you must pick one: you need to define your goal. One possible way to tackle this problem is to pick a single objective. If your main concern is time, you may want to find the recipe that allows you to spend the least time possible cooking the dish.

This may entail a cost in terms of energy or sophistication — but however costly, these are not set as relevant goals. Consistently, a single-objective optimisation problem usually yields a single solution, which is the optimal. (Deb, 2014:V)

But it is also possible to optimise your solution with multiple goals in mind. In such case, multi-objective optimisation strategies try to strike a balance, finding solutions capable of accommodating more than one competing goal. The price to pay for that is that such solutions are rarely 'optimal' — in the sense that conflicting objectives cannot be optimised simultaneously — but rather 'pareto optimal', or non-dominated. Moreover, multi-objective optimisation does not propose a single solution, but a set of non-dominated alternatives derived from a strategy which attempts to optimise all relevant goals, but since it fails to do so, it proposes a set of trade-offs. (Ibid.) Hence, pareto optimal solutions are not better than the optimal solution in each one of the relevant senses, but better compared to the optimal if the problem is defined in terms of multiple goals.

At this point, however, it is worth noting that optimisation problems are not single-objective or multi-objective: *strategies* are. Thus, the way in which a given problem is defined and addressed — say cooking a dish or going from one point to another — determines the nature of the possible solutions. It is with this that the philosophical relevance of picking a search strategy is made clearer.

§2.1. Individual strategies

When we use a navigation system we look for the best way to get from A to B. Once we set the general question to a particular one by providing the input data (i.e. the root and the goal states), the search algorithm begins operating in order to optimise the route. Nonetheless, this optimisation process is subject to certain constraints: existing roads, whether such roads have one or two sides, speed limits…

Some constraints are the consequence of cars driven by humans being non-cooperating objects. As such, each car has certain degrees of freedom, which are in turn not bound by the degrees of freedom of other cars. Take for example the driving speed. A particular car's speed does not determine unequivocally the speed of the other cars; it does, indeed, influence it if drivers aim to avoid accidents, but a car circulating at a low speed does not impede other cars to overtake it or drive at a different velocity — and the same goes for direction. It is before this non-cooperating nature of cars that driving rules are necessary. Speed limits and recommendations, or the direction of different roads are introduced as mechanisms to facilitate the circulation of free objects that would otherwise be unable to operate in a given space. Because of this, driving in the wrong direction or traversing non-paved ways are options that the search algorithm does not even consider.

Navigation systems make use of data available to them in order to optimise the solution to the problem raised by the user. Traffic congestion, for example, is estimated by gathering and interpreting GPS tracking units, mobile phones, or real time video surveillance. (Petrovska & Stevanovic, 2015:1489) With this, navigation systems alter the cost values associated with each action, informing the search algorithm to find a better solution for the driver in a more efficient way. Hence, and by stipulating the optimisation problem as one to be tackled from an individual perspective, traffic is treated as a part of the environment in which the problem must be solved, a boundary condition, and not as a set of coupled problems with competing interests.

These algorithms tend to work fine in normal conditions: the solutions they propose satisfy the needs of the user without causing any further disruptions in the general traffic. Limit cases such as heavy traffic or accidents, however, put individual strategies between a rock and a hard place. By means of a practical experiment, Aya Kojima et al. found that some navigation systems encourage a practice called rat-running, which consists of using narrower streets — mostly in residential areas — in order to avoid congested arterial roads. (Kojima et al., 2015:15) However, and even though the central claim in Kojima et al. is that intelligent rat-runners expose pedestrians to a higher risk of suffering an accident, their results can also be interpreted in terms of traffic management.

