Increasing User Trust in Optimisation through Feedback and Interaction

JIE LIU, Monash University
KIM MARRIOTT, Monash University
TIM DWYER, Monash University
GUIDO TACK, Monash University

User trust plays a key role in determining whether autonomous computer applications are relied upon. It will play a key role in the acceptance of emerging AI applications such as optimisation. Two important factors known to affect trust are system transparency, i.e. how well the user understands how the system works, and system performance. However, in the case of optimisation it is difficult for the end-user to understand the underlying algorithms or to judge the quality of the solution. Through two controlled user studies we explore whether the user is better able to calibrate their trust in the system when: (a) they are provided feedback on the system operation in the form of visualisation of intermediate solutions and their quality; (b) they can interactively explore the solution space by modifying the solution returned by the system. We found that showing intermediate solutions can lead to over-trust while interactive exploration leads to more accurately calibrated trust.

CCS Concepts: • **Human-centered computing → Human computer interaction (HCI)**; Visualization; • **Applied computing → Multi-criterion optimisation and decision-making**.

Additional Key Words and Phrases: HCI, Interactive Optimisation, Human-in-the-loop Optimisation, Trust, Feedback, Vehicle Routing

ACM Reference Format:
Jie Liu, Kim Marriott, Tim Dwyer, and Guido Tack. 2018. Increasing User Trust in Optimisation through Feedback and Interaction. 1, 1 (December 2018), 35 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Trust plays a critical role in determining user reliance on automated systems [30]. As a consequence there has been considerable research into trust and the factors affecting it [30, 38, 61]. Too much or too little trust are equally dangerous. For instance under-trust of the ship's navigation system by the captain may have led to the Costa Concordia running aground in 2012 while over-trust is believed to have contributed to the crash of a Turkish Airlines flight in 1951 [30]. The Goldilock's trust dilemma is, therefore, how do we design computer applications that engender exactly the right amount of trust, not too much, not too little.

Early research focussed on trust of real-time monitoring and control systems such as those used for power-plant monitoring or flight monitoring and control, e.g. [53]. With the arrival

Authors' addresses: Jie Liu, jliu120@gmail.com, Monash University, 900 Dandenong Rd, Caulfield East, Victoria, 3145; Kim Marriott, kim.marriott@monash.edu, Monash University, 900 Dandenong Rd, Caulfield East, Victoria, 3145; Tim Dwyer, tim.dwyer@monash.edu, Monash University, 900 Dandenong Rd, Caulfield East, Victoria, 3145; Guido Tack, guido.tack@monash.edu, Monash University, 900 Dandenong Rd, Caulfield East, Victoria, 3145.

XXXX-XXXX/2018/12-ART $15.00
https://doi.org/10.1145/1122445.1122456

of the internet the focus was on trust in on-line systems, e.g. [14] and most recently there has been considerable interest in trust for emerging AI applications such as autonomous vehicles [28], robots [60] or medical assistance devices [28] since it is clear that adoption of these new technologies depends upon building appropriate trust [64].

One important but often overlooked field of AI is optimisation. In general, optimisation techniques aim to find the best solution to a given problem by modelling it mathematically using *constraints* (which dictate the valid solutions) and an *objective function* (which measures the quality of the solution) and then using constrained optimisation techniques to find the solution. Mathematicians and AI researchers have been developing faster and more powerful techniques for solving constrained optimisation problems since the 1950s. As a result optimisation is now routinely used in various application areas including transport and logistics, timetabling, production scheduling and the design of modern energy systems. But there is considerable scope for more widespread use of optimisation software, and a major factor limiting its use is a lack of trust in automated optimisation systems [47].

Given this, it is surprising there has been virtually no empirical research into user trust of optimisation systems. The main contribution of this paper is to address this significant gap through two controlled studies investigating how feedback about solver progress and interactive manipulation of the solution affect trust in optimisation systems. Hoff and Bashir identify three kinds of factors affecting trust: *dispositional*, *situational* and *learned trust* [30]. We focus on learned trust as this takes into account the user's understanding and experience with the actual system. It is therefore the only category affected by the system design and interface.

One important factor affecting learned trust is whether the user understands the algorithm being used by the system and believes that it is capable of achieving their goals [38]. Unfortunately, state-of-the-art optimisation software is very complex, often non-deterministic, and typically works on an internal representation of the problem that is difficult even for an expert in optimisation to understand. Thus, it is infeasible to present the detailed algorithm execution in terms of the actual internal representation. However, we hypothesised that trust in the system would be increased by providing high-level feedback on progress towards the final (near-)optimal solution in a way that could be understood by the user.

We investigated this in our first study (Section 4). Participants were asked to rate their trust in two solvers. Each solver was shown with or without feedback in the form of displaying interim solutions and their associated objective values as the system proceeded toward the final solution. Such feedback allows the user to see the range of solutions being considered and their relative quality in terms of the objective function. We constructed the two solvers to produce solutions of different quality; the *good* solver returned near-optimal solutions, and the *poor* solver returned solutions 30% worse than those of the good solver. As hypothesised, feedback led to significantly greater trust in the poor solver (but not the good solver). Indeed we found that there was unwarranted trust in the poor solver with feedback.

But the main factor affecting learned trust is the user's evaluation of the system *performance*. We wanted to allow the user to be able to evaluate the quality of the system's performance in a way that allows them to discriminate between various levels of performance [38]. Unfortunately, it can be very difficult for the typical user to evaluate the quality of a solution produced by an optimisation system for a real-world problem. Unless the solution has some obvious faults it is very difficult to know if there is another solution that improves the objective. The problem is that the *solution space (or fitness landscape)* is typically disjoint and/or the objective function is non-continuous or non-convex. It is also not feasible to compute and display more than a fraction of the solution space. A possible approach is to use a visual-analytics-based approach in which the user can interactively explore and visualise the solution space around the solution returned by the

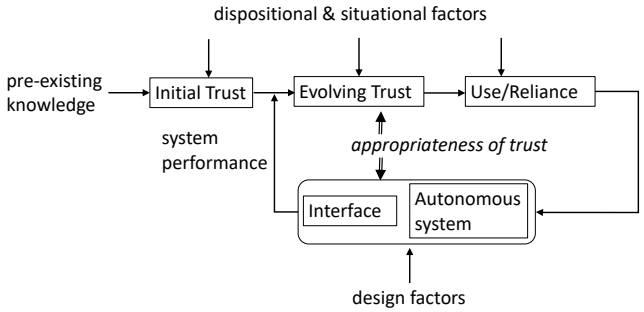

Fig. 1. High-level framework for understanding the factors that influence trust and reliance on autonomous systems. It incorporates the three-layered trust framework of Hoff and Bashir [30] into the framework of Lee and See [38].

system. Indeed some optimisation researchers have previously suggested (without experimental evidence) that interaction leads to increased trust [47].

In our second study (Section 5) we investigated this. We examined whether allowing the user to interactively manipulate the solution—either completely manually or with the help of a semi-automatic re-solve capability—enables the user to more accurately assess the quality of the solver. Our hypothesis was that interaction with the solver would lead to better assessment of solver quality. This study used an additional *medium* solver which returned solutions 15% worse than those of a good solver. Each solver was shown in three conditions: no interactive manipulation of solutions, manual interaction and interaction with re-solve. As hypothesised we found that interaction led to significantly better calibration of trust.

The two studies were conducted using a purpose-built optimisation tool for routing and scheduling of electrician service provision. The underlying problem is a variant of the *Vehicle Routing Problem*, a well-studied optimisation problem [66] that is known to be challenging in practice. The system utilised a state-of-the-art optimisation solver and featured a carefully designed visual representation of a problem instance and solution (Section 3).

The two user studies we have presented are the first we know of to explicitly explore in a controlled setting the impact of the design of the optimisation system on user trust. As such they are only a first step and more experimentation is required. Nonetheless our results do have implications for the design of optimisation systems. They strongly suggest that optimisation systems should allow the user to interactively manipulate solutions returned by the system. This allows the user to better evaluate their quality and so calibrate and build their trust in optimisation system.

~~Our results provide new insight into the factors affecting trust. Most interestingly, the results suggest that, for AI applications such as optimisation and machine learning (which utilises optimisation) in which it is difficult for the user to evaluate solution quality, users will be better able to judge and calibrate their trust if the tool allows them to interactively manipulate solutions returned by the system so that they can better evaluate their quality.~~

2 BACKGROUND

2.1 Trust in Automation

Over the past four decades there have been hundreds of papers written about trust in automation. Parasuraman and Riley [53] pointed out that over-trust and under-trust are equally dangerous

because they may cause misuse or disuse of automation systems. Dzindolet *et al.* [20] investigated the link between trust and automation reliance and concluded that trust plays an important role in decisions about automation reliance. It found that explaining the reasons when an automation system makes mistakes resulted in increased trust and thereby increased automation reliance. Herlocker *et al.* [29] also found that explanations can build trust and increase the acceptance of a recommendation system. Recently, Abdul *et al.* [1] emphasised the need to build more transparent intelligent systems to help users understand the behind-the-scenes decision making to gain trust. Most recently, Roy *et al.* [59] suggested that a well designed human-in-the-loop automation system that allows manual rectification after automation may result in higher user satisfaction.

A number of researchers have suggested general frameworks for understanding trust. The following review is based on Lee and See's influential theoretical framework [38] and the three-layered trust framework of Hoff and Bashir [30]. Figure 1 shows the combined framework.

Lee and See [38] define *trust* to be *the attitude that an agent, i.e. computer system, will help to achieve a user's goals in a situation characterised by uncertainty and vulnerability.* Trust is based on the user's beliefs and their intentions and actions, such as degree of use and *reliance* on the system. *Calibration* refers to the correspondence between the system's capabilities and the level of trust in the system: *over-trust* occurs when trust exceeds the capabilities while *distrust* occurs when capabilities exceed the level of trust.

In Lee and See's framework, trust is based on information about the system as well as individual, organisational and cultural context. They identified *performance*, *process* and *purpose* as the general basis for trust. *Performance* refers to *what* the system does: its ability to achieve the user's goals. A user will tend to trust a system if it has performed well in the past. *Process* refers to how the system operates: the degree to which the system's algorithms are appropriate for the situation. A user will tend to trust a system if they understand its algorithms and believe they are appropriate to the goals in the current situation. *Purpose* refers to *why* the system was developed. A user will tend to trust a system if it is being used within the realm of the designer's intent.

Hoff and Bashir [30] present a three-layered model of trust arranged around: *dispositional*, *situational* and *learned trust*. Dispositional trust refers to a user's overall tendency to trust automation, independent of context or the specific system. Factors include culture, age, gender and personality type. Situational trust takes into account the context in which the system is used [13]. Factors such as workload, difficulty of task and associated risk, as well as user self-confidence, mood, and expertise in the application domain affect both trust and reliance. *Learned trust* refers to the user's evaluation of the actual system. It depends upon the user's pre-existing knowledge and system performance. They distinguish between *initial* and *dynamic* learned trust. Pre-existing information, such as: system reputation; prior experience with similar automated systems; as well as knowledge about the purpose and the algorithms being used, all affect initial trust. On the other hand, numerous studies show that users adjust their trust based on the system's performance, e.g. [5, 19]. As also discussed by Schaefer *et al.* [61], reliability, validity of result, predictability, usefulness, dependability, and the kind and seriousness of errors are all important.

Both Lee and See and Hoff and Bashir discuss how perception of the system crucially depends upon the design of the system and its user interface. Many studies have found that the content and format of the interface affect credibility and trust, e.g. [8, 23, 35, 41]. Lee and See conclude that trust tends to increase if the interface provides concrete details that are consistent and clearly organised. Hoff and Bashir suggest that increasing saliency of automated feedback can increase trust. Ease-of-use [25], level of control [68], and communication style also affect trust. They also identify *transparency* of automation as a factor affecting trust and recommend providing accurate, useful feedback on the system's operation. Similarly, Lee and See recommend to: *"show the process*

and algorithms of the automation by revealing intermediate results in a way that is comprehensible to the operators."

The above high-level framework is based on user-studies of trust in a number of different application domains. Early research focussed on trust of real-time monitoring and control systems [37, 48, 50, 53]. With the arrival of the internet the focus was on trust in on-line systems. Corritore *et al.* [14] describe a model for online trust based on perception of website risk, credibility and ease of use. Other researchers have investigated models of trust for online shopping [25, 39], cyberdomains [31], recommendation systems [15, 27], adaptive agent systems [27], and information security classification [55]. More recently, AI applications such as autonomous vehicles [28], robots [60], medical assistance devices [28] and machine learning [34, 58] have received attention.

2.2 Explainable AI

Recently, there has been considerable interest in *explainable artificial intelligence (XAI)*. This aims to provide human-understandable explanations of AI systems to increase users' understanding and trust of AI systems. Ribeiro *et al.* [58] found, for instance, that providing explanations allows users to calibrate their trust in machine learning classifiers. Specifically, users' trust dropped substantially when the explanation for the bad classifier was revealed. Explainable AI approaches fall into two main categories: *transparent models* and *post-hoc explainability*. According to Lipton [42], the purpose of transparent models is to open the black-box of the AI model so that users can understand how the model works. More specifically, there are three levels of model transparency:

- **Simulatability:** refers to the ability of the underlying AI model to be simulated by a user: this requires the model to be relatively simple.
- **Decomposability:** denotes the ability to decompose the model and explain each part of it, such as input and parameters without the need for any other tools.
- **Algorithmic transparency:** emphasises the ability to allows users to understand the model's behaviours for producing outputs.

