Effective User Interfaces for Human-in-the-loop Optimisation

by

Jie Liu
BSE (Hons)

Supervisor: Prof. Kim Marriott
Associate Supervisor: Prof. Tim Dwyer

A thesis submitted for the degree of Doctor of Philosophy (0190) at
Monash University in 2019
Caulfield School of Information Technology

To My Wonderful Parents,

Without you I will not be who I am today

To My Partner,

Who is always by my side

Contents

List of Figures

Effective User Interfaces for Human-in-the-loop Optimisation

Jie Liu, BSE (Hons)
jie.liu3@monash.edu
Monash University, 2019

Supervisor: Prof. Kim Marriott
kim.marriott@monash.edu
Associate Supervisor: Prof. Tim Dwyer
tim.dwyer@@monash.edu

Abstract

Optimisation is an important tool for decision support. It has tremendous success in helping people solve complex real-world problems in many application areas such as control systems, engineering design, data mining and even the food industry. Manual optimisation is labour-intensive and time-consuming. It becomes infeasible when the problem becomes larger and more complex. Automatic optimisation is much faster and scales much better than manual optimisation. However, unrealistic solutions are often produced by automatic optimisation. Mistrust and misunderstanding of solutions and automatic optimisation systems can also happen.

There is a growing realisation by the optimisation community of the need to directly engage the user in automatic optimisation. Such a human-in-the-loop approach for optimisation combines manual optimisation and automatic optimisation. While previous research has focussed on algorithmic aspects there has been virtually no research into HCI considerations. This thesis looks into the HCI aspect which includes appropriate visual representation and interaction techniques to support human-in-the-loop optimisation. The research draws on ideas from visual analytics which shares the same intent with human-in-the-loop optimisation to leverage the complementary strengths of humans and computers to solve difficult decision problems.

We examine the HCI aspects through two case studies. The first is treatment planning for prostate cancer treatment using focal brachytherapy; the second is vehicle routing with time windows. The main outcomes are: (1) a theoretical framework for the high-level user goals called the problem-solving loop; (2) design guidelines for the user interface; (3) improved understanding of how to engender appropriate trust in the user.

Vita

Publications arising from this thesis include:

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. In IEEE Transactions on Visualization and Computer Graphics, vol. 24, no. 1, pp. 319-329.

Permanent Address: Caulfield School of Information Technology
Monash University
Australia

This thesis was typeset with LaTeX 2ε[1] by the author.

[1] LaTeX 2ε is an extension of LaTeX. LaTeX is a collection of macros for TeX. TeX is a trademark of the American Mathematical Society. The macros used in formatting this thesis were written by Glenn Maughan and modified by Dean Thompson and David Squire of Monash University.

Effective User Interfaces for Human-in-the-loop Optimisation

Declaration

I declare that this thesis is my own work and has not been submitted in any form for another degree or diploma at any university or other institute of tertiary education. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

Jie Liu
May 8, 2019

Acknowledgments

First, I would like to express my sincere gratitude to my main supervisor, Prof. Kim Marriott, for his patience, motivation, guidance, and immeasurable support. Throughout the four years working on my PhD, I have learned tremendously from working with him. In the final months of my thesis writing, he has dedicated his time to providing insightful feedback even during Christmas time. It is very grateful to have him being such a perfect supervisor for me.

Second, I am thankful to Prof. Tim Dwyer, my associate supervisor, for being the other wonderful support for my PhD. He has always kindly shared his knowledge with me and provided valuable advice to me. Without his continuous support, this PhD work would not have been possible.

I am thankful to Annette Haworth, for her help in contacting clinical professionals and arranging appointments for my user studies. I am also thankful to Christopher Mears, for his generous help in adapting his optimisation solver to my user studies. I would also like to thank Guido Tack, for all his assistance in explaining and setting up optimisation models being used in my other user studies.

NICTA, now Data61 and Monash University have my gratitude for providing the four-year generous scholarships.

I am grateful both to my supervisors and other collaborators, from whom I can always learn something new every time we met to discuss the possibilities to improve the current human-in-the-loop interface. I am also grateful to my family and friends, who helped to make life fun and enjoyable amid all the hard work.