In 1968, Dietrich Braess observed how adding one or more roads to an existing congested traffic network does not speed up, but actually slows everyone down. (Steinberg & Zangwill, 1983:302) Thereafter, the so-called Braess's Paradox affected research in multiple fields, but its impact became most notorious in a particular class in game theory called *congestion games*. Congestion games are particularly useful when it comes to traffic problems, since they model non-cooperative resource allocation in large-scale networks. Each player in the game selfishly selects a strategy from those available, trying to minimise her individual cost. (Frank, 1981:283) Even though there are cases in which Braess's paradox does not occur — e.g. if the network is *series of linearly* *independent* (SLI)[2] —, the practical prevalence of this problem is well recorded. In Europe, the United States of America or even South Korea, instances of road-removal leading to better traffic flows are legion.[3]

Thus, intelligent rat-runners and, consequently, navigation systems force the paradox by introducing new routes into the network — this is especially visible in congested highways, where drivers take exits to cut traffic and re-enter the highway —, causing a disruption in an already fragile equilibrium such as heavy traffic. A key factor here is that players do not cooperate. In a framework where drivers are, de facto, non-cooperating agents, individual strategies seem the most reasonable strategies for optimising route planning problems. But maybe something more could be done. One particular line of research to address this fact has focused on the use of connected autonomous vehicles to stabilise traffic, for even though human drivers do not cooperate, they react to the vehicles preceding them.[4] Thus, and by embedding controlled autonomous vehicles within traffic, disruptions are mitigated. In this scenario, however, strategies can no longer focus on the individual: they must be defined collectively.

§2.2. COLLECTIVE STRATEGIES

A natural alternative to single objective optimisation strategies would be multi-objective strategies. These find, from the set of possible solutions, not the optimal one (in terms of a single parameter) but rather a set of trade-offs that satisfy multiple goals at the same time. Along these lines, if a given navigation system did not optimise a user's route from one point to another taking traffic as a constraint, but actually considering its impact on the traffic which the user is part of as another goal to be minimised, some of the problems described above would not even arise. However, a mere shift from single to multi-objective optimisation strategies would be insufficient for multiple reasons, considering the way in which current traffic and navigation systems operate.

First, navigation systems are designed to make the driving experience easier for their users. Maybe drivers do not know how to get to a given destination, or maybe they

[2] See Acemoglu et al., 2018:898 for an exhaustive discussion on Series of Linearly Independent networks

[3] The Cheonggyecheon restoration project in Seoul is a good example of an initiative to revert the Braess's Paradox

[4] See Vinitsky et al., 2018 or Wu, 2018 for further discussion

know multiple alternatives and expect the algorithm to find the one that is best *for them*. Therefore, a stark shift from single to multi-objective optimisation strategies would not be enough: it could work if it came accompanied by an increased reciprocal altruism. But that, simply put, seems too much to ask for. Second, and in line with this, navigation systems are commercial products developed by private companies that expect a benefit from drivers using them. Thus, asking, or even forcing them to provide a "worse" service to their users — even if it is collectively convenient — seems to stand contrary to any business purpose. Last, the advantages of moving beyond single-objective optimisation strategies would only be truly significant if *all* the navigation systems did so. If not, services grounded on individual search strategies would compete with a rather non-competitive alternative in terms of "user experience", and the problem of intelligent rat-runners causing traffic jams, for example, would reappear quickly.

Such problems prevail for the same reasons single-objective optimisation algorithms do: drivers — or cars, for that matter — do not cooperate. Consequently, traffic services for different users and the search-strategies behind them need to be conceptualised as independent from car to car. One consequence of this, as we have seen, is that algorithms learn to prompt abnormal behaviours such as rat-running, which may occasionally benefit some, causing nonetheless larger scale problems.

But the days for this may be numbered. We seemingly are at the brink of a paradigm shift: for decades, since vehicles powered by engines appeared, the idea of driverless cars has been fuelling the research agendas of many. It has been, however, with the latest developments in self-driving technology that what once seemed a mere fiction is now closer to reality.[5] Driverless cars raise legal, technical, or even ethical questions. Nonetheless, and besides the challenges that will need to be faced, these technologies also have significant upsides.