For more complex AI models, post-hoc explainability provides alternative approaches to improve the interpretability of the model. They are:

- **Text explanations:** provide users with explanations of the results from the model using natural language text and symbols.
- **Visual explanations:** utilise visualisations to facilitate the understanding of the model's behaviour. It is typically used in conjunction with other techniques to improve users' understanding, especially for non-expert users.
- **Local explanations:** produce explanations for a subset of the model rather than presenting the behaviour of the whole model.
- **Explanations by example:** extract representative examples to give users a better demonstration of how the model works, similar to how we explain a process, by using specific examples.

Later on, post-hoc explainability has been extended by Arrieta *et al.* [3] to include another two kinds of explanations:

- **Explanations by simplification:** describe a simpler new system that is essentially equivalent to the original. Because of the relative simplicity, the new system is easier to interpret.
- **Feature relevance explanations:** show an indirect explanation of a model by quantifying the relevance of input variables in relation to the output. Comparing the relevance scores gives the user insights into the importance of different variables.

2.3 Trust in optimisation

Most work on explainable AI has focused on explaining the results of machine learning (ML) for classification, prediction or recommendation. While there are strong links between optimisation and ML–in particular optimisation is often used for ML–and similarities in that both typically use complex algorithms that make it difficult even for experts to understand and interpret the results there is one significant difference. In ML the optimisation problem is often abstract and ill-defined at least from the point of view of a non-expert user so it is difficult for them to judge performance. In many optimisation applications, however, an end-user who has knowledge of the problem domain but is not an optimisation expert has the ability to judge performance. In particular, while the end-user may find it hard to know if a solution is a global optimum they can readily judge, at least for problems with a well-defined objective, if one solution is better than another. Thus ML and optimisation have potentially different characteristics when it comes to explainability and trust.

~~Despite the recognition that user trust is vital in building acceptance of optimisation, there has been virtually no research in this area. While some researchers have conjectured that interactive optimisation will increase trust~~ [47], ~~user studies evaluating interactive optimisation systems have focussed on solution quality and time spent to find a solution rather than trust~~ [2, 6, 9, 10, 17, 33, 54, 62, 63, 65].

A number of researchers have investigated explainable optimisation, e.g. [18, 52]. However, to the best of our knowledge there have not been empirical evaluations of how explainable optimisation affects user trust. The other main suggestion for increasing user trust in optimisation algorithms is through user *interaction*. While the most common reasons for providing interaction are to allow the user to tailor the constraint problem or to guide the search for a better solution, it has also been conjectured that interaction may increase trust [47]. However, user studies evaluating interactive optimisation systems have focused on solution quality and time spent to find a solution rather than trust [2, 6, 9, 10, 17, 33, 54, 62, 63, 65].

Meignan *et al.* [47] provide a summary of the main kinds of interaction provided in interactive optimisation systems:

- **Trial and error:** The simplest approach is simply to allow the user to adjust the existing constraints, objectives and/or the parameters of the optimisation solver and then re-run the optimisation from the beginning.
- **Interactive re-optimisation:** This ranges from simply allowing the user to manually modify the solution and see the impact to true re-optimisation in which the user makes changes to the current solution and then the solution is re-optimised without overwriting the previous user-specified changes.
- **Interactive multi-objective optimisation:** This aims at balancing the trade-offs between different conflicting objectives.
- **Interactive evolutionary algorithms:** In this case, the user subjectively evaluates solutions and the underlying optimisation systems apply evolutionary algorithms to continuously improve and evolve solutions.
- **Human-guided search:** This allows users to guide the optimisation search process in order to improve search efficiency.

The only study we are aware of that has empirically investigated trust in optimisation is by Liu *et al.* [43] who conducted a small qualitative study with 8 oncology professionals to evaluate a new interactive optimisation technique for brachytherapy seed placement for prostate cancer treatment. They found some evidence that the participants gained trust through interactive optimisation in a treatment protocol that was unfamiliar to most participants (focal brachytherapy) but little evidence that interaction built trust in the solver as opposed to the treatment protocol. Furthermore, the

Fig. 2. Study 1 interface. The *objective line chart* is shown only in the *feedback* condition.

study was small-scale, difficult to generalise to other applications, and did not tease apart which aspects of the tool engendered trust or whether the increased trust was warranted. In a follow-up paper Liu *et al.* [44] gives 9 design recommendations for interactive optimisation systems. A user study evaluated these with reference to improving the quality of the solution but did not consider user trust.

Thus, the two studies presented here significantly extend our understanding of how to engender appropriate trust in optimisation systems. ~~They also add to a more general understanding of trust because—unlike most other applications—it is very difficult for the end-user to understand the optimisation system (i.e. it is not transparent) or to judge system performance.~~

3 EXPERIMENTAL SYSTEM DESIGN

In both of our studies participants were presented with a vehicle routing problem with time windows [16, 21, 36, 67]. We chose this problem because it is easy to understand and explain to non-expert users but still a difficult optimisation problem. It was also used in [44]. The scenario presented to the participants was that of a company that sends electricians to customers. Each customer requires a certain fixed amount of time for the service, and has a time window when that service needs to happen. All electricians start from a central depot and return to the depot after servicing all their customers. During each experiment participants were given different instances of this problem. In each instance the locations of the customers, their service times (time required to complete the job at that customer) and time windows, the location of the depot and the number of electricians were

given. The goal of the optimisation was to find a schedule allocating customers to electricians and specifying the order in which the customers are visited so that all time constraints are respected and the overall distance travelled is minimised.

In Study 1 we investigated the impact of providing feedback on the solvers progress towards the final solution. Most constraint solving methods work by iteratively finding better solutions to the problem. We therefore chose to display the current solution each time the solver found a better solution together with the associated objective function value. This is a way of increasing *algorithmic transparency* [42], allowing the user to "see" into the optimisation black-box.

In Study 2, we investigated the effect of interactive re-optimisation [46] on users' trust. We believed that this might provide a kind of *local explanation* [42] by allowing the user to explore the solution space around the solution returned by the solver. We provided both a purely manual modification of the solution without re-optimisation and modification with limited re-optimisation. We provided two conditions because allowing the user to manually modify the solution and immediately see the impact on the quality of the solution is straightforward to implement with any constraint solving algorithm while true re-optimisation requires a solver that can support this.

We pre-computed the solutions before the study so as to control for the quality of solutions returned by the solvers in the study. More precisely:

- **Off-line solver:** We solved each problem scenario off-line before conducting the user studies, using state-of-the-art constraint solving technology (the problems were modelled in the MiniZinc constraint modelling language [51] and solved with Gecode [24] as the back-end algorithm). Each instance was run for up to 30 minutes and the best solution as well as any sub-optimal intermediate solution found in that time was recorded. Afterwards, for each of the best solutions found, we ran the solver again, adding constraints to limit solution quality to 15% worse, and then 30% worse, compared to the best solution found. These additional solutions were used to simulate different solver qualities while maintaining full control over the experimental conditions.
- **On-line solver:** In the experiments users are presented with two or three different "on-line" solvers: a *good* solver, a *poor* solver and a *medium* solver. In reality, all three on-line solvers were simulated based on the results computed off-line. The *good* solver is the best solution found in the off-line computation; the *medium* solver shows the 15% worse solution, and the *poor* solver condition uses the 30% worse solution. This allowed us to closely control the user experience and to make sure it was consistent between participants, i.e. to keep the apparent solve time constant across participants, solver condition and problem instance.

3.1 Visualisation of Routing Schedule as used in Study 1

We took considerable care developing a browser-based visual interface for use in the two studies that provided participants with readily understandable information about a scheduling solution and its quality. In Study 1 we defined two conditions: *feedback* and *non-feedback*. The interface (see Fig. 2) provided three different view panels with an additional panel for the condition with *feedback*. Colour is used to distinguish individual electricians, and is consistent across all views [56].

The following views were used in both conditions:

- **Map:** Shows the location of the home depot, customer locations and the route for each electrician. The implementation uses the OpenStreetMap[1] on-line map resource and the Leaflet[2] JavaScript library to create overlay visuals.

[1] www.openstreetmap.org
[2] www.leafletjs.com

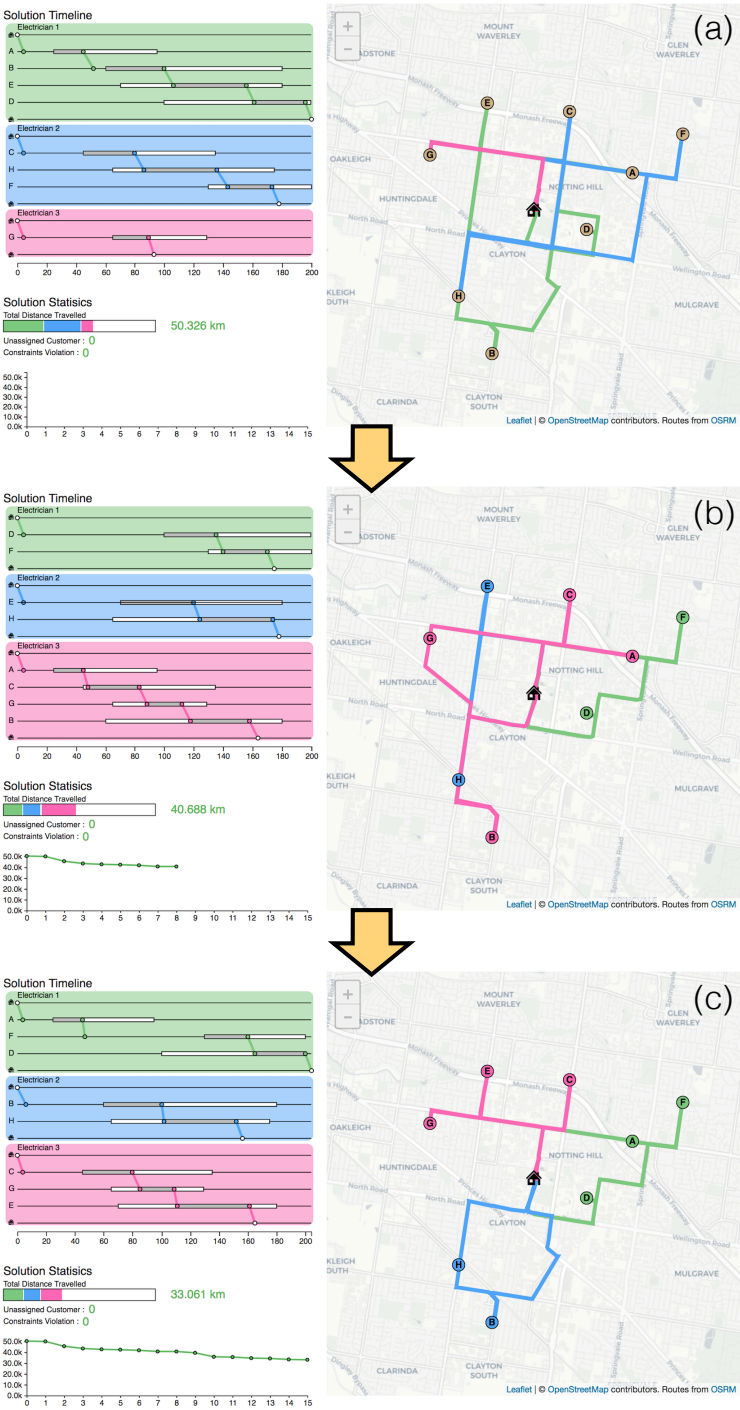

Fig. 3. Study 1 interface with feedback on solver progress. (a) first solution; (b) an intermediate solution; (c) final solution.

- **Schedule:** Each electrician's vehicle is represented by its own track in a *faceted timeline* [7]. We chose a horizontal layout of the timelines to better resemble a Gantt Chart, a commonly used representation for schedules. Each letter represents a single customer. An electrician's schedule for service delivery to a customer is represented as a bar with two parts: grey and white. The grey part indicates the actual period for the electrician to deliver the service to the customer, whereas the white part represents the customer's time constraints for the service delivery.
- **Solution statistics:** The objective value to be minimised is the total distance travelled by all electricians. We presented the total distance numerically as well as graphically in a small, colour-coded, stacked-bar chart showing a breakdown of distance travelled by each electrician.

The following panel was only shown in the *feedback* condition of Study 1.

- **Feedback on solver progress:** The solver that we used outputs a solution every time a configuration with an objective lower than the previous best solution is reached. We showed each of these interim solutions in an animated sequence, together with an *objective line chart* that graphed the tour length for each of the interim solutions. We believed that showing interim solutions would help users to understand the operation of the solver and its exploration of the search space and that showing the objective line chart would help them understand the solver's progress towards the eventual solution. We also considered showing the search space or displaying the fitness landscape but felt that end-users would find these less meaningful and harder to understand. Fig. 3 shows an example of the feedback view. We scaled and adjusted the minimum value of the y-axis of this chart such that the slope of the line was similar across all instances and solvers, with the vertical decline taking up about $\frac{1}{3}$ of the vertical range. We did this because piloting revealed that users were quick to judge solver quality based on line slope.

Interaction The interface provided limited interaction:

- Brushing a route in either map or schedule highlights the route in the other view.
- Hovering over a time window shows an infobox of precise time window details.

3.2 Interactive Version as Used in Study 2

In Study 2, we had three conditions: *No Interaction (NI)*, *Manual Interaction (MI)* and *Semi-automatic Interaction (SI)*.