Jie Liu

Monash University
May 2019

Chapter 1

Introduction

After rapid industrialisation, people have been looking for faster and more powerful techniques to solve more complex real-world optimisation problems. Logisticians have spent years developing advanced algorithms to find the best routes to reduce travel distances. Project managers have spent days to come up with better schedules to meet everyone's needs and requirements to shorten the overall project time. Industrial engineers have spent countless efforts on creating the optimal trajectories for aircraft to minimise possible collisions.

Optimisation helps people to solve such complex decision problems by modelling them mathematically using *constraints* (which dictate the valid solutions) and *objective functions* (which measure the quality of the solutions) and then using powerful constrained optimisation techniques to find the solutions. The state-of-the-art algorithms can now solve a decision problem with hundreds of variables in a few minutes. As a result, optimisation is now routinely used in various application areas to solve complex real-world problems. In control systems, optimisation can be used to find the optimal trajectories for multiple aircraft to avoid collisions [12]; in engineering design, optimisation is applied to improve the material rigidity of automobile bodies [13]; and in the food industry, optimisation helps improve the quality of beer by finding out the optimal temperature for fermentation [14]. Optimisation is also used in other application areas, including arts, animation, image processing, audio processing, virtual reality, data mining, robotics, education and entertainment [15].

Approaches to solving optimisation problems can be categorised into three types, manual, human-in-the-loop and automatic. Historically many optimisation problems are solved manually. However, a purely manual approach is often very time-consuming and labour-intensive. Thus, the manual approach only works when the size of the problem is fairly small. It is infeasible when the problem becomes larger and more complex.

Scalability is one of the main reasons for the use of automatic optimisation. Automatic optimisation is much faster than manual optimisation, which allows it to be used to solve larger and more complex problems. The standard approach of the optimisation community has been to build a fully automatic system in which the solver is a "black box" and the user simply provides the problem data and then waits for the system to spit out a solution. However, for this approach to be practical, it requires the model to adequately capture the actual real-world problem and for the solver to be powerful enough to find a sufficiently good solution in a reasonable time. For many, if not most, real-world problems these are not reasonable assumptions. This means automatic optimisation may produce unrealistic solutions because of the mismatch between the optimisation model and the actual problem [1]. Furthermore, how a solution is generated and why it is optimal are not clear to the person using the system because of the "black-box" nature of automatic optimisation. Consequently, mistrust or misunderstanding of solutions and automatic

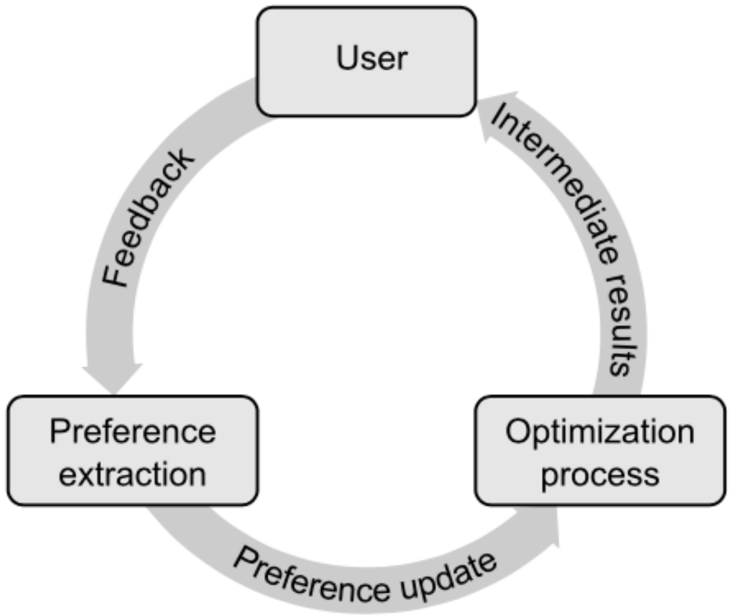

Figure 1.1: Human-in-the-loop approach for optimisation. Image from [1].

optimisation systems can occur [1]. Finally, when incomplete search methods are used, it is easy for an automatic optimisation system to become trapped in local optima [16].

As a result, there is now recognition by the optimisation community that in many applications there is a need to directly engage the user in the optimisation process (see Fig. 1.1). Such human-in-the-loop optimisation (also known as semi-automatic optimisation or interactive optimisation) has been used in practice for many decades but has only recently been recognised as a topic worthy of study in its own right.

Human-in-the-loop Optimisation.