Researchers working on self-driving vehicles develop and assess different protocols to deal with all the manoeuvres that any car must be able to execute. In short, protocols are standard sets of rules that enable the communication between objects. Therefore, one of the foundational premises of driverless technology is that objects are not independent: they conform to a mesh. Shunsuke Aoki and Ragunathan Rajkumar, for example, developed a merging protocol for self-driving vehicles in order to avoid collisions when

[5] The American company Waymo, for example, released a service of driverless taxis in Arizona in 2020

two vehicles intend to occupy the same spatial area at the same time. (Aoki and Rajkumar, 2017:219) But perhaps the most relevant thing here is that the idea behind their work is fundamentally at odds with the principles that guide road traffic nowadays: while driverless cars will inevitably need to negotiate and cooperate, traffic as we know it is not based on cooperation.

The fact that different agents bear dissenting goals encourages drivers to constantly engage in cooperative and competitive strategies in order to increase positive outcomes or decrease negative ones respectively. (Vanderhaegen et al., 2006:192) But when competitive behaviours are reinforced by systems that are not based on a conception of traffic as a coupled problem, undesirable outcomes like the ones discussed above arise. While multi-objective optimisation strategies could indeed eliminate these problems, implementing them in an environment of non-cooperating objects would be detrimental, for there would be a conflict between self-driven and regular cars.

§ 2.3. Striking a compromise: hybrid strategies

If we consider the discussion so far, it is possible to represent search strategies in two different sets. In the first set we find individual strategies, which prime the benefit of one individual or user inspired by single optimisation strategies. The second set comprises collective or multi-objective optimisation strategies, which yield Pareto-optimal solutions.

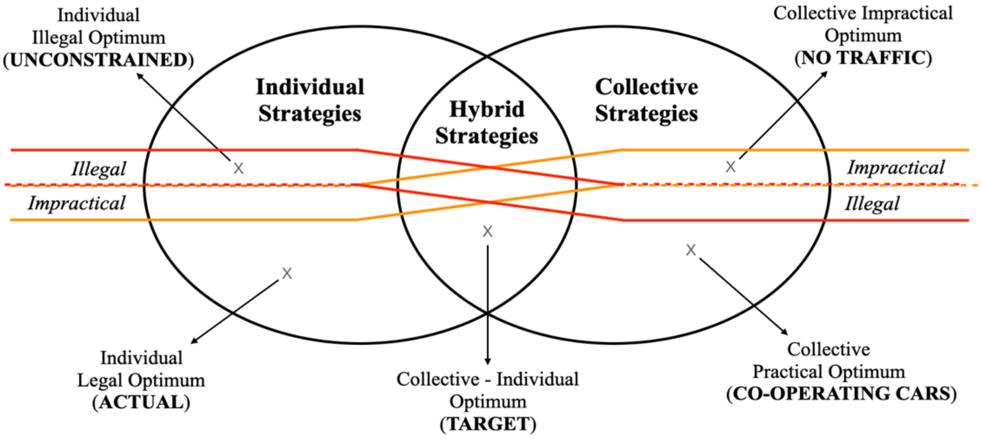

Figure 2. Optimisation strategies for navigation systems (areas are not to scale). Source: the author.

The figure above displays the relationships between two approaches to optimisation strategies and two further constraints: legality and practicality. In accordance, the set of individual strategies comprises optimal and sub-optimal solutions to the navigation problem when tackled form the perspective of a single individual within an external environment — i.e. the rest of the traffic. Furthermore, the set of collective strategies captures a similar variety with a slight methodological difference: traffic is not seen here as an aggregation of single problems that constraint each other, but as a larger-scale problem with individual, yet dependent, actors. From both of these sets, and from a developer's perspective, the orange and red areas that comprise illegal and/or impractical solutions should be disregarded.

The individual optimum *may* be one that infringes traffic regulations. If that is the case, however, unconstrained solutions should not be considered, and thus the resulting set of individual strategies available decreases. In a similar way, the collective optimum for cooperating cars *may* imply a complete root out of traffic. But this is impractical, for even though congestion problems, emissions or accidents would indeed disappear, it would not solve the problem: it would just dismantle it by means of a trivial solution. Moreover, some strategies could be both impractical and illegal, a status that applies to the three subsets represented in Figure 2: individual strategies, collective strategies, and their intersection.