- **Manual modification:** In both MI and SI we introduced a new interaction allowing users to modify the solution by dragging the lines representing customers, either to change the delivery order within one electrician's tour, or to reassign a customer to a different electrician.
- **Re-optimise:** In the SI condition, we also introduced interactive optimisation in the form of a "re-optimise button" to perform a local optimisation of the customer visit ordering for an individual electrician. Since the number of possible permutations of customer order for one electrician is relatively small, we were able to do this optimally using a simple complete search algorithm running in the browser.

The interface is shown in Fig. 4 and an example of a participant exploring different solutions in the SI condition is shown in Fig. 5. The map, schedule and solution statistics views were the same as in Study 1, supporting the same basic brushing and hovering interactions, and were provided in all three conditions.

In the MI and SI conditions, to support interactive exploration of the various user-generated solutions, we provided a new histogram view of the objective function for different solutions. The

Fig. 4. Study 2 interface. The *objective histogram* is shown in both *SI* and *MI* conditions. The *re-optimise button* is only available in *SI* condition. The button turns off (greys out) when it is not possible to improve the solution by reordering the order in which customers are visited by the associated electrician. The button turns on when the there is a better solution with a shorter total distance.

objective histogram replaced the line chart view of objectives used in the *feedback* condition of Study 1 to provide a more suitable interface for interaction, as follows:

- Because we no longer show the interim solutions from the solver, a linear connection of points no longer makes sense. That is, we only show final solutions from the solver and solutions after each user interaction (including infeasible solutions).
- The bars of the objective histogram provide a larger click target for time-travelling through solutions compared to the small circles of the objective line chart.
- Some of the solutions may be infeasible as a result of user interaction. The bars corresponding to such solutions are indicated in red.

Thus, the histogram allows participants to interrogate the *provenance* of each solution [57].

4 STUDY 1: EFFECT OF SOLVER FEEDBACK ON TRUST

The first user study examined whether feedback on solver progress affected user evaluation of solution quality and trust in the solver. **We hypothesised that it would increase trust.** Participants were asked to compare their trust in a *good* solver returning near-optimal solutions and a *poor* solver returning solutions 30% worse than those of the good solver. Each solver was shown in two

Fig. 5. Study 2: an example of exploring solutions in the *SI* condition. (a) feasible and worse solutions; (b) an infeasible solution; (c) a feasible and better solution.

Mixed Designs	Between-subjects Variables		Within-subjects Variables			
	Expertise		Feedback		Solver Quality	
Levels	Expert	Non-expert	[with] Feedback	No Feedback	Good Solver	Poor Solver

Fig. 6. Study 1 mixed designs overview.

conditions: *non-feedback* and *feedback* during execution showing (simulated) partial solutions and quality of solution as discussed above.

4.1 Participants and Setting

We recruited 28 participants: students, researchers from our institution and a few employees from other organisations. We had 15 male participants and 13 female participants. 22 aged between 20 and 29, the remaining 6 participants were aged between 30 and 39. All participants had normal or corrected-to-normal vision and were without any colour vision impairment. We carefully balanced the participants so that we had exactly 14 experts and 14 non-experts. Participant expertise was determined based on the answers to a short questionnaire given at the very beginning of the experiment. Participants were considered as experts if they were either: at least somewhat familiar with optimisation; or, they had previously encountered vehicle routing problems. The study was run on a PC equipped with an Intel i5-6600K 3.5 GHz processor and a 27-inch screen (1920 × 1080). Participants were equipped with a Tobii X3-120 eye tracker at the beginning of the study.

4.2 Tasks and Design

For each problem instance, participants were asked to evaluate the quality of the solution returned by the solver (scale 1–7).

After evaluating all solutions produced by the same optimisation solver, participants were asked to evaluate the optimisation solver using different measures of trust (scale 1–7). The trust measures were based on those used in the trust literature but the questions were slightly reworded to fit an optimisation context. This did not change the definitions or structures of the original measurements. They consisted of a single-item measurement of overall trust plus six single-item measures of different components of trust as identified by prior research. A similar model was used in [37]. ~~Single-item measures of trust have been found to be reliable.~~ Christophersen and Konradtn [12] used a four-item questionnaire to evaluate trust in online stores. These four items are all included in our trust measurement. McKnight and Chervany [45] provided five categorised characters to define trust including competence, predictability, benevolence, integrity and other (such as shared understanding). All five characters are essentially reflected in our measurement but competence is mapped to functionality, and integrity is mapped to dependability and faith. The exact questions and their source in past work is as follow:

> **Functionality**[4, 12, 32, 45, 61]–the competence of the optimisation solver to solve the problem: "To what extent does the optimisation solver perform its function properly?"
> **Understanding**[26, 45, 58, 61]–user understanding of the solver's operation: "How well do you understand the strategy used by the solver to find this solution?"
> **Dependability**[4, 12, 32, 37, 45, 49, 61]–how reliable the solver is: "To what extent can you count on the optimisation solver to do its job?"
> **Consistency**[4, 37, 45, 49]–how consistently the solver performs on different problems: "To what extent does the optimisation solver perform consistently?"

Satisfaction[26, 32, 45, 61]–acceptability of the solutions: "How satisfied are you with the performance of the optimisation solver?"
Faith[12, 37, 45, 49]–confidence of the solvers' performance in solving future problems: "What degree of faith do you have that the optimisation solver will be able to cope with future problems?"
Trust[12, 32]–the overall degree of trust in the solver: "Overall, how much do you trust the optimisation solver?"

This experiment was a mixed design with one between-subjects variable and two within-subjects variables (see Fig. 6). Specifically, the between-subjects variable was *expertise* with two levels: experts and non-experts. The two within-subjects variables were *feedback* and *solver quality*. Feedback had two levels: feedback and no feedback. Solver quality also had two levels: good solver and poor solver.

We generated different problem instances by fixing the central depot but randomly selecting customer locations around the central depot. Customers' service times as well as time windows were assigned randomly but manually refined afterwards by trying to overlap close-by customers' time windows to control difficulty of problem instances.

For each combination of solver and feedback we showed the solution to 4 problem instances. These had different levels of difficulty, varying in the number of customers and electricians. We presented 2 *easy* problem instances, and then 2 *hard* instances, with the intent of allowing users to build their understanding of the solver. After piloting, we chose easy problem instances to have 8 customers and 3 electricians and hard problem instances to have 15 customers and 4 electricians.

Thus, in the experiment we had 28 participants $\times$ 4 optimisation solvers $\times$ (4 solutions $\times$ 1 solution evaluation question + 7 solver evaluation questions) = 1,232 responses (44 responses per participant).

4.3 Procedure

After answering a short questionnaire capturing demographic information and expertise, participants were guided through the calibration process of the screen-based eye tracker and instructed to start screen recording. Both calibration and recording were done using the Tobii Pro Studio software.

Participants were then *trained* in the use of the system. Training was designed to ensure that all participants thoroughly understood the goal of the optimisation process and could evaluate the quality of a solution returned by a solver. A short video provided a brief introduction to the experimental problem and interface. The video could be paused or replayed at any time and participants were encouraged to ask questions. Afterwards a hands-on exercise was given in which participants were asked to develop a routing solution to a problem with 5 customers and 2 electricians using drag and drop to assign customers to electricians. Participants were then shown a sample solution with 5 questions to further test their understanding of the experiment's problem context. Next participants were asked to inspect a *good* solution and a *poor* solution to a problem. At the end of *training* participants were asked to inspect 8 sample questions (1 solution evaluation question + 7 solver evaluation questions). They were informed that the 8 questions would be exactly the same during the *experiment* and they were encouraged to clarify their understanding.

Next, during the *experiment* participants were asked to evaluate the four optimisation solvers. For each solver they were shown the sample solutions on a separate page and asked to rate their quality. Participants were then asked to answer the 7 questions evaluating trust in the solver. A 7-point Likert scale was used to capture answers to both evaluation tasks. Latin square design

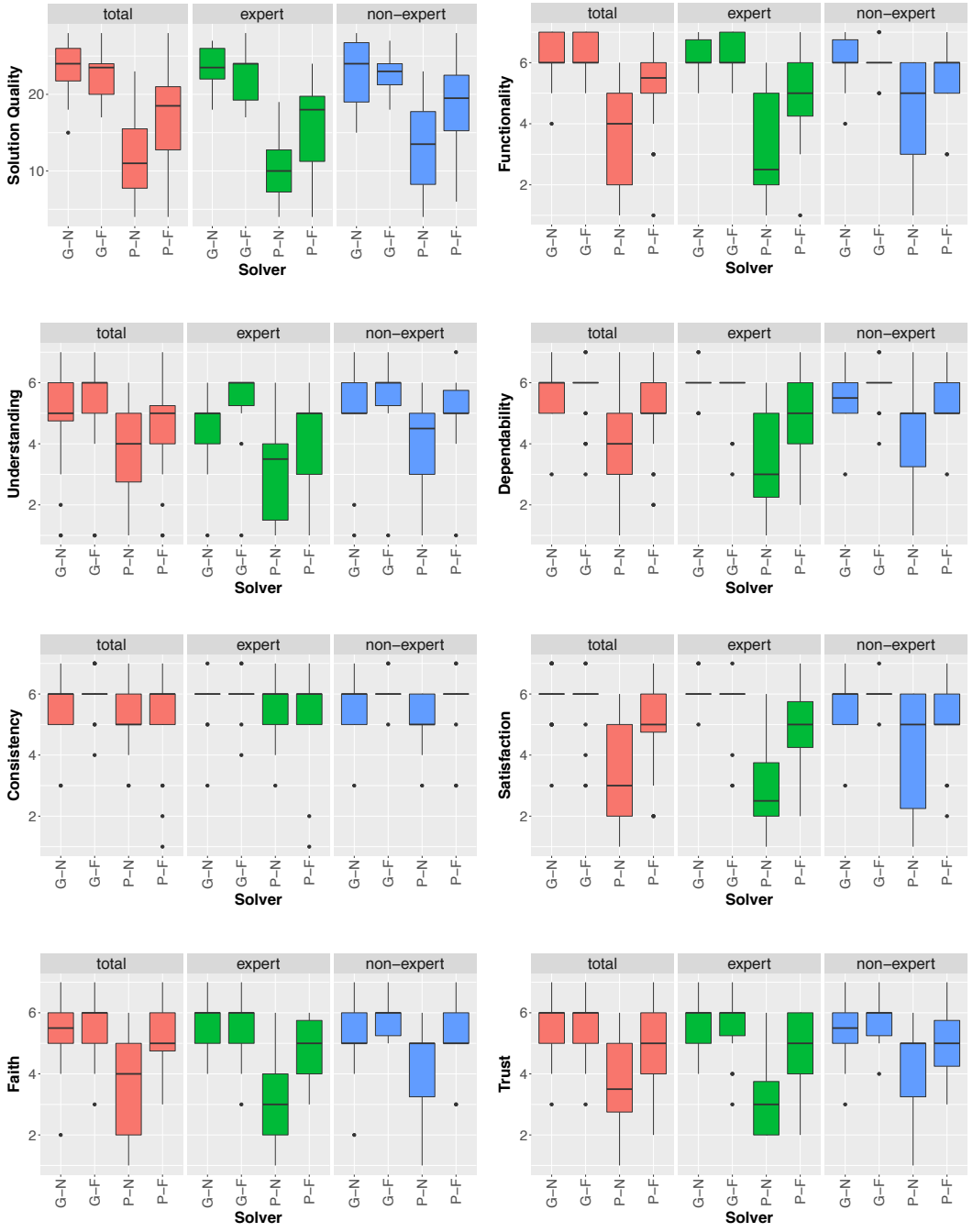

Fig. 7. Study 1 aggregate *solution quality*, solver *functionality*, *understanding*, *dependability*, *consistency*, *satisfaction*, *faith* and overall *trust* measures. For the x-axis labels the first letter represents the solver quality: *G – Good*; *P – Poor* and the second letter represents the feedback option: *N – Non-feedback*; *F – Feedback*. So G–N represents the good solver without feedback.

was used to balance the order of solvers. The recording was stopped after participants finished all evaluations.

Finally, a post-questionnaire was administered. Responses were audio recorded. The questionnaire had eight questions. The first question asked participants to explain their evaluation process. The second question clarified whether different strategies were used for evaluating easy and hard problems. The next four questions asked about the usefulness of both the objective line chart and the display of intermediate solutions in animated sequences for easy and hard problems. The next question asked for feedback about the interface design. The last question asked participants how useful they found the easy and hard problems for solver evaluations.

4.4 Data Analysis and Discussion

Quantitative Analysis

The ratings of the four solutions shown for each solver were summed to give an aggregate *solution quality* for each solver. This rating as well as the seven solver measures for the two solvers with and without feedback are shown in Fig. 7.

Residuals of *solution quality* were normally distributed (visually checked with histogram and Q—Q plots). The variances of the participants' expertise were equal (Levene's test). The residuals of the seven individual measures were normally distributed except *consistency*, however, using a Box–Cox transformation *consistency* was corrected and normally distributed. Therefore we used a three-way mixed ANOVA with multilevel linear models to analyse the impact solver feedback and expertise had on *solution quality* and the other seven measures. Simple effects analysis was performed using linear mixed-effects models if any significant interaction was found. Otherwise we conducted Tukey's HSD post hoc tests on significant main effects [22]. ~~The statistical analysis is shown in Fig. 8.~~ Details of the statistical analysis can be found in the appendix.