Human-in-the-loop optimisation is the combination of manual optimisation and automatic optimisation. It is a promising approach to solve real-world complex optimisation problems without the drawbacks of purely automatic optimisation: unrealistic solutions, mistrust and misunderstanding of solutions and systems. The intent of human-in-the-loop optimisation is to combine the strengths of both humans and computers when solving optimisation problems. Human-in-the-loop optimisation has been tested on traditional optimisation problems, such as vehicle routing, bin packing and travelling salesmen [17]. It also succeeds in solving real-world problems in different application areas, such as segmenting the foreground and background of a complex image [18] and improving the classification rate of a machine-learning classifier [19].

Meignan *et al.* [1] clarifies the rationale for human-in-the-loop optimisation: "the main goal of interactive optimisation (or human-in-the-loop optimisation) is to turn efficient optimisation methods into effective decision tools." They identify the following reasons and roles for human-in-the-loop optimisation:

Role 1: Inherent limitations of mathematical models. A mathematical model is almost always a partial approximation of the real problem. The real problem may contain aspects that cannot easily be modelled mathematically [15], such as multiple conflicting criteria whose complex tradeoffs cannot be captured in an objective function; uncertainty or probabilistic constraints; or constraints that model human preferences. Allowing users to modify constraints of the problem or to adjust complex tradeoffs from conflicting objectives via interaction can minimise the limitations of mathematical models.

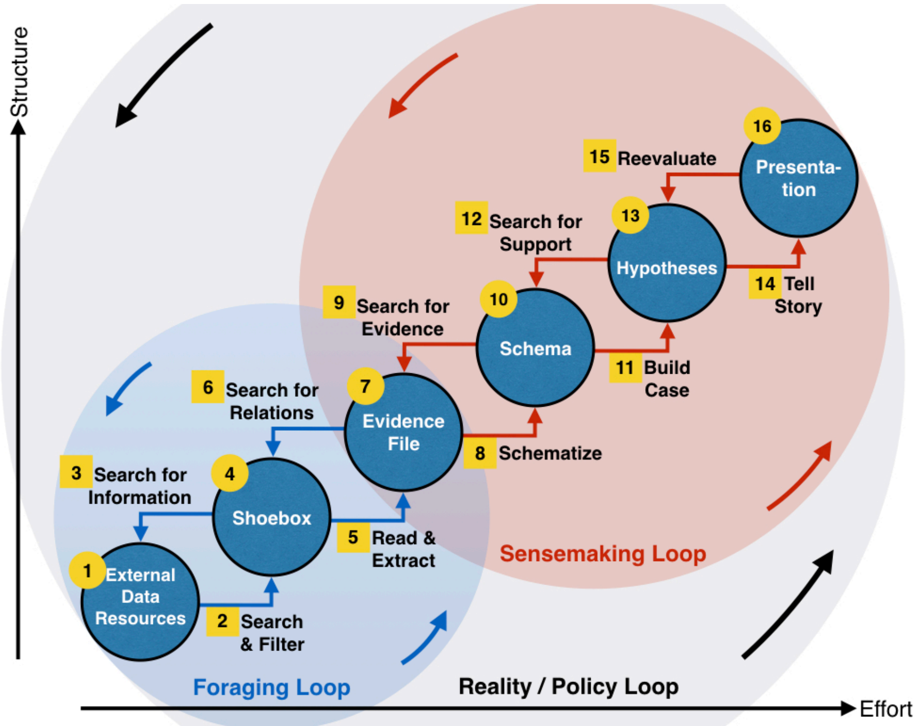

Figure 1.2: Pirolli and Card's "sense-making loop" model for the visual analytics workflow features multiple nested loops. In particular, there are two important loops: the foraging loop (blue circle) and the sensemaking loop (red circle). The foraging loop is about data collection and data processing. The sensemaking loop is about building hypotheses and presenting the results.

Role 2: Solver performance. The quality and efficiency of the solving process are crucial to the use of optimisation tools. It is difficult for the designers of the generic model of a problem to predict how efficiently the model will be solved on real-world data. With incomplete heuristic methods, such as simulated annealing, the solver may never explore that part of the search space containing the best solution. Users can assist by providing feasible starting configurations, guiding the search or tuning solver parameters for a more efficient search.

Role 3: Non-acceptance and misunderstanding of optimisation systems. The opacity of the black-box approach to optimisation may lead users to either mistrust or place too much trust in the quality of the solutions produced by the system. Allowing users to interactively explore and manually manipulate solutions being produced by optimisation systems can improve their understanding of how the systems work and how the solutions are produced. The interaction also allows what-if scenarios to be tried out.