Current approaches to navigation systems most likely fall within the kind of Individual Legal Optimum approaches.[6] They abide by the law and take the user's command as their main priority. On the other hand, developing frameworks for autonomous cars belong to the set of collective strategies. Consequently, self-driven cars are not treated as independent objects, but rather as decentralised units in a centralised co-operation scheme ruled by protocols that allow for planning and anticipation, but also for the mitigation of the effects of traffic anomalies caused by other drivers in mixed-autonomy traffic. (Wu, 2018:6012) But with the fully automated traffic paradigm still far, and before the problems that individual approaches to traffic entail, a compromise beyond individual strategies and collective ones needs to be reached.

[6] Google, for example, does not disclose what algorithms does Maps run on. However, it probably relies on a modified version of A* or Dijikstra's algorithms. (Mehta et al., 2019)

In this regard, at the intersection of both sets we find an area containing a group of hybrid strategies that take an individual's objective as their main optimisation goal without disregarding, at the same time, the collective effects of optimising such goal. In contrast with collective strategies, hybrid strategies enable a decentralised co-operation scheme, one in which traffic is conceptualised as the aggregation of multiple competing interests instead of as a boundary condition to an individual problem. When interpreted from a collective perspective, hybrid strategies contemplate multiple objectives which are not collectively evaluated, but rather fragmented into interactively conditioned optimisation sub-routines, which tackle narrower problems. However, this interpretation does not capture the full picture: hybrid strategies are a subclass in their own right.

Continuing with the example of navigation systems, if the search strategy implemented was a hybrid one, it would still aim to minimise the cost of the path between the starting point and the destination of a given user, in order to provide the best solution to her. Even so, the benefit of the solution proposed would be assessed against the costs that it would involve for traffic in general: the cost would not solely be computed as the time, tolls, or fuel needed between two nodes, but by means of a combined metric that contemplates such cost and its effects on the overall traffic. Hence, engaging in rat-running to bypass a congested section — e.g. if one is driving upon a highway and before the evidence of heavy traffic, takes a small diversion to then merge back — would be ruled out as an unacceptable praxis. This is neither Pareto optimal nor the best solution for the individual: my taking a certain road may impose an additional cost to other users that could be minimised via a collective approach, while it neither nudges me to engage in rat-running and, therefore, save some time. On the contrary, it tries to optimise the goal of a single user without putting them before the general interest, ruling out seemingly feasible solutions and behaviours that, regardless of the benefits they entail for the individual, end up triggering larger-scale problems.

Hybrid strategies strike a balance between local benefit and global cost: the forces behind the optimisation strategy are adopted from individual approaches, but the assessment of the performance is inspired by collective ones. They cannot be simply thought of as single-goal optimisation strategies with mere constraints, for the constraints here do not pick key features for the definition of the graph like speed limits, road networks or tolls, but rather the interests of other actors. Moreover, they cannot be conceptualised as multi-objective optimisation strategies either, for they do not aim for the best collective solutions: the objective is to maximise the local benefit without imposing too high a global cost. Thus, hybrid strategies are those that yield solutions to the navigation problem which satisfy an individual's interests by means of actions that are constrained by the effect they have on the overall ecosystem.

In the article "Urgency-aware Optimal Routing in Repeated Games through Artificial Currencies", Mauro Salazar et al. show a possible path towards materialising such hybrid strategies. Mirroring their results, navigation systems could be compelled to modify the ponderation — i.e. the cost allocation — of several alternatives via an artificial currency. Originally, such currency would be equally distributed and could not be bought or exchanged, but only spent or gained when traveling. (Salazar et al., 2020) In accordance, hybrid strategies would impose a higher cost to those alternatives that prompted the least beneficial behaviours from a global perspective. With this, users would be inclined to adopt prosocial alternatives that contributed to the overall benefit — which would not only be substantially cheaper, but even rewarded —, for otherwise they should have to bear a cost proportional to the global burden entailed by their decision. Consequently, users would need to selfishly evaluate whether the cost associated to their individual preferences and circumstances was acceptable relative to the tokens they had left, boosting in turn the motivation to save the most tokens to spend in the future, when they really were in need.