The key findings are as follows:

- The analysis found that participants ranked solution quality, functionality, dependability, faith and trust significantly higher with feedback than without for the poor solver but not for the good solver.
- The analysis found that participants ranked understanding significantly higher with feedback regardless of solver.
- Experts ranked satisfaction significantly higher with feedback than without only for the poor solver. The satisfaction ratings from non-experts were similar but not significant.
- The only significant difference between experts and non-experts was that non-experts ranked functionality significantly higher than experts for the poor solver but not for the good solver.

Qualitative Analysis

We collected qualitative feedback from the post-questionnaire at the end of the study asking participants specifically about the usefulness of the objective line chart: 21 out of 28 (75%) participants (9 experts and 12 non-experts) thought the objective line chart is useful, 5 out of 28 (17.86%) participants (4 experts and 1 non-expert) thought it is not that useful and they did not use it during the study. All the remaining 2 (7.14%) participants (1 expert and 1 non-expert) were neutral about its usefulness.

Some participants thought the objective line chart was a good indication that the optimisation solver was functioning properly. One participant said:

> "Of course, it (the objective line chart) is [useful]. I can see that it is decreasing always. Always a good sign at least. It's a sign that the optimisation is going in the right direction."

Another participant commented:

> "It is clearly dropping down, which convinces me that the solver is doing its work properly."

Some participants thought the objective line chart helped solution evaluation and comparison. One stated:

> "Providing more information to help me verify and decide [the quality of a solution]."

Another one said:

> "Yes, it is useful because we can visualise the distance and compare it with the other solutions."

Other participants believed the objective line chart helped them build up confidence about the underlying algorithm. One participant told us:

> "It helps me to develop a lot of confidence about the algorithm by looking at that [objective line chart]."

Another participant also confirmed:

> "Having a progress bar is definitely provides me with [a] certain confidence."

Eye-tracking Analysis

Eye-tracking data was collected from 24 of the 28 participants. Data was not collected for the remaining 4 participants because they completed the study at another location without the eye tracker. Eye-tracking revealed that 18 out of the 24 (75%) participants spent on average more than 4 seconds looking at the objective line chart for each problem instance. While this was only a small part of the time spent evaluating the quality of a solution for each problem instance, on average nearly 100 seconds, this confirms that most participants did look at the objective line chart. We believe the short duration most likely reflects the simplicity of the chart, allowing it to be comprehended in a short glance.

Discussion

The most important finding of the analysis is that adding feedback significantly increased participants' rating of the poor solver for *solution quality* and for *functionality*, *dependability*, *satisfaction*, *faith* and overall *trust*. However, feedback did not significantly affect rating of the good solver: this is perhaps due to a ceiling effect (participants gave high ratings to all criteria for the good solver). This generally supports our hypothesis that providing feedback about intermediate solutions will increase user trust in an optimisation system. This hypothesis is also supported by the qualitative analysis with most participants finding the objective line chart useful and their comments suggesting that showing the improvement of solution quality in intermediate solutions during the search process helped to build their confidence in the solver.

Our findings are in accord with Ribeiro *et al.* [58] who reported that providing explanations of a machine learning classifier was useful in understanding the performance of the classifier. Such explanations can also be regarded as a form of feedback. Using feedback to increase the transparency of a system is critical in building trust in the system [26], especially in high-consequence domains, such as medical treatment [11, 40].

What was unexpected was that providing feedback led to an unwarranted increase in the level of trust for the poor solver. Examination of Figure 11 reveals that without feedback participants recognised the superior performance of the good solver. Once feedback was provided they marginally increased their ranking of the good solver but greatly increased their ranking of the poor solver to the point that they gave almost identical rankings to the poor and good solver with feedback.

In fact, many participants rated the *solution quality* and overall *trust* of the poor solver with feedback as equal to, or even higher, than the rating they gave for the good solver with and without

Mixed Designs	Between-subjects Variables					Within-subjects Variables		
	Condition			Expertise		Solver Quality		
Levels	Semi-automatic Interaction (SI)	Manual Interaction (MI)	No interaction (NI)	Expert	Non-expert	Good Solver	Medium Solver	Poor Solver

Fig. 8. Study 2 mixed designs overview.

feedback. From our observations of participants' ratings, 6 out of 28 participants (3 experts and 3 non-experts) believed the poor solver with feedback produced solutions at least as good as the good solver with feedback. Whereas only 2 out of 28 participants (both non-experts) thought the poor solver without feedback produced solutions as good as or better than the good solver without feedback. Furthermore, 13 out of 28 participants (6 experts and 7 non-experts) trusted the poor solver with feedback more than the good solver with feedback and 13 out of 28 participants (not exactly the same participants, 5 experts and 8 non-experts) also trusted the poor solver with feedback more than the good solver without feedback. In contrast, only 5 participants (1 expert and 4 non-experts) trusted the poor solver without feedback more than the good solver without feedback. This is surprising, and shows that providing feedback on intermediate solutions can lead to over-trust even by experts.

This suggests that we need to be cautious about providing feedback to increase system transparency and user trust in optimisation tools. In such tools it may be difficult for the user of the system to easily evaluate the quality of a solution provided by the system and so feedback can lead to over-trust in a poorly performing system. Therefore, a better approach for providing correctly calibrated trust in an optimisation system may be to provide the user with a way to more readily evaluate the quality of solutions produced by the solver. In the next study, we explore whether allowing the user to interactively modify solutions as a way of better understanding their quality will lead to more accurately calibrated trust.

5 STUDY 2: EFFECT OF INTERACTION ON TRUST

The second user study investigated whether trust is affected by allowing the user to interact with a solution in order to better understand its quality. We used an additional *medium* solver which returned solutions 15% worse than those of the good solver. Each solver was shown in three conditions: no interactive manipulation of solutions (*No Interaction (NI)*); fully manual interaction (*Manual Interaction (MI)*); and interaction with re-solve (*Semi-automatic Interaction (SI)*). Because of the number of conditions a between-participant design was used. The tasks and protocol were the same as the first study.

5.1 Participants and Setting

We recruited 30 participants in total including students, researchers from universities and employees from outside organisations. All 30 participants had normal or corrected-to-normal vision without any colour vision impairment. 22 participants were males, and the other 8 participants were females. 25 participants were aged 20 to 29, the other 5 participants were aged 30 to 39. As in the first study we distinguished between experts and non-experts. We divided the 15 expert and 15 non-expert participants equally between the three different conditions: *no*, *semi-automatic* and *manual* interaction. In the end, we had 5 experts and 5 non-experts in each condition. The study was run on a MacBook Pro notebook with a 2.6 GHz Intel i5 processor and a 13-inch screen (1280×800).

We did not collect eye-tracking data but took a screen recording of each experiment.

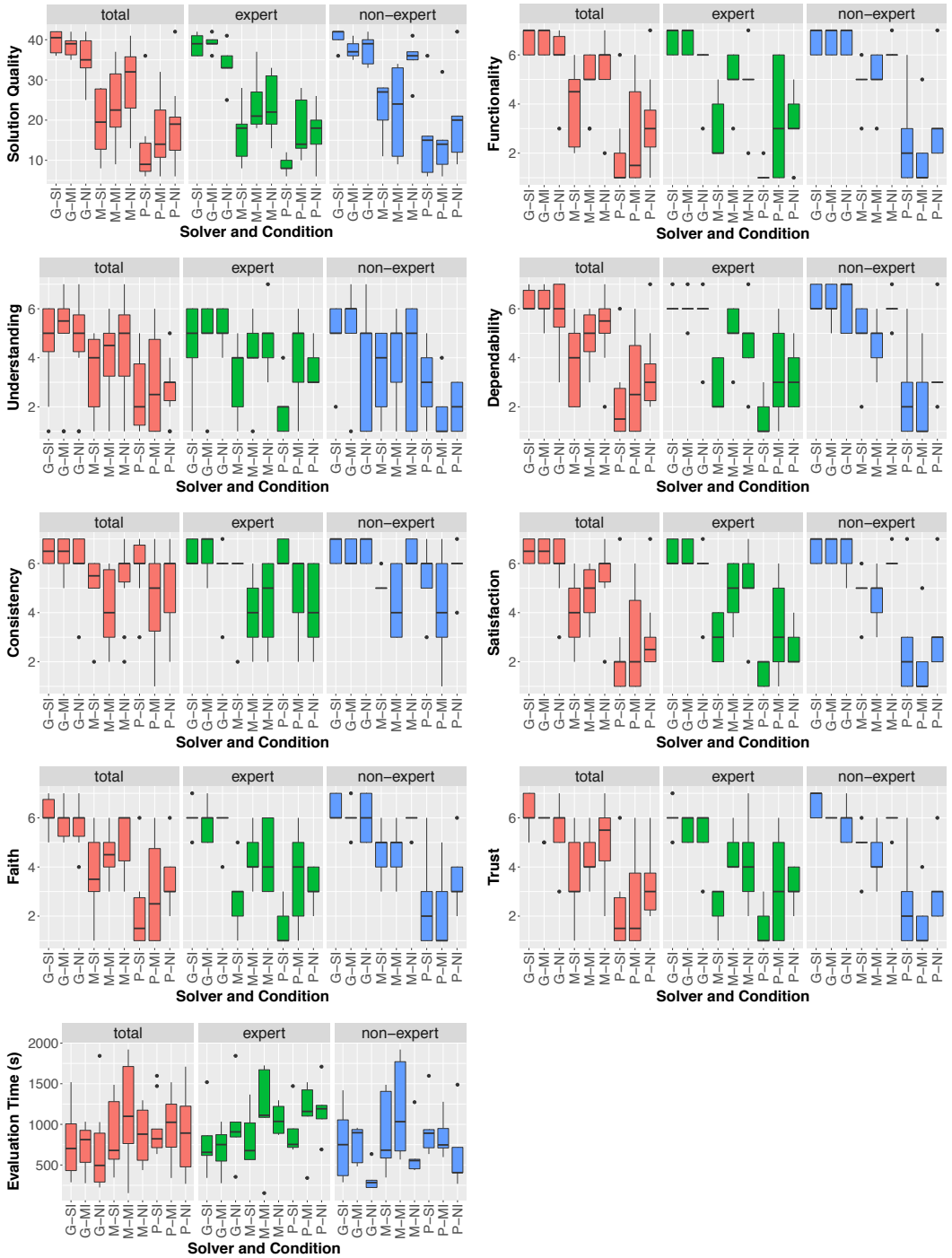

Fig. 9. Study 2 aggregate *solution quality*, total solution evaluation time (in seconds) and the seven solver measures. For the x-axis labels the first letter represents solver quality: *G – Good*; *M – Medium*; *P – Poor* and the second double letters represent experimental condition: *SI – Semi-automatic Interaction*; *MI – Manual Interaction*; *NI – No Interaction*. So G–SI represents the good solver in the semi-automatic interaction condition.

5.2 Data and Design

We used the same approach as in Study 1 to generate different problem instances. However, we added one more customer to the *easy* problem instances to make it more challenging for the interaction in this study. We did not change the difficulty of *hard* problem instances because they were difficult enough to evaluate based on our observations from Study 1. Again *easy* problem instances were presented before *hard* instances.

The *training* was extended to explain solution manipulation to participants in the *semi-automatic* and *manual* interaction conditions at the start of the hands-on exercises. They were encouraged to use interaction freely during the *experiment*.

A post-questionnaire asking participants about the effect of interaction on trust was administered to those conditions providing solution interaction. Eight questions were asked for both the SI and MI conditions. The first two questions asked participants about the evaluation processes of solutions and solvers respectively. The third question aimed to clarify whether different strategies were used to evaluate easy and hard problems. The next question asked participants about the usefulness of manually changing a solution to evaluate its quality. The following two questions focused on whether manually changing a solution could increase participants' confidence and trust in solutions and solvers. The last two questions asked for general feedback on the interface and the usefulness of easy and hard problems for solver evaluations. For the SI condition, three additional questions were included. Participants were asked about the usefulness of re-optimising a solution as well as whether re-optimisation could increase the confidence and trust in solutions and solvers.

Participants' responses were timed and a maximum of 5 minutes was allowed for each problem instance, to control the overall experiment time.

Similar to Study 1, this experiment also used mixed designs (see Fig. 8). More specifically, the between-subjects variables were *condition* and *expertise*. Condition had three levels: Semi-automatic Interaction (SI), Manual Interaction (MI) and No Interaction (NI). Expertise had two levels: experts and non-experts. The single within-subjects variable was *solver quality* with three levels: good solver, medium solver and poor solver.

This experiment was also between- and within-subjects. We had 30 participants × 3 optimisation solvers × (2 difficulty levels × 3 repetitions × 1 solution evaluation question + 7 solver evaluation questions) = 1,170 responses (39 responses per participant).

5.3 Data Analysis and Discussion

Quantitative Analysis

The ratings for the 6 solutions shown for each solver were summed to give an aggregate *solution quality* for each solver (range 6–42). The time to evaluate the 6 solutions was summed to give a total *evaluation time* for each solver. These, together with the seven solver rating measures (as per Study 1) for the three solvers and three conditions are shown in Fig. 9.

Residuals of the aggregate *solution quality* were normally distributed (visually checked with histogram and Q—Q plots). Residuals for the total *evaluation time* were not normally distributed. Therefore, a log transformation was used to correct residuals to follow a normal distribution. Variances of the experimental conditions were equal for both aggregate *solution quality* and total *evaluation time* (Levene's test). The residuals of the seven solver rating measures were normally distributed except *faith*, however, using a Box–Cox transformation *faith* measures were corrected and normally distributed. Both *faith* and *trust* measures violated the homogeneity assumption with Levene's test, however, conducting Welch's tests on both, we were able to correct for the violations.