Current research in human-in-the-loop optimisation has focused on solver speed and the quality of solutions compared with fully-automatic systems. It has not included the user experience. However, a key question of doing human-in-the-loop optimisation is the human-computer-interaction (HCI) aspect which includes appropriate visual representations and interaction techniques to support human-in-the-loop optimisation. Unfortunately, this is a subject that has received little attention from optimisation researchers. This is the subject of this thesis.

My first main research question:

RQ 1: How do we develop effective interfaces to support human-in-the-loop optimisation?

Optimisation is a widely-used tool for decision support. Another essential tool for decision support is visual analytics, which focusses on the design and evaluation of interactive visual interfaces and has many findings regarding how best to incorporate the human analyst in the data sense-making and knowledge discovery process. Both visual analytics and human-in-the-loop optimisation aim to achieve better decision making by bringing humans into the analytics-loop. With the similarity of intent between visual analytics and human-in-the-loop optimisation, we therefore believe there is considerable potential for sharing and transferring knowledge between the two research fields.

The field of visual analytics has sophisticated models of high-level goals and processes employed by human evaluators. *Sense-making* is a widely used theoretical framework for understanding visual analytics tasks. For instance, Pirolli and Cards sense-making loop (see Fig. 1.2) captures the processes employed by intelligence analysts when making sense of information. It identifies critical steps in the analytical reasoning process such as information forging to find problem-relevant information and development of insight by supporting interactive exploration of the data and presentations of the findings. However, to the best of our knowledge, there is no analogue of the sense-making loop for human-in-the-loop optimisation. Therefore building up a similar theoretical framework for understanding the high-level user goals and processes in human-in-the-loop optimisation is crucial. Thus, in order to answer the main research question **RQ 1**, we need to first consider the supplementary research question:

RQ 1-1: What are the high-level goals and processes in human-in-the-loop optimisation?

The field of visual analytics ("the science of analytical reasoning facilitated by interactive visual interfaces" [20]) combines automatic and visual analysis through interaction and visualisation techniques to gain knowledge from data [9]. In particular, the automatic analysis allows analysts to fit mathematical models built through data mining or machine learning, while visual analysis gives freedom to analysts to explore the data and model through interactive visualisation.

The combination of machine processing and human inspection in human-in-the-loop optimisation can be seen as analogous to the process in visual analytics. In human-in-the-loop optimisation, solutions are usually generated by automatic solvers. However, users can set up different parameters and choose different optimisation algorithms. Also, users can add or remove constraints from optimisation models. It is even better to support interactive visualisation for the objective spaces, which allows users to guide the solvers to explore and find better solutions.

It is critical to use visualisation and interaction techniques for the creation of an effective interface for humans to control and explore optimisation. However, the interaction and visualisation aspects are often overlooked in human-in-the-loop optimisation. Therefore, to answer the main research question **RQ 1**, we need to consider the second supplementary research question:

RQ 1-2: What are effective visualisation and interaction techniques to support human-in-the-loop optimisation?

As previously discussed in *Role 3*, optimisation solutions may not be accepted and optimisation systems may be misunderstood. Trust is the key to the problem. Trust plays a critical role in determining user reliance on automated systems [10]. As a consequence, there has been considerable research into trust and the factors affecting it [10, 11, 21]. Too much or too little trust are equally dangerous. For instance, under-trust of the ships 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 [10]. The Goldilocks trust dilemma is, therefore, how do we design computer applications that engender exactly the right amount of trust: not too much; not too little.

This leads us to my second main research question:

RQ 2: How to engender the right amount of trust in optimisation systems?

One important factor affecting trust is whether the user understands the algorithm being used by the system and believes it capable of achieving their goals [11]. 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, on the one hand, it is challenging to explain the progress towards the optimum solution in a way that could be understood by general users to gain trust. On the other hand, users can easily lose trust in optimisation systems when they lack an understanding of the underlying optimisation algorithms.

Users may still lose trust when optimisation algorithms are not so complex. A standard local search algorithm iteratively improves a candidate solution. It is intuitive to understand by the general public without a computer science background. However, users lose trust if the presentation of solutions makes no sense or even causes confusion.