In conducting a numerical simulation of a similar incentive mechanism in a two-arc network via artificial currencies, Salazar et al. found that such mechanism attained a system-optimal solution with a substantial reduction of the agents' perceived discomfort, in comparison with a centralised optimal solution insensitive to the urgency of the agents. (Salazar et al., 2020:1) This suggests that artificial currency mechanisms could indeed be a feasible way to incorporate the principle underlying hybrid strategies, reflecting a weighing of the global cost of any single decision against its local benefit.

In this regard, intelligent rat-running would be one of those cases in which the local benefit would not justify the global cost it entails, and thus the price to pay in terms of the artificial currency would be higher. There may be other examples too. The key point here, however, is to acknowledge the impossibility of anticipating all of them. Consistently, when developing search strategies, the identification of highly rewardable or punishable behaviours should be understood as an iterative and interactive process, and not as an all-or-nothing approach to design. Algorithms will continue to be flawed — this is unavoidable. Nonetheless, if antisocial behaviours are detected, developers should be obliged to eliminate them.

§3. Mapping the problem onto the future

Current navigation systems nudge drivers into undesirable behaviours, and even though such behaviours *do* affect overall traffic, identifying what portion of the cost is directly caused by such services is no easy task. But beyond the difficulties to break down the economical or social burdens of such systems, the philosophical relevance of this example resides in the fact that it allows singling out some of the hazards of strategy selection in optimisation problems in general, well beyond the case at hand.

Together with the multiple technical challenges that artificial intelligence presents to researchers, the philosophical questions regarding this growing technology are complex and rarely explored in detail. That algorithms are value neutral is common belief. Computational tools are mathematical tools: they take an input and, according to their internal structure, they manipulate it to deliver an output. One clearly identified as a possible source of bias is the data used to train the algorithms; but the discussion of both the effects of biased datasets and the value neutrality of algorithms as ground truth falls outside of the scope of this particular paper. However, in focusing on navigation systems as a practical implementation of a search algorithm, some of the pitfalls of poorly selecting a search strategy have become clearer: by means of trying to avoid a specific behaviour — induced rat-running via navigation systems — and its consequences — traffic disruption —, a deeper philosophical puzzle has emerged.

As I have insisted at the beginning of Section 2, problems of optimisation are not single- or multi-objective: strategies are. Moreover, the selection of the goal [or goals] to be optimised and thus, the strategy, likewise constrains the set of possible solutions. Hence, and in order to avoid undesirable recommendations or other negative side effects, the goals of such programs should be defined beyond the mere interests of their users. But if we fail to do so, we may soon encounter several technologies reinforcing and reproducing selfish behaviours that work against the common good — something that, over the longer term, could derive into overreaching and unsafe AI applications.

In this regard, AI safety is a line of research that aims for the alignment of AI technology with the fundamental values of our current societies. Its main purpose is to avoid harmful technologies that, if sufficiently widespread, would jeopardise social organisation as we know it. However, and perhaps due to an excessive influence of fiction, most discussions tend to gravitate around dystopian futures, even if such futures are far-fetched. AI is growing fast, but even though there is a possibility that general artificial intelligence explodes, the actual dangers of AI cannot be reduced to such possibility. In fact, the real dangers of AI start with minor drifts embedded in successful products, and navigation systems, going back to the question raised at the beginning of this section, present a vivid example of unsafe AI hiding in plain sight. For maybe the overt hassles of poorly chosen search strategies are heavier traffic and higher greenhouse gas emissions, yet the latent danger is the precedent they set.

In the article ‘Concrete problems in AI safety’, Dario Amodei et al. discuss the impact of “accidents” derived from the implementation of machine learning algorithms that yield harmful behaviour — albeit unintended — from poorly designed real-world AI systems. (Amodei et al., 2016:1) Such accidents can be the consequence of a wrongly defined objective function, an objective function that is costly to evaluate frequently, or an undesirable behaviour during the design phase. (Ibid.) In line with this, and even though search algorithms are not machine learning algorithms but symbolic systems, the aim of this paper has been to emphasise the problems entailed by the definition of the objectives of a given program and the strategies implemented to reach such objectives. With this, I have tried to draw the line between accidents and sloppy design a little bit clearer.