We used a three-way mixed ANOVA with multilevel linear models to analyse aggregate *solution quality*, total *evaluation time* and the other seven solver measures. As in the first experiment, simple

effects analysis was performed using linear mixed-effects models if any significant interaction was found. Otherwise we conducted Tukey's HSD post hoc tests on significant main effects [22]. ~~The statistical analysis is shown in Fig. 11.~~ Details of the statistical analysis can be found in the appendix.

The key findings are as follows:

- The analysis found that participants ranked both solution quality and trust significantly higher in the semi-automatic interaction condition than the no interaction condition for the good solver.
- For the medium solver, however, participants ranked solution quality, functionality, satisfaction and trust significantly lower in the semi-automatic interaction condition than the no interaction condition.
- The analysis found that participants ranked solution quality, functionality, satisfaction and trust significantly higher for the good solver compared with the poor solver and for the medium solver compared with the poor solver in all three interaction conditions: semi-automatic, manual and no interaction conditions.
- The analysis found that participants ranked solution quality, functionality, satisfaction and trust significantly higher for the good solver than the medium solver in both the semi-automatic and manual interaction conditions but not in the no interaction condition.
- Non-experts spent significantly longer time evaluating solutions in both the semi-automatic and manual interaction conditions than the no interaction condition. However, there was no significant difference for experts.

Qualitative Analysis

Qualitative feedback from the post-questionnaire administered to the 20 participants from the *semi-automatic* and *manual* interaction conditions, revealed that 17 out of 20 (85%) participants (8 experts and 9 non-experts) thought manually changing a solution via interaction can increase trust in good optimisation solvers. The remaining 3 participants (2 experts and 1 non-expert) were neutral about its effect on trust.

Most participants indicated that their trust increased because they could not find a better solution by manipulating the solution returned by the solver. One participant said:

> "If the solver gives a good first impression and I don't find any improvements [to the solution] after I test and verify my own, I am more sure that this solver is really good and I trust it more even though there might still exist better solutions."

Another participant commented:

> "Because you are still sceptical about it (the solution) and want to try it yourself, you realise you cannot actually find anything better [after you tried] which makes you believe this is a good solver."

While a third participant told us:

> "Verifying makes you confident. Confidence leads to trust."

Nearly all of the participants (9 out of 10, 4 experts and 5 non-experts) from the *semi-automatic* interaction condition believed that re-optimisation was useful. They felt that re-optimisation made it easier, faster and more reliable to evaluate the quality of a solution. The same 9 participants also believed that re-optimisation could increase trust in good optimisation solvers. The remaining participant was neutral about the usefulness and the effect of re-optimisation on trust.

We also analysed the impact of re-optimisation on whether participants could find a better solution. We first consider the *semi-automatic* interaction conditions. For solutions produced by the

good solver, there was only one example of a participant finding a better solution using re-optimisation. The distance was shortened by 3.8%. For the medium solver, participants were able to find better solutions for 3 out of 6 problem instances. On average 0.7 better solutions were found using re-optimisation, with the distance shortened by 7.7%. Whereas for the poor solver, participants found better solutions for 5 out of 6 problem instances. Two were found using re-optimisation. Distances were shortened by 10.5% on average.

On the other hand with *manual* interaction condition, no participant found a better solution for the good solver and on average only 2.4 out of 6 solutions from the medium solver were improved by participants. This is slightly less than for the semi-automatic condition. And for the poor solver, on average 3.6 out of 6 solutions were improved in the manual interaction condition. Again, this is less than for the semi-automatic condition. Overall, this accords with the participant feedback and suggests that the semi-automatic interaction condition with the re-optimisation allows the user to more readily discover if a solution is poor than with the manual interaction condition.

Discussion

Our results support our hypothesis that allowing interactive modification of the solution will lead to better calibrated trust. We see that both the semi-automatic and manual interaction conditions helped participants to distinguish between the good and medium solver while in the no interaction condition this was more difficult. For instance, none of the ~~30~~ 10 participants trusted the medium solver more than the good solver in the semi-automatic interaction condition. However, there were 5 participants (1 expert and 4 non-experts) who trusted the medium solver over the good solver in the no interaction condition.

However, if the solver is sufficiently bad then regardless of whether interaction is allowed, people will generally recognise the poor solution quality and not trust it. Nonetheless, while none of the participants trusted the poor solver over the good solver in the semi-automatic condition there were 2 participants (1 expert and 1 non-expert) who rated the poor solver higher than the good solver in the no interaction condition.

Our findings are explained by the fact that trust is dynamic, and it can be earned via verification [26]. At the beginning all solvers are trusted equally. Trust increases when the user cannot find a way to improve the solution returned by the solver, but decreases when they can. Participant feedback indicates that interaction makes it easier for the users to verify the solution. We would also expect that re-optimisation makes it easier than purely manual interaction to find when a solution is sub-optimal. This explains why semi-automatic interaction leads to the greatest trust in the good solver, and the least trust in the medium and poor solvers, while no interaction leads to the lowest trust in the good solver and greatest trust in the poor and medium solvers.

In general our findings are in accord with those of Glass *et al.* [26] who investigated user trust in adaptive agents. They found that "trust is an earned property" and that users would prefer to supervise the system and interact with the system in a mixed-initiative manner in order to verify its behaviour. This is a good reflection of our findings: the ability to interactively verify solution quality is a powerful way of building trust.

~~In related work, Ribeiro *et al.* [58] found that providing explanations allows users to calibrate their trust in machine learning classifiers. Specifically, users' trust dropped substantially when the explanation for the bad classifier was revealed. In Ribeiro's experiment, domain experts had the necessary knowledge to verify the correctness of a classification result from its explanation. However, producing an explanation as to why a particular solution is optimal in interactive optimisation is more difficult and such an explanation may not be understood by the end-user.~~

Here we have shown that allowing the user to modify the solution returned by the solver and immediately see the quality of the solution can support solver verification as it allows the user to

check that they cannot find a better solution. In a sense, by allowing the user to explore the local neighbourhood of the solution, it provides a local explanation [42] of the local optimality of the solution.

This technique works because the user has a good understanding of the optimisation problem being solved and so feels confident in understanding how to modify a solution and in comparing the quality of two solutions. Thus it seems applicable to optimisation applications where the user has reasonable knowledge of the problem domain and there is a well-defined objective function. This means that it is less applicable to multi-criteria objective problems and for evaluating machine-learning applications even when they use optimisation as it is unlikely that the user will have sufficient knowledge to manipulate a solution or to compare the quality of two solutions.

Much previous work on increasing trust in optimisation systems has focused on providing textual explanations of why a solution is optimal. We believe that our alternative approach of allowing the user to interactively modify a solution and the result has two advantages. Supporting manual modification is relatively simple and does not require solver modification while creating explanations is more complex and may not work with all solving techniques. Secondly, we conjecture that users will find the results of the manipulation easier to understand than a textual explanation though this needs to be confirmed.

6 LIMITATIONS AND FUTURE WORK

One potential limitation of our two studies was the definition of expertise. An expert had both familiarity with optimisation and familiarity with vehicle routing problems, potentially confounding these two kinds of expertise. To check for this we re-ran the analysis for both studies using a single factor definition by only considering either familiarity with optimisation or familiarity with vehicle routing problems. This led to only minor differences and did not change any findings we have identified and reported. Nonetheless, the expertise factor is self-reported and therefore subjective.

A limitation of our studies is that there was no consequence to our participants of mistakenly trusting a solution. That is they were not "vulnerable." While this is a limitation of many HCI experiments investigating trust, e.g. [20, 59] it does mean that, at least in situations in which the consequences for the user of getting it wrong are high, our experimental findings may not carry over. Further studies are required to explore this.

A further limitation was the relatively small sample size in each of the conditions.

We also only considered a single kind of optimisation problem. While it was chosen to be representative of the kind of resource allocation problems that optimisation is commonly used for, it would be useful to verify the results with other kinds of optimisation problems. In particular it would be interesting to consider optimisation problems with a multi-criteria objective.

Future work also includes examining other possible factors affecting trust. Our two studies could necessarily only look at a limited range of feedback and user interaction. As discussed earlier, we would expect a user's understanding of the algorithm and belief that it is capable of achieving their goals to impact on trust. Indeed our first study suggests that this is true. We would also like to investigate whether a high-level explanation of how the solver works builds trust or whether providing interactive visualisations to show the exploration of the search space can further improve the transparency of an optimisation system and hence increase trust. It also seems fruitful to explore the impact of other kinds of user interaction. For instance, allowing the user to check the performance of the solver on simpler problems and verify that it finds the answer they expect. It would also be interesting to see the impact of interactive multi-objective optimisation and interactive evolutionary algorithms on user trust.

7 CONCLUSION

We have presented two controlled user studies investigating two important factors affecting user trust in optimisation systems. In our first user study we found that providing feedback about intermediate solutions and the objective function leads to increased trust in a poor solver producing low-quality solutions. In fact, we found that it can lead to over-trust. Specifically, providing feedback leads many people, including experts, to increase trust in a poor solver to such a degree that they trust it as much as a good solver that consistently produces high quality solutions.

In our second study we found that allowing the user to semi-automatically manipulate solutions returned by an optimisation system leads to a better calibration of trust. This is an important finding because it provides the first empirical support for the belief by some optimisation researchers [47] that interaction leads to greater trust.

Our results have significant implications for the design of new optimisation systems. They strongly suggest that if optimisation systems are to be trusted by users, then implementors will have to move away from the current "black-box" model, in which the user simply inputs the problem data and accepts the output solution. Instead, systems should support interactive exploration of solutions, allowing the user to gain a better understanding of their quality, and hence build users' (justified) trust in the system. Our results also suggest that optimisation systems should be careful if providing information about intermediate solutions and progress to the final solution as this may lead to unwarranted trust in the solver.

~~One limitation of the two studies is that they only considered a single kind of optimisation problem. While it was chosen to be representative of the kind of resource allocation problems that optimisation is commonly used for, it would be useful to verify the results with other kinds of optimisation problems. It would also be interesting to consider optimisation problems with a multi-criteria objective.~~

~~Future work also includes examining other possible factors affecting trust. As discussed earlier, we would expect a user's understanding of the algorithm and belief that it is capable of achieving their goals to impact on trust. Indeed our first study suggests that this is true. We would also like to investigate whether a high-level explanation of how the solver works builds trust or whether providing interactive visualisations to show the exploration of the search space can further improve the transparency of an optimisation system and hence increase trust.~~

ACKNOWLEDGEMENTS

We acknowledge the support of CSIRO and Data 61 (formerly NICTA) which is funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre for Excellence Program.

REFERENCES

[1] Ashraf Abdul, Jo Vermeulen, Danding Wang, Brian Y Lim, and Mohan Kankanhalli. 2018. Trends and trajectories for explainable, accountable and intelligible systems: An hci research agenda. In *Proceedings of the 2018 CHI conference on human factors in computing systems*. 1–18.

[2] David Anderson, Emily Anderson, Neal Lesh, Joe Marks, Brian Mirtich, David Ratajczak, and Kathy Ryall. 2000. Human-guided simple search. In *AAAI/IAAI*. 209–216.

[3] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. 2020. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion* 58 (2020), 82–115.

[4] Donovan Artz and Yolanda Gil. 2007. A survey of trust in computer science and the semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web* 5, 2 (2007), 58–71.

[5] Nathan R Bailey and Mark W Scerbo. 2007. Automation-induced complacency for monitoring highly reliable systems: the role of task complexity, system experience, and operator trust. *Theoretical Issues in Ergonomics Science* 8, 4 (2007), 321–348.

[6] Gilles Bailly, Antti Oulasvirta, Timo Kötzing, and Sabrina Hoppe. 2013. Menuoptimizer: Interactive optimization of menu systems. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*. ACM, 331–342.

[7] Matthew Brehmer, Bongshin Lee, Benjamin Bach, Nathalie Henry Riche, and Tamara Munzner. 2017. Timelines revisited: A design space and considerations for expressive storytelling. *IEEE transactions on visualization and computer graphics* 23, 9 (2017), 2151–2164.

[8] Pamela Briggs, B Burford, and C Dracup. 1998. Modelling self-confidence in users of a computer-based system showing unrepresentative design. *International Journal of Human-Computer Studies* 49, 5 (1998), 717–742.

[9] Eric Brochu, Tyson Brochu, and Nando de Freitas. 2010. A Bayesian interactive optimization approach to procedural animation design. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Eurographics Association, 103–112.

[10] Keith A Butler, Jiajie Zhang, Chris Esposito, Ali Bahrami, Ron Hebron, and David Kieras. 2007. Work-centered design: A case study of a mixed-initiative scheduler. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 747–756.

[11] Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. 2015. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 1721–1730.

[12] Timo Christophersen and Udo Konradt. 2011. Reliability, validity, and sensitivity of a single-item measure of online store usability. *International Journal of Human-Computer Studies* 69, 4 (2011), 269 – 280.

[13] Marvin S Cohen, Raja Parasuraman, and Jared T Freeman. 1998. Trust in decision aids: A model and its training implications. In *in Proc. Command and Control Research and Technology Symposium*. Citeseer.

[14] Cynthia L Corritore, Beverly Kracher, and Susan Wiedenbeck. 2003. On-line trust: concepts, evolving themes, a model. *International journal of human-computer studies* 58, 6 (2003), 737–758.

[15] Dan Cosley, Shyong K Lam, Istvan Albert, Joseph A Konstan, and John Riedl. 2003. Is seeing believing? How recommender system interfaces affect users' opinions. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 585–592.

[16] Teodor G Crainic and Gilbert Laporte. 2012. *Fleet management and logistics*. Springer Science & Business Media.