Another factor affecting trust is the users evaluation of the system performance. Often, it is very difficult for the typical user to evaluate the quality of a solution produced by an optimisation system on a real-world problem unless it has some obvious faults. Also, it is very hard to know if there is another solution that improves the objective. This may lead to a situation were users place little trust in the system. A possible approach to gain users' trust is using a visual-analytics-based method to allow them to interactively explore and visualise the solution space to develop an understanding of the solution returned by the system. Indeed some optimisation researchers have previously suggested that interaction leads to increased trust [1]. However, this lacks support from experimental evidence.

1.1 Research Methodology

We investigated the HCI aspects of human-in-the-loop optimisation through two case studies. We followed the exploratory case studies methodology in the first case study to understand the clinical treatment planning procedure for prostate cancer. The case studies provided insights into the identification of user goals and processes in the context of human-in-the-loop optimisation. Such identifications were valuable for the development of an effective human-in-the-loop optimisation interface. For the second case study, we used the explanatory case studies methodology to seek approaches to engender the right amount of trust. In particular, we investigated how trust is affected by feedback and interaction.

Design science focusses on creating and evaluating IT artefacts to solve real problems [22]. Similarly, my research also tries to solve the problem of building an effective interface to support human-in-the-loop optimisation using two case studies. In order to solve it, we developed an interactive optimisation tool for each case study. The aim of the first tool was to support interactive treatment planning for prostate cancer using a new focal therapy approach. The second tool was intended to be used as a testbed to identify the important factors contributing to the development of an effective human-in-the-loop optimisation interface.

Design science research also expects researchers to evaluate the solutions based on appropriate criteria [22]. Our first tool was evaluated by seven radiation oncology professionals in the form of a two-stage semi-structured interview. Professionals were asked to compare, evaluate and improve different treatment plans using the tool. We used the methodology of controlled experiments for the evaluation of the second tool with two experiments. The first controlled experiment is a within-participants experiment, in which we controlled the option of providing optimisation solving feedback to different solvers and asked participants to evaluate solutions under different conditions. In the second experiment, we controlled the experimental conditions with the setup of three different interaction levels. It was a between-participants experiment with participants equally assigned to each experimental condition to evaluate solutions produced by different optimisation solvers.

1.2 Contributions

The contributions of this research fall into five main categories:

- Prostate brachytherapy case study.
  We conducted a case study investigating human-in-the-loop optimisation for prostate brachytherapy planning with treatment planning problems. This is one of the few user studies to investigate human-in-the-loop optimisation with real clinical professionals. It clarifies the reasons of why clinical professionals do not use fully-automatic optimisation systems. It allows us to explore the use of an interactive optimisation tool.

- Vehicle routing problem with time windows case study.
  In the second case study, we developed an interactive optimisation tool for vehicle routing. A small case study reveals the effectiveness of the design and reflections of the design guidelines.

- The problem-solving loop.
  Based on our first case study, we have suggested a novel theoretical framework, the problem-solving loop. It identifies the high-level user goals and tasks in human-in-the-loop optimisation.

- Nine design guidelines.
  We have extracted and generalised nine design guidelines based on the problem-solving loop. These design guidelines include the core visualisation components and the fundamental interactions for the development of an effective human-in-the-loop optimisation interface.

- Engenderment of an appropriate level of trust.
  Through two controlled experiments investigating user's trust, we find empirical evidence on how providing feedback about intermediate solutions and the objective function can lead to increased trust though it may be unwarranted. We also find evidence that allowing users to semi-automatically manipulate solutions returned by optimisation systems can lead to a better calibrated trust. [1]

1.3 Outline

Chapters 1, 2 and 7 provide introduction, background and conclusions respectively.

Chapter 3 starts with the detailed description of the problem context of the first case study: prostate brachytherapy. It explains different treatment methods for prostate cancer, including the standard method called whole gland therapy and a new focal therapy method. This chapter also includes the development of our prototype interactive optimisation tool for focal therapy, which explains how the tool evolved and the design decision made during the development period.

Chapter 4 presents the evaluation of the prototype tool discussed in Chapter 3 using two studies with radiation oncology professionals. The first study aims to understand why

[1] A calibrated trust means good solutions are not under trusted and poor solutions are not over trusted.

manual planning is preferred by professionals in traditional whole gland therapy. Then we propose our theoretical framework called the *problem-solving loop*, which identifies high-level user goals and tasks in human-in-the-loop optimisation, as a result after the interview. In the second study, we explore the professionals' preferred method for creating treatment plans for focal therapy.