Harmful behaviour can indeed start as an accident. However, when designers choose not to fix them, accidents are integrated into products as design choices: just like in performance optimisation, which is pushed until the cost of a better trained ML algorithm surpasses its benefits, accepting some unwanted consequences as affordable byproducts becomes part of the design. To address such negative side effects, Amodei et al. propose multiple strategies. Most of them draw on the impact of a reinforcement learning agent on the environment in which it acts, but they are suitable beyond reinforcement learning problems. (Ibid., 3-5)

One of the main obstacles to get rid of such behaviours is that applications and programmes — as commercial products in a market — fight a battle for dominance.[7] Consequently, when companies have successful products that solve a given problem and that are well-positioned within the race for market dominance, the cost of re-developing them — in terms of positioning or economical benefit — seems to surpass the social advantages of more prosocial versions of the software. Moreover, and since the "problem" behind a given product is hard to pinpoint and even harder to solve, unwanted effects remain as "the price to pay" for a great service.

The long-term dangers of accommodating unwanted suggestions in relatively low-risk products can be better understood from a psychological perspective. On the one hand, the effect of navigation systems that nudge users into selfish behaviours with negative consequences for the overall traffic works as a foot-in-the-door technique. If we accept the thesis that unaddressed accidents become design choices, what antisocial navigation systems do — at least from a social perspective—, is to normalise certain behaviours that are collectively undesirable. By accepting such "unfair" albeit affordable consequences, we are lowering the bar that could alert us before future and more harmful products, forfeiting the ability to go back if necessary. On the other hand, and if we stand by the view that accidents are just that, unpredicted effects, a conformity problem remains, for products — and the algorithms they run on — contribute to the process of forming the intuitions and expectations of users. Thus, if we decide not to address such effects now, when they are still minor, perhaps some of the tragic consequences that we fear in future technologies will irretrievably be taken as given once they appear, for we will have been renouncing the tools to confront them with every step into the future.

§4. CONCLUSION

Practical implementations of artificial intelligence are spreading rapidly: we are facing a technological boom that will undoubtedly modify how countless things have been done up until now. However, and far from dystopian fictions, such changes shall continue to be subtle. Navigation systems provide a good example of this paradigm shift: what once required vast collections of road maps and guides can now be done via an

[7] For a comprehensive theoretical framework regarding battles for technological dominance, see Suarez, 2003.

intuitive programme that not only *knows* all possible routes, but also updates continually the state of traffic or the road system. With this in mind, and throughout the discussion above, I have raised two main claims regarding the nature and the perils of this shift: on the one hand, strategy selection in optimisation problems does affect the social nature of an algorithm. On the other, when antisocial behaviours — even if accidental — are not properly addressed, they become design choices that, if incorporated as normal design, may hinder the path towards safe AI technologies in the future.

I have first provided a short introduction about search algorithms. In section 2 I have then considered three different approaches to optimisation problems: individual, collective and hybrid strategies, which differ in their goals. While individual strategies focus on the best solution for a single user — treating overall traffic as a boundary condition —, collective strategies reach trade-offs between multiple objectives, offering non-dominated or Pareto optimal solutions. Hybrid strategies, on the contrary, prioritise a user's individual goal insofar it does not impose too high a cost on the overall environment, striking a balance between the local benefit and larger scale burdens. One possible way to implement such hybrid strategies is by means of artificial currencies, which allow a fair and decentralised computation of the trade-off between local benefit and global cost.