[17] Mary L Cummings, Jessica J Marquez, and Nicholas Roy. 2012. Human-automated path planning optimization and decision support. *International Journal of Human-Computer Studies* 70, 2 (2012), 116–128.

[18] Kristijonas Čyras, Dimitrios Letsios, Ruth Misener, and Francesca Toni. 2019. Argumentation for explainable scheduling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 2752–2759.

[19] Ewart de Visser and Raja Parasuraman. 2011. Adaptive aiding of human-robot teaming: Effects of imperfect automation on performance, trust, and workload. *Journal of Cognitive Engineering and Decision Making* 5, 2 (2011), 209–231.

[20] Mary T Dzindolet, Scott A Peterson, Regina A Pomranky, Linda G Pierce, and Hall P Beck. 2003. The role of trust in automation reliance. *International journal of human-computer studies* 58, 6 (2003), 697–718.

[21] Nasser A El-Sherbeny. 2010. Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods. *Journal of King Saud University-Science* 22, 3 (2010), 123–131.

[22] Andy Field, Jeremy Miles, and Zoë Field. 2012. *Discovering Statistics using R*. Sage publications.

[23] BJ Fogg and Hsiang Tseng. 1999. The elements of computer credibility. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 80–87.

[24] Gecode Team. 2016. Gecode: A Generic Constraint Development Environment. http://www.gecode.org

[25] David Gefen, Elena Karahanna, and Detmar W Straub. 2003. Trust and TAM in online shopping: An integrated model. *MIS quarterly* 27, 1 (2003), 51–90.

[26] Alyssa Glass, Deborah L McGuinness, and Michael Wolverton. 2008. Toward establishing trust in adaptive agents. In *Proceedings of the 13th international conference on Intelligent user interfaces*. ACM, 227–236.

[27] Jason L Harman, John O'Donovan, Tarek Abdelzaher, and Cleotilde Gonzalez. 2014. Dynamics of human trust in recommender systems. In *Proceedings of the 8th ACM Conference on Recommender systems*. ACM, 305–308.

[28] Monika Hengstler, Ellen Enkel, and Selina Duelli. 2016. Applied artificial intelligence and trust?The case of autonomous vehicles and medical assistance devices. *Technological Forecasting and Social Change* 105 (2016), 105–120.

[29] Jonathan L Herlocker, Joseph A Konstan, and John Riedl. 2000. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*. 241–250.

[30] Kevin Anthony Hoff and Masooda Bashir. 2015. Trust in automation: Integrating empirical evidence on factors that influence trust. *Human Factors* 57, 3 (2015), 407–434.

[31] Robert R Hoffman, John D Lee, David D Woods, Nigel Shadbolt, Janet Miller, and Jeffrey M Bradshaw. 2009. The dynamics of trust in cyberdomains. *IEEE Intelligent Systems* 6 (2009), 5–11.
[32] Jiun-Yin Jian, Ann M Bisantz, and Colin G Drury. 2000. Foundations for an empirically determined scale of trust in automated systems. *International Journal of Cognitive Ergonomics* 4, 1 (2000), 53–71.
[33] Ashish Kapoor, Bongshin Lee, Desney Tan, and Eric Horvitz. 2010. Interactive optimization for steering machine classification. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1343–1352.
[34] Been Kim. 2015. *Interactive and interpretable machine learning models for human machine collaboration.* Ph.D. Dissertation. Massachusetts Institute of Technology.
[35] Jinwoo Kim and Jae Yun Moon. 1998. Designing towards emotional usability in customer interfaces?trustworthiness of cyber-banking system interfaces. *Interacting with Computers* 10, 1 (1998), 1–29.
[36] Suresh Nanda Kumar and Ramasamy Panneerselvam. 2012. A survey on the vehicle routing problem and its variants. *Intelligent Information Management* 4, 03 (2012), 66.
[37] John Lee and Neville Moray. 1992. Trust, control strategies and allocation of function in human-machine systems. *Ergonomics* 35, 10 (1992), 1243–1270.
[38] John D Lee and Katrina A See. 2004. Trust in automation: Designing for appropriate reliance. *Human Factors: The Journal of the Human Factors and Ergonomics Society* 46, 1 (2004), 50–80.
[39] Matthew KO Lee and Efraim Turban. 2001. A trust model for consumer internet shopping. *International Journal of Electronic Commerce* 6, 1 (2001), 75–91.
[40] Benjamin Letham, Cynthia Rudin, Tyler H McCormick, David Madigan, et al. 2015. Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. *The Annals of Applied Statistics* 9, 3 (2015), 1350–1371.
[41] Yung-Ming Li and Yung-Shao Yeh. 2010. Increasing trust in mobile commerce through design aesthetics. *Computers in Human Behavior* 26, 4 (2010), 673–684.
[42] Zachary C Lipton. 2018. The Mythos of Model Interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue* 16, 3 (2018), 31–57.
[43] Jie Liu, Tim Dwyer, Kim Marriott, Jeremy Millar, and Annette Haworth. 2018. Understanding the Relationship between Interactive Optimisation and Visual Analytics in the Context of Prostate Brachytherapy. *IEEE Transactions on Visualization and Computer Graphics* 24, 1 (2018), 319–329.
[44] Jie Liu, Tim Dwyer, Guido Tack, Samuel Gratzl, and Kim Marriott. 2020. Supporting the problem-solving loop: Designing highly interactive optimisation systems. *IEEE Transactions on Visualization and Computer Graphics* 27, 2 (2020), 1764–1774.
[45] D Harrison McKnight and Norman L Chervany. 2000. What is trust? A conceptual analysis and an interdisciplinary model. *AMCIS 2000 proceedings* (2000), 382.
[46] David Meignan, Jean-Marc Frayret, and Gilles Pesant. 2011. An interactive heuristic approach for the P-forest problem. In *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*. IEEE, 1009–1013.
[47] David Meignan, Sigrid Knust, Jean-Marc Frayret, Gilles Pesant, and Nicolas Gaud. 2015. A Review and Taxonomy of Interactive Optimization Methods in Operations Research. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 5, 3 (2015), 17.
[48] Neville Moray, Toshiyuki Inagaki, and Makoto Itoh. 2000. Adaptive automation, trust, and self-confidence in fault management of time-critical tasks. *Journal of Experimental Psychology: Applied* 6, 1 (2000), 44.
[49] Bonnie M Muir. 1987. Trust between humans and machines, and the design of decision aids. *International Journal of Man-Machine Studies* 27, 5-6 (1987), 527–539.
[50] Bonnie M Muir and Neville Moray. 1996. Trust in automation. Part II. Experimental studies of trust and human intervention in a process control simulation. *Ergonomics* 39, 3 (1996), 429–460.
[51] Nicholas Nethercote, Peter J. Stuckey, Ralph Becket, Sebastian Brand, Gregory J. Duck, and Guido Tack. 2007. MiniZinc: Towards a Standard CP Modelling Language. In *Principles and Practice of Constraint Programming – CP 2007*, Christian Bessière (Ed.). Springer Berlin Heidelberg, 529–543.
[52] Barry OâĂŹCallaghan, Barry OâĂŹSullivan, and Eugene C Freuder. 2005. Generating corrective explanations for interactive constraint satisfaction. In *International Conference on Principles and Practice of Constraint Programming*. Springer, 445–459.
[53] Raja Parasuraman and Victor Riley. 1997. Humans and automation: Use, misuse, disuse, abuse. *Human factors* 39, 2 (1997), 230–253.
[54] James Patten and Hiroshi Ishii. 2007. Mechanical constraints as computational constraints in tabletop tangible interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 809–818.
[55] Wolter Pieters. 2011. Explanation and trust: What to tell the user in security and AI? *Ethics and information technology* 13, 1 (2011), 53–64.

[56] Zening Qu and Jessica Hullman. 2018. Keeping multiple views consistent: Constraints, validations, and exceptions in visualization authoring. *IEEE transactions on visualization and computer graphics* 24, 1 (2018), 468–477.
[57] Eric D Ragan, Alex Endert, Jibonananda Sanyal, and Jian Chen. 2016. Characterizing provenance in visualization and data analysis: an organizational framework of provenance types and purposes. *IEEE transactions on visualization and computer graphics* 22, 1 (2016), 31–40.
[58] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1135–1144.
[59] Quentin Roy, Futian Zhang, and Daniel Vogel. 2019. Automation Accuracy Is Good, but High Controllability May Be Better. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–8.
[60] Kristin E Schaefer, Deborah R Billings, James L Szalma, Jeffrey K Adams, Tracy L Sanders, Jessie Y Chen, and Peter A Hancock. 2014. *A meta-analysis of factors influencing the development of trust in automation: Implications for human-robot interaction.* Technical Report. ARMY RESEARCH LAB ABERDEEN PROVING GROUND MD HUMAN RESEARCH AND ENGINEERING DIRECTORATE.
[61] Kristin E Schaefer, Jessie YC Chen, James L Szalma, and Peter A Hancock. 2016. A meta-analysis of factors influencing the development of trust in automation: Implications for understanding autonomy in future systems. *Human factors* 58, 3 (2016), 377–400.
[62] Stacey D Scott, Neal Lesh, and Gunnar W Klau. 2002. Investigating human-computer optimization. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 155–162.
[63] Mitsuhiro Shibuya, Hajime Kita, and Shigenobu Kobayashi. 1999. Integration of multi-objective and interactive genetic algorithms and its application to animation design. In *Systems, Man, and Cybernetics, 1999. IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on*, Vol. 3. IEEE, 646–651.
[64] Keng Siau and Weiyu Wang. 2018. Building Trust in Artificial Intelligence, Machine Learning, and Robotics. *Cutter Business Technology Journal* 31, 2 (2018), 47–53.
[65] Christian Thieke, Karl-Heinz Küfer, Michael Monz, Alexander Scherrer, Fernando Alonso, Uwe Oelfke, Peter E Huber, Jürgen Debus, and Thomas Bortfeld. 2007. A new concept for interactive radiotherapy planning with multicriteria optimization: First clinical evaluation. *Radiotherapy and Oncology* 85, 2 (2007), 292–298.
[66] Paolo Toth and Daniele Vigo (Eds.). 2001. *The Vehicle Routing Problem.* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.
[67] Paolo Toth and Daniele Vigo. 2002. *The vehicle routing problem.* SIAM.
[68] Frank MF Verberne, Jaap Ham, and Cees JH Midden. 2012. Trust in smart systems: Sharing driving goals and giving information to increase trustworthiness and acceptability of smart systems in cars. *Human Factors* 54, 5 (2012), 799–810.

A STATISTICAL ANALYSIS

A.1 Study 1: Effect of Feedback on Trust

We follow the conventional statistical results reporting: if a higher-order interaction is significant, we interpret the interaction using contrasts. We do not and should not interpret its main effects and lower-order interactions. We only interpret the main effects when an interaction effect is not significant [22].

There are two main paths in Fig. 10. Specifically, the first path was *part 1* → *part 2* when interactions were significant. Significant interaction effects were reported in part 1, a simple effect analysis was performed and presented in part 2 to further interpret each significant interaction. The other path was *part 3* → *part 4* when interactions were not significant but main effects were significant. Significant main effects are presented in part 3, and a follow-up post hoc Tukey's HSD is presented in part 4, allowing us to interpret each main effect. When both interaction and main effects were not significant, they were not included in Fig. 10 and were excluded from analysis.

Understanding. The main effect of feedback was significant on how well participants believed they understood the solver, $\chi^2(1) = 11.36, p = .0007$ (see Fig. 10, part 3). Tukey's HSD tests revealed significant differences between the feedback condition and the no-feedback condition, $z = 3.61, p = 0.0003$ (see Fig. 10, part 4, outline C; Fig. 11). This tells us that participants believe they have a better understanding of how the solver works when feedback is provided.

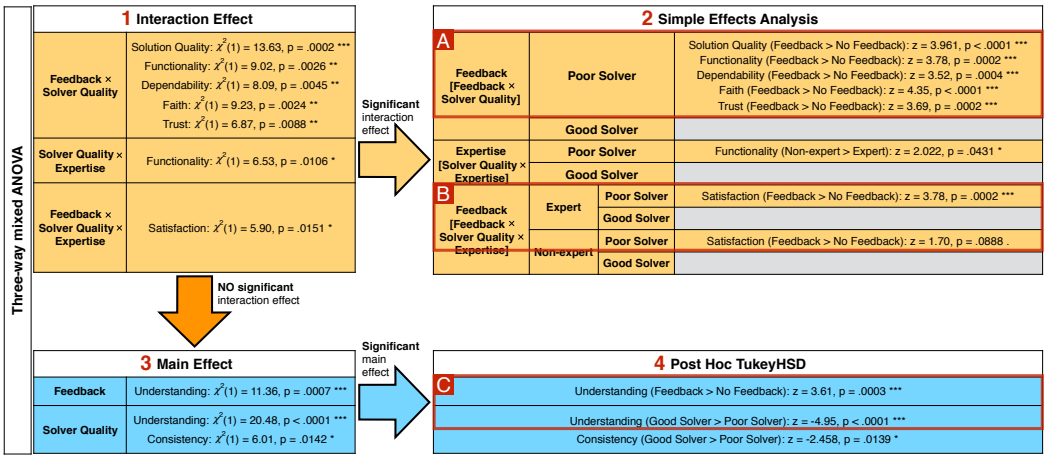

Fig. 10. Study 1 statistical analysis results. Part 1 is the interaction effect analysis. Part 2 is the simple effects analysis. Part 3 is the main effect analysis. Part 4 is the post hoc analysis. Noteworthy findings are indicated by a red rectangular outline: A, B and C. Significance codes are as follows: p ⩽ 0.001 (***); p ⩽ 0.01 (**); p ⩽ 0.05 (*); p ⩽ 0.1 (.).