In Chapter 5 we present nine design guidelines extracted and generalised based on the *problem-solving loop* proposed in Chapter 4. Then we re-evaluate the prototype tool for prostate brachytherapy discussed in Chapter 3 using the nine guidelines. Afterwards, we describe our second case study: vehicle routing problems with time windows. Before we jump into the experimental results in Chapter 6, it is worthwhile to justify whether vehicle routing interface development benefits from the guidelines. At the end of this chapter, we include a heuristic evaluation of the vehicle routing interface.

Chapter 6 presents the results of two controlled experiments, investigating how users' trust is affected in the context of human-in-the-loop optimisation based on the use case described in Chapter 5. The first experiment investigates how providing feedback about interim solutions and the objective function affects users' trust. In the second experiment, we investigate how interactions can affect trust by allowing users to make changes to solutions returned by optimisation solvers.

All figures in the thesis are properly attributed. The ones without attribution were created by the author.

Chapter 2

Background

In this chapter, we begin with a brief look at the field of optimisation in general. After this, we present algorithms and characteristics of optimisation problems, followed by a discussion of the three issues with the current fully-automatic approach to optimisation briefly mentioned in Chapter 1. The first issue is the mismatch between the optimisation model and the real-world problem; the second is the solver performance; while the third is non-acceptance and misunderstanding of optimisation solutions and systems.

Then we briefly introduce human-in-the-loop optimisation, which can address the limitations of fully-automatic optimisation. In particular, we identify a lack of research into the HCI aspects of human-in-the-loop optimisation. For this reason we consider the research field of visual analytics. Human-in-the-loop optimisation and visual analytics share many similarities, but visual analytics research has intensively studied the design of effective interactive interfaces for analytics. So we believe it can bring insight into the design of interfaces for human-in-the-loop optimisation systems.

2.1 Optimisation

Logisticians optimise. They aim to find the best routes to reduce travel distances. Engineers optimise. They seek to create the optimal trajectories for multiple aircraft to minimise the possibility of collisions. Hospitals optimise. They schedule doctors and equipment for maximum efficiency.

Optimisation is an important tool for decision support. It aims to find the best solution to a decision problem. The first step of using optimisation is to identify the *objective*, which represents the goal in a quantitative way [23]. In a vehicle routing problem, a typical objective might be minimum travel distance or minimum fuel consumption.

The objective depends on a number of *variables*, which capture different aspects or characteristics of an optimisation problem [23]. Variables are also called the *unknowns* of the problem [23]. Taking the same vehicle routing problem, the variables involved can include choice of vehicle, the goods transported to a particular customer and the time of delivery. Variables can be *constrained* or *unconstrained*. If a variable is constrained, it means there are restrictions applied to the variable [23] so that the available values of this variable are limited in some sense. For example, the load of a vehicle must not exceed its maximum loading capacity.

The process of identifying objective, variables and constraints associated with variables for an optimisation problem is called *modelling* [23]. The model is specified in terms of *parameters* which are fixed for a particular problem instance. These might be the number of vehicles, number of customers and their location as well as the number of items to be delivered to each customer and their weight. With such a mathematical model constructed, producing a solution to the optimisation problem means finding values of the variables such that the objective is optimised without breaking any of the constraints on the variables. Considering the vehicle routing problem example, if the objective is to minimise travel distance, a solution is to find routes from customers to customers that require the least cumulative total distance travelled by all vehicles without exceeding the loading capacities of each vehicle.

We give another example to explain modelling. In portfolio optimisation, we seek the best way to distribute assets. A variable can be the investment in a particular asset. Possible constraints are the budget limitation and the minimum expected portfolio return. The objective might be a measure of the overall portfolio return. The portfolio optimisation problem is to find a portfolio allocation which maximises the return.

In general, when we mathematically formulate an optimisation problem, it can be represented as follows:

$$\begin{aligned} min \quad & f(x) \\ subject\ to \quad & g_i(x) \leqslant b_i, \quad i \in E. \end{aligned}$$

$f(x)$ is the objective function. x is a vector of variables. $g(x)$ is a vector of constraints, in which b is a vector of known parameters of the problem and E represents the indices. For simplicity and without loss of generality we can assume that the objective function is to be minimised.