Individual strategies like the ones used now work well until the traffic network is stressed. But with heavy traffic, for example, the cost of a given road increases, and single-objective strategies try to avoid it by nudging users into behaviours such as rat-running. Trying to minimise the local cost of a particular user in this fashion creates larger scale problems, such as an increased overall cost of road navigation. To counter this, a possible solution for the future would be to develop a network of cooperative driverless cars based on collective strategies. In the meantime, navigation systems should adopt hybrid strategies that treat traffic as a coupled problem and not a boundary condition, which would in turn foster more prosocial behaviours.

Finally, in section 3, I have outlined some of the ways in which analysing current search strategies in navigation systems can inform the discussion on AI safety. By tackling a rather affordable example, I have delved on the importance of properly defining the problem that is to be optimised when choosing a search strategy. Yet even then, accidents can and will happen: negative side effects will keep prowling the implementation of new algorithms. Yet this cannot justify the acceptance of such effects as the cost of progress: once a flaw is detected, if nothing is done to redress it, it remains in the product as a design choice of the developers. This, however, entails serious hindrances for a safe development of AI, for if we decide not to revert negative side effects now, we may lack the resources to detect and disarm unsafe behaviours later.

BIBLIOGRAPHY

- Acemoglu, D., Makhdoumi, A., Malekian, A. and Ozdaglar, A., (2018). Informational Braess ’paradox: The effect of information on traffic congestion. *Operations Research*, **66**(4), pp.893-917.
- Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J. and Mané, D., (2016). Concrete problems in AI safety. arXiv preprint arXiv:1606.06565.
- Aoki, S. and Rajkumar, R., (2017, April). A merging protocol for self-driving vehicles. In 2017 ACM/IEEE 8th International Conference on Cyber-Physical Systems (ICCPS), pp. 219-228. IEEE.
- Deb, K., (2014). Multi-objective optimization. In E. Burke, G. Kendall (eds.) *Search methodologies,* Springer: Boston, MA., pp. 403-449.
- Frank, M., (1981). The braess paradox. Mathematical Programming, 20(1), pp.283-302.
- Kojima, A., Elfferding, S. and Kubota, H., (2015). Intelligent rat-runners: impact of car navigation systems on safety of residential roads. *International journal of intelligent transportation systems research*, **13**(1), pp. 9-16.
- Mehta, H., Kanani, P. and Lande, P., (2019). Google Maps. *Int. J. Comput. Appl*, 178, pp.41-46.
- Petrovska, N., Stevanovic, A., (2015, September) "Traffic Congestion Analysis Visualisation Tool," 2015 IEEE 18th International Conference on Intelligent Transportation Systems, Las Palmas, 2015, pp. 1489-1494, doi: 10.1109/ITSC.2015.243.

– Sanders, P. and Schultes, D., (2007, June). Engineering fast route planning algorithms. In *International Workshop on Experimental and Efficient Algorithms* (pp. 23-36). Springer, Berlin, Heidelberg.

– Salazar, M., Paccagnan, D. and Agazzi, A., 2020. Urgency-aware optimal routing in repeated games through artificial currencies. arXiv preprint arXiv:2011.11595.

– Suarez, F. F. (2003) Battles for technological dominance: an integrative framework. *Research Policy*, **33**(2), pp. 271-286.

– Steinberg, R. and Zangwill, W.I., (1983). The prevalence of Braess' paradox. *Transportation Science*, **17**(3), pp. 301-318.

– Russell, S.J. and Norvig, P., (2010). Artificial Intelligence-A Modern Approach, Third International Edition.

– Vanderhaegen, F., Chalmé, S., Anceaux, F. and Millot, P., (2006). Principles of cooperation and competition: application to car driver behavior analysis. *Cognition, Technology & Work*, **8**(3), pp. 183-192.

– Vinitsky, E., Kreidieh, A., Le Flem, L., Kheterpal, N., Jang, K., Wu, C., Wu, F., Liaw, R., Liang, E. and Bayen, A.M., (2018, October). Benchmarks for reinforcement learning in mixed-autonomy traffic. In *Conference on Robot Learning*, pp. 399-409.

– Wu, C., Bayen, A.M. and Mehta, A., (2018, May). Stabilizing traffic with autonomous vehicles. In 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE. pp. 1-7.