The main effect of solver quality was also significant on how well participants believed they understood the solver, $\chi^2(1) = 20.48, p < .0001$ (see Fig. 10, part 3). Tukey's HSD test revealed significant differences between the good solver group and the poor solver group, $z = -4.95, p < 0.0001$ (see Fig. 10, part 4, outline C and Fig. 11). Participants believe they have a better understanding of the good solver compared with the poor solver. This was not expected. Perhaps it is because the good solver tends to produce "cleaner" solutions with fewer routes overlapping and more cluster-like routes compared with the poor solver.

None of the three possible interactions were significant on participants' understanding of the solver. The interaction of feedback and solver quality was not significant, $\chi^2(1) = 0.80, p = .3702$, the interaction of feedback and expertise was not significant, $\chi^2(1) = 0.10, p = .7529$, and the interaction of solver quality and expertise was not significant either, $\chi^2(1) = 1.14, p = .2864$.

Consistency. The main effect of solver quality was significant on solver consistency, $\chi^2(1) = 6.01$, $p = .0142$ (see Fig. 10, part 3). Tukey's HSD test revealed a significant difference between the good solver and the poor solver, $z = -2.458, p = .0139$ (see Fig. 10, part 4 and Fig. 11). This indicates that participants believed that the good solver performed more consistently than the poor solver. In reality, both the good solver and the poor solver were consistent in producing similar-quality solutions at all times. We conjecture that participants believed that the poor solver was inconsistent because they did not recognise that all of the solutions produced by the solver were bad. Thus they believed it was producing both good and bad solutions, hence inconsistent. On the other hand, they believed that the good solver generally produced good solutions, so they believed it behaved consistently.

None of the three possible interactions were significant on solver consistency. The interaction of feedback and solver quality, $\chi^2(1) = 0.03, p = .8727$, the interaction of feedback and expertise, $\chi^2(1) = 1.60, p = .2065$, the interaction of solver quality and expertise, $\chi^2(1) = 0.34, p = .5611$, were not significant.

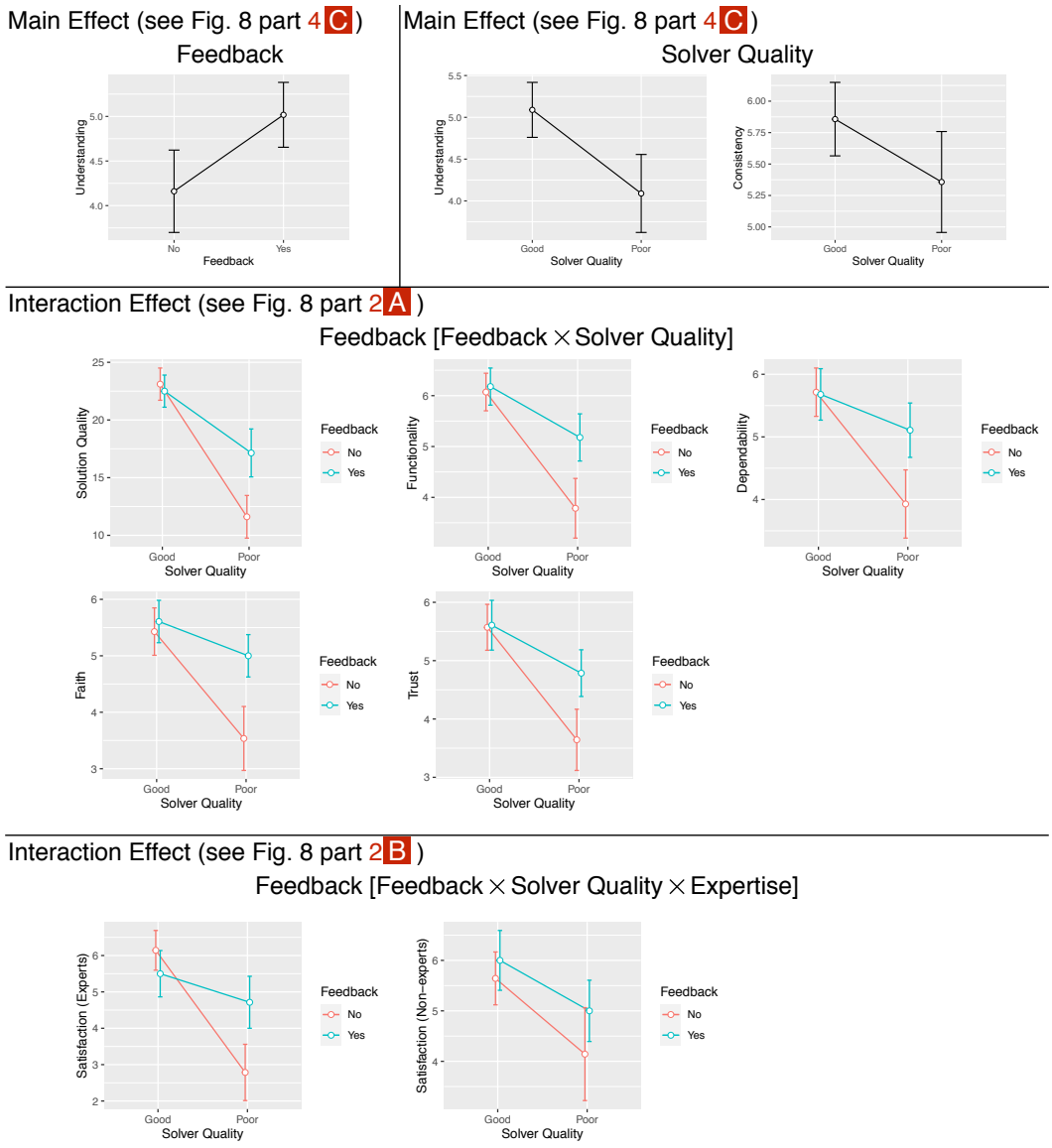

Fig. 11. Study 1 interaction graphs. [] indicates the interaction effect.

Solution quality. The interaction feedback × solver quality was significant on participants' ratings of multiple criteria (see Fig. 10, part 1; Fig. 11) including their evaluation of solution quality. Specifically, we found that the feedback × solver quality interaction was significant on participants' ratings of solution quality, $\chi^2(1) = 13.63, p < .0002$ (see Fig. 10, part 1). The main effects of both feedback, $\chi^2(1) = 4.09, p = .0432$, and solver quality, $\chi^2(1) = 64.39, p < .0001$, were significant.

Simple effects analysis with contrasts was conducted to further interpret the significant interaction between the feedback and solver quality. These contrasts compared the solution quality ratings at each level of solver quality compared to feedback at each level. The first contrast revealed

a significant difference between the feedback and no-feedback conditions for the poor solvers, $z = 3.961, p < .0001$ (see Fig. 10, part 2, outline A). The second contrast looked for differences between the feedback and no-feedback conditions for the good solvers. This contrast was not significant, $z = -0.764, p = .445$, and tells us that unlike the poor solvers, feedback does not affect people's judgement of solution qualities of the good solvers. Thus we see that providing feedback leads to an increased ranking of the solution quality for the poor solver, but that it does not significantly change the ranking of solutions produced by the good solver which is already very high (see Fig. 11).

Trust. The pattern for participant ranking of trust is very similar to that for solution quality. For participants' ratings of solvers' trust, the feedback × solver quality interaction was significant, $\chi^2(1) = 6.87, p < .0088$ (see Fig. 10, part 1). The main effects of both feedback, $\chi^2(1) = 4.93, p = .0264$, and solver quality, $\chi^2(1) = 32.45, p < .0001$, were significant.

Again we performed simple effects analysis with contrasts to interpret this significant interaction. We found a significant difference in participants' ratings of solvers' trust between the feedback and no-feedback conditions for the poor solver, $z = 3.69, p = .0002$ (see Fig. 10, part 2, outline A) but the difference of trust between feedback and no-feedback for the good solver, was not significant, $z = 0.145, p = .885$. Thus we see that providing feedback leads to an increased ranking of trust in the poor solver, but that it does not significantly change the level of trust in the good solver which is already very high (see Fig. 11).

Functionality, Dependability, Faith. Participant ranking of solver functionality, dependability and faith exhibited the same pattern as their ranking of quality of solution and trust: feedback increases their ranking for the poor solver but does not change their ranking of the good solver which is already very high (see Fig. 10, part 1; Fig. 11).

Satisfaction. We also found that feedback × solver quality interaction was significant for participants' ranking of solver satisfaction, $\chi^2(1) = 11.35, p = .0008$. However, this time we also found that the feedback × solver quality × expertise interaction was significant on participants' ratings of solver satisfaction, $\chi^2(1) = 5.90, p = .0151$ (see Fig. 10, part 1; Fig. 11). Simple effects analysis with contrasts was used to break down this interaction. Specifically, these contrasts compared participants' ratings of solver satisfaction at each level of solver quality compared to the category of expertise compared to feedback at each level (See Fig. 11). The first contrast revealed that there was a significant difference between the feedback conditions for experts of the poor solvers, $z = 3.78, p = .0002$ (see Fig. 10, part 2, outline B). This tells us that experts are more satisfied with the poor solvers when feedback is provided compared with the no-feedback poor solvers. The second contrast looked at the differences between the feedback conditions for non-experts of the poor solvers, $z = 1.70, p = .0888$ (see Fig. 10, part 2, outline B). This contrast was noticeable but not significant and tells us that non-experts are more satisfied with the poor solvers with feedback than the poor solvers without feedback. The third and forth contrasts looked at the differences between the feedback conditions for both experts, $z = -1.868, p = .0618$, and non-experts, $z = 1.282, p = .2$, of the good solvers respectively. Both contrasts were not significant. The third contrast tells us that there is a tendency for experts to be more satisfied with good solvers when feedback is provided compared with no feedback. However, there is no such tendency for non-experts. This may be because experts can make better use of the feedback about intermediate solutions to understand how the solver functions and how well the solver performs.

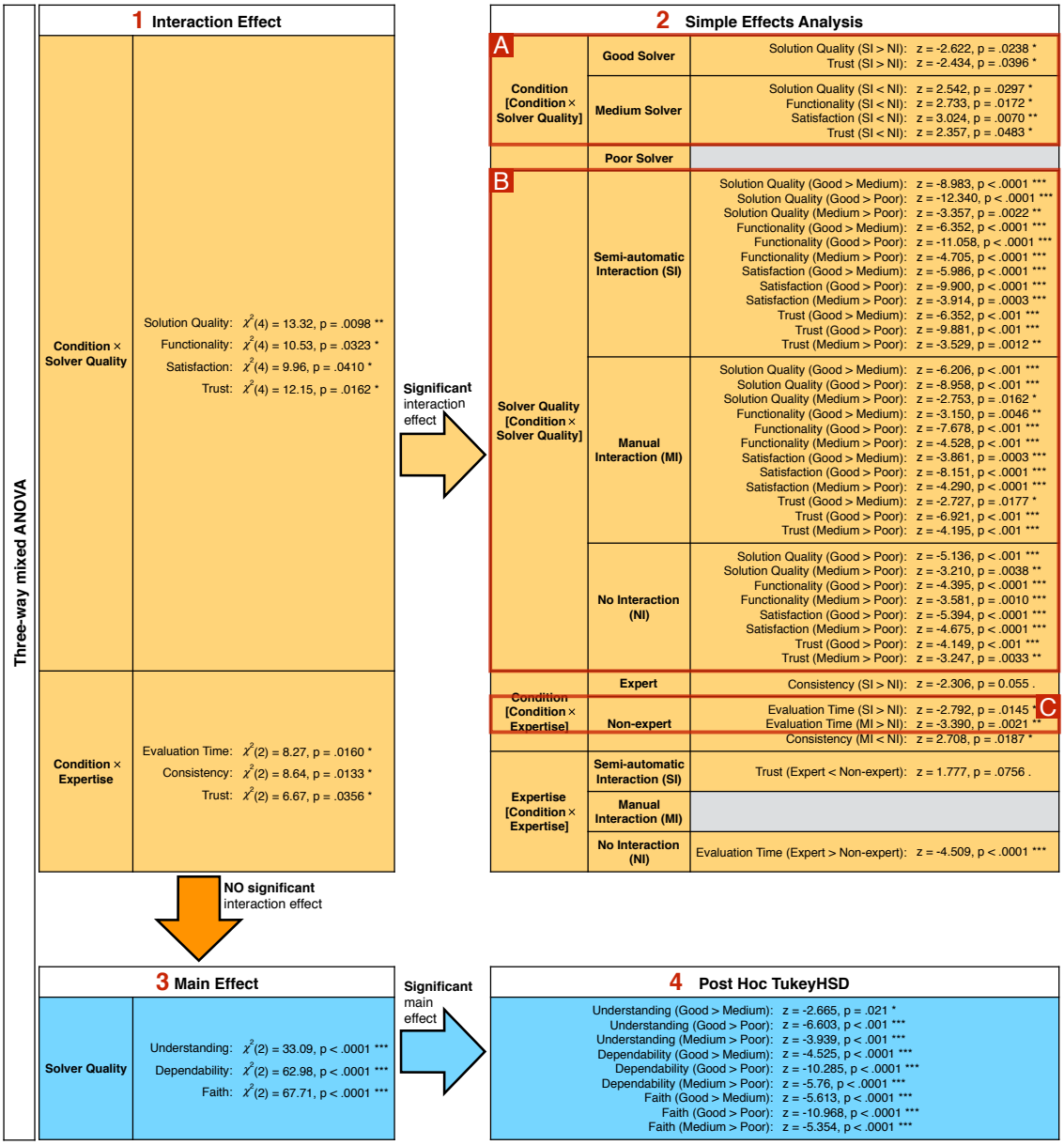

Fig. 12. Study 2 statistical analysis results. The layout is as follows: Part 1 is the interaction effect analysis; Part 2 is the simple effects analysis; Part 3 is the main effect analysis; Part 4 is the post hoc analysis. Noteworthy findings are indicated by a red rectangular outline: A, B and C. Significance codes are as follows: $p \leqslant 0.001$ (***); $p \leqslant 0.01$ (**); $p \leqslant 0.05$ (*); $p \leqslant 0.1$ (.).

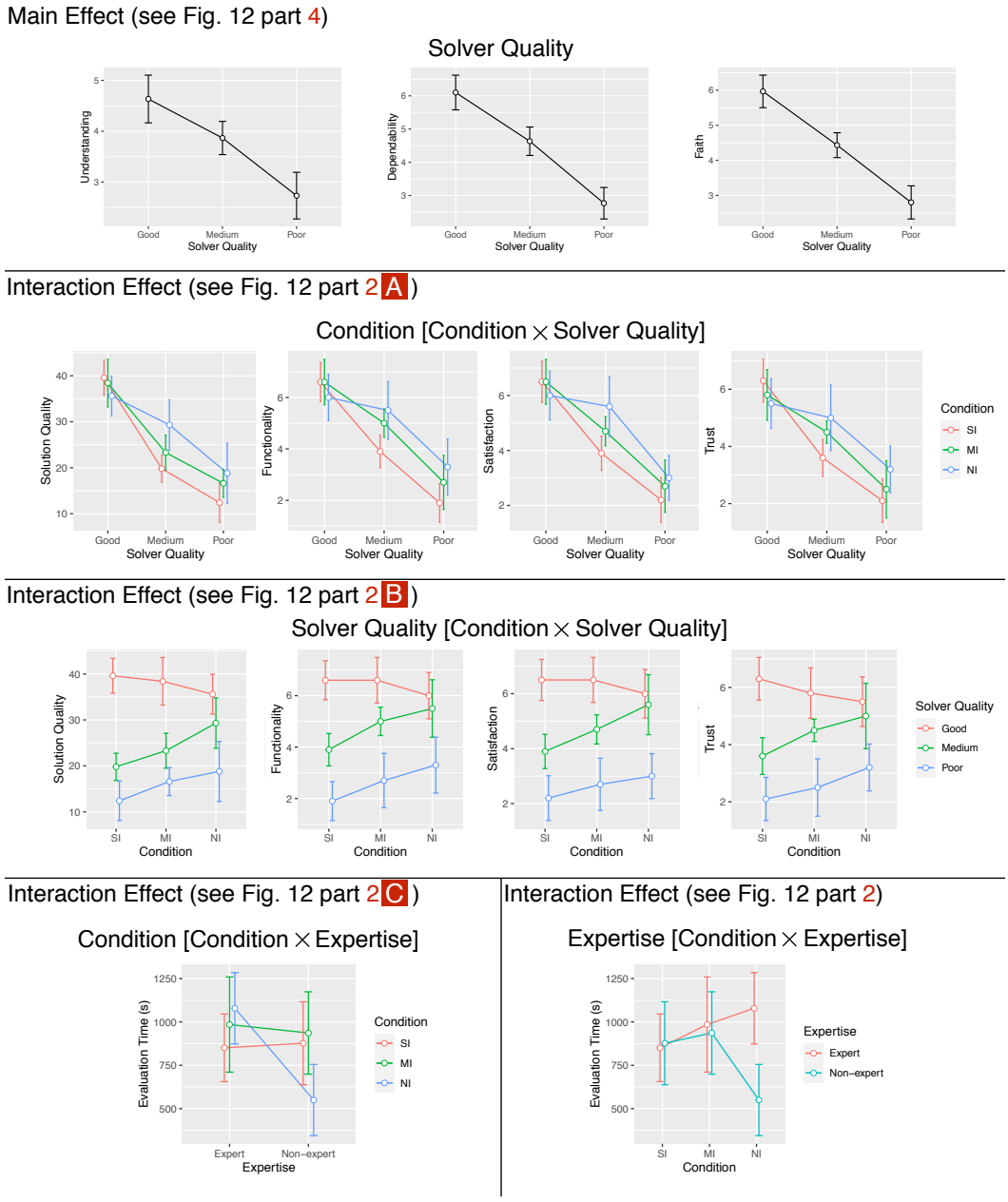

Fig. 13. Study 2 interaction graphs. [] indicates the interaction effect.

A.2 Study 2: Effect of Interaction on Turst

We report our results using the same conventions as Study 1. We interpret higher-order interactions when they are significant rather than lower-order interactions and their main effects. We only interpret main effects when they are significant but an interaction effect is not significant.

Understanding, Dependability, Faith. Participant ratings of solver understanding, dependability and faith in the solver exhibit the same general pattern. Below we analyse participant ratings of solver understanding in detail. The analysis for the two other ratings is similar.

The interaction of solver quality × experimental condition × participants' expertise was not significant, $\chi^2(4) = 8.06, p = .0893$, for ratings of solver understanding. None of the lower-order interactions were significant: $\chi^2(2) = 2.02, p = .3642$ (solver quality × expertise); $\chi^2(4) = 2.65, p = .6182$ (solver quality × condition); and $\chi^2(2) = 2.20, p = .3324$ (expertise × condition).

However, the main effect of solver quality was significant, $\chi^2(2) = 33.09, p < .0001$ (see Fig. 12, part 3). Tukey's HSD revealed that there were significant differences between solver pairs: $z = -2.665, p = .021$ (good and medium); $z = -6.603, p < .001$ (good and poor); $z = -3.939, p < .001$ (medium and poor) (see Fig. 12, part 4 and Fig. 13). Participants believed they had the best understanding of how the good solver works and a relative better understanding of the medium solver than the poor solver.

Analysis of participants' ratings of dependency and faith, reveals the same pattern (see Fig. 12, part 3 and part 4, and Fig. 13, for more details). That is, the better the solver the more highly they rank its reliability and its future performance (see Section 4.2 for detailed explanations of both criteria).

Trust, Solution quality, Functionality, Satisfaction. Participants' ratings of solver trust, solution quality, solver functionality and satisfaction with the solver exhibited the same general pattern. Again we analyse only one of these ratings in detail: overall trust in the solver.

The condition × solver quality × expertise interaction was not significant, $\chi^2(4) = 4.35, p = .3612$, for solver trust. The lower-order interaction of solver quality × expertise was not significant either, $\chi^2(2) = 5.26, p = .0719$. However, the interaction of condition × solver quality, $\chi^2(4) = 12.15$, $p = .0162$ was significant (see Fig. 12, part 1). The main effects of condition, $\chi^2(2) = 2.38, p = .3035$, and expertise, $\chi^2(1) = 3.13, p = .0768$, were not significant. But the main effect of solver quality was significant, $\chi^2(2) = 65.48, p < .0001$. We first look at the condition × solver quality interaction.

We used the same simple effects analysis with contrasts approach as in Study 1 to break down this interaction. Specifically, this group of contrasts compared participants' ratings of the overall solver trust at each level of solver quality to the interaction condition at each level with a focus on conditions. The first contrast revealed a significant difference between the semi-automatic and no interaction conditions for the good solver, $z = -2.434, p = .0396$ (see Fig. 12, part 2, outline A and Fig. 13). This tells us that the semi-automatic interaction makes people trust the good solver more compared with no interaction.

The second contrast revealed a significant difference between the semi-automatic and no interaction conditions for the medium solver, $z = 2.357, p = .0483$ (see Fig. 12, part 2, outline A; Fig. 13). Unlike the good solver, people trust the medium solver less in the semi-automatic interaction condition than the no interaction condition.

The third contrast investigated differences between the semi-automatic and no interaction conditions for the poor solver. Unlike the previous two contrasts, this contrast was not significant, $z = 1.622, p = .236$, and tells us that people do not trust the poor solver in any of the conditions.

Overall, the first two contrasts showed that semi-automatic interaction can help people better calibrate their trust in the good and medium solvers.

Other contrasts looked at differences between either the semi-automatic and manual interaction conditions, or the manual and no interaction conditions for all three solvers. None of the contrasts were significant. This suggests that manual interaction is not as good as semi-automatic interaction at allowing users to calibrate their trust due to the lack of re-optimisation ability, which plays an important role in the interactions.

The other group of contrasts investigated participants' ratings of the overall solver trust at each level of condition compared to solver quality at each level with a focus on solver qualities. The first contrast revealed that there existed a significant difference between the good solver and the medium solver in the semi-automatic interaction condition, $z = -6.352, p < .001$ (see Fig. 12, part 2, outline B; Fig. 13). This tells us that people trust the good solver more than the medium solver with the semi-automatic interaction. Similarly, the second and third contrasts looked for differences between the good solver and the poor solver, and between the medium solver and the poor solver. Both contrasts were significant, $z = -9.881, p < .001$ (good and poor solver contrast), $z = -3.529, p = .0012$ (medium and poor solver contrast) (see Fig. 12, part 2, outline B; Fig. 13). Putting everything together, these three contrasts tell us that people trust the good solver the most, then the medium solver, and trust the poor solver the least. Thus, people have the right level of trust for each solver in the semi-automatic interaction condition.

The next three contrasts investigated differences between the three solver pairs: the good and medium solver; the good and poor solver; the medium and poor solver in the manual interaction condition. All three contrasts were significant, $z = -2.727, p = .0177$ (good and medium solver), $z = -6.921, p < .001$ (good and poor solver), $z = -4.195, p < .001$ (medium and poor solver) (see Fig. 12, part 2, outline B). Again, these three contrasts tell us that people trust the good solver the most, then the medium solver, and trust the poor solver the least. Thus, people have the right level of trust for each solver in the manual interaction condition.

Using the same approach, another three contrasts looked for differences between the same three solver pairs in the no interaction condition. Two of the three contrasts were significant, $z = -4.149, p < .001$ (good and poor solver), $z = -3.247, p = .0033$ (medium and poor solver) (see Fig. 12, part 2, outline B; Fig. 13). So far the trend is the same and tells us that people trust the good solver more than the poor solver, and trust the medium solver more than the poor solver. However, the contrast between the good and medium solver was not significant, $z = -0.902, p = .6391$. The tells us that in the no-interaction condition, unlike the other two conditions, people do not significantly trust the good solver more than the medium solver.

We see an identical pattern for ratings of solution quality, solver functionality and satisfaction with the solver.

This is very interesting. Based on the above analysis and looking at Figure 13, we can see that in all three conditions the good solver and medium solver are ranked above the poor solver. For the non-interaction condition, however, participants find it difficult to distinguish between the good and medium solver. But with the semi-automatic and manual interaction conditions, they are able to determine that the good solver is better than the medium solver. This supports our hypothesis that interaction will lead to better calibrated trust.

Furthermore, we see that the difference in ranking between the three solvers is the least for the non-interaction condition and the most for the semi-automatic condition. This suggests that the semi-automatic condition allowed participants to better calibrate their trust than the manual interaction condition.

Evaluation time. Now we will briefly look at the participants' total evaluation time. The condition × solver quality × expertise interaction was not significant, $\chi^2(4) = 2.01, p = .7343$. The lower-order interactions of condition × solver quality, $\chi^2(4) = 1.48, p = .8304$, and solver quality × expertise, $\chi^2(2) = 1.00, p = .6070$, were not significant either. However, the condition × expertise interaction was significant, $\chi^2(2) = 8.27, p = .0160$ (see Fig. 12, part 1). Both the main effects of condition, $\chi^2(2) = 1.41, p = .4945$, and expertise, $\chi^2(1) = 2.94, p = .0865$, were not significant. But the main effect of solver quality was significant, $\chi^2(2) = 10.09, p = .0064$.

Again we used simple effects analysis with contrasts to break down the condition × expertise interaction. These contrasts compared participants' total evaluation time at each level of expertise to condition at each level. The first contrast revealed a significant difference between the semi-automatic and no interaction conditions for non-experts, $z = -2.792, p = .0145$ (see Fig. 12, part 2, outline C; Fig. 13). This tells us that non-experts spent more time to evaluate solutions in the semi-automatic interaction condition than the no interaction condition. The second contrast looked for differences between the manual and no interaction conditions for non-experts. This contrast was significant, $z = -3.390, p = .0021$, and tells us that non-experts took more time to evaluate solutions in the manual interaction condition than the no interaction condition. This suggests that when given the ability to interact with a solution non-experts take longer to evaluate than if they cannot interact with it.

However, we did not find significant differences between all three conditions for experts: $z = 1.006, p = .573$ (semi-automatic and no interaction); $z = 0.842, p = .677$ (manual and no interaction); $z = 0.164, p = .985$ (semi-automatic and manual interaction). This suggests that experts are more rigorous than non-experts when evaluating solutions in the no interaction condition and spend more time trying to determine if it is a good solution. This is also supported by the significant contrast between experts and non-experts in the no interaction condition, $z = -4.509, p < .0001$ (see Fig. 12, part 2; Fig. 13). We did not find any significance from other contrasts, which tells us that both experts and non-experts spent a similar amount of time evaluating solutions in both of the interaction conditions.