2.1.1 Algorithms and Characteristics of Optimisation Problems

The choice of optimisation algorithm used to solve an optimisation problem is heavily dependent on the the characteristics of the problem. Therefore, we discuss optimisation algorithms and characteristics of optimisation problems together, with an emphasis on the characteristics.

Unconstrained and Constrained Optimisation.
An important distinction between optimisation problems is whether the problem has constraints on the variables.

Sometimes the constraints can be safely disregarded if they do not affect solutions [23]. This is an *unconstrained optimisation* problem. It is possible to translate constrained optimisation problems into unconstrained optimisation problems by introducing constraints as penalty terms in the objective function. By minimising a penalty term in the objective a constraint will tend to be satisfied. This is not guaranteed however. Solutions may violate some of the constraints, which is why such constraints are called *soft constraints*. When constraints are explicitly defined, they are *hard constraints*. Problems with hard constraints are *constrained optimisation* problems. Violations of hard constraints are not acceptable and result in an invalid solution to the problem.

The characteristics of the objective function and the constraints is another important criterion used to classify optimisation problems.

Linear and Nonlinear Programming.
In *linear programming* (LP, also called linear optimisation) problems, both the objective function and all constraint functions are linear. It is easy to solve a linear programming problem with hundreds or thousands of variables. Dantzig's simplex method is probably one of the most commonly used algorithms to solve such problems. It solves the problem by finding the optimal vertex of the polytope, the feasible region containing all possible values of variables when searching along its edges to minimise or maximise the objective function.

When either the objective function or some of the constraints are not linear, the problem is a *nonlinear programming* (NLP) problem. In practice, many problems in the fields of engineering and economics are nonlinear optimisation problems, such as the portfolio optimisation problem given above.

Depending on the characteristics of the objective function and constraints, there are various kinds of nonlinear programming problems. For instance, when the objective function is quadratic and all constraints are linear, the problem is called a *quadratic programming* problem [23]. Interior-point methods can be used to solve the problem. Unlike the simplex method discussed above, the interior-point method traverses the interior of the feasible region to find the best solution. There exist many other algorithms to solve quadratic programming problems such as the active set method and the gradient projection method.

A important difference between linear and nonlinear programming is the difficulty of finding optimal solutions. This leads to the introduction of global and local optimisation.

Local and Global Optimisation.
In local optimisation, algorithms seek a *locally optimal* solution, which means that the objective function is minimised among all nearby feasible points. However, there is no guarantee that the objective function is lower than all other feasible points. A solution for which this is true is said to be a *global optimum* [24].

A large amount of research in nonlinear programming has focused on local optimisation methods [25]. Hill climbing is one of the many local optimisation methods to solve nonlinear optimisation problems. It is an iterative method, which begins from a random solution and then tries to find a better solution by applying changes to the current solution. When a better solution is found, the algorithm applies changes to the better solution to find even better ones. The process terminates when no further improvements can be made to the current best solution. The resulting solution is locally optimal but may not be globally optimal. Many other optimisation methods, such as ant colony optimisation (ACO) or genetic algorithms (GA), are only guaranteed to find a local optimum. GAs are methods based on natural selection which drives biological evolution.

A big advantage of local optimisation is speed. It can find a relatively good solution in a reasonable amount of time. Therefore, local optimisation methods can be used to solve large-scale problems with more than a hundred thousand variables, where there is no need to find the very best solution.

One of the disadvantages of local optimisation is randomness. The methods require a starting point to begin the search of all neighbourhood points around it. Often, the starting point is randomly determined. The randomness can critically affect the quality of the local solution. Every time the methods use a random starting point, it is possible to produce drastically different local solutions. Therefore, fine-tuning the algorithm parameters becomes necessary and it can be tricky to do so. However, randomness is not always an issue. It can improve the diversity of the solutions in some algorithms.

Another disadvantage of local optimisation is incompleteness. As previously mentioned, the methods work by searching the neighbourhood of a candidate solution in order to discover a better one. However, if no better solutions can be found from the neighbourhood of the current candidate, the methods get stuck in a local optimum, the current best solution found. The methods are incomplete in the sense that not all feasible points are visited and checked. Thus, whether or not the current local optimum is also a global optimum cannot be proven.

In global optimisation, the aim is to find a global solution. Unlike a local solution, a global solution guarantees that the value of the objective function is the lowest considering all possible feasible points.

The biggest disadvantage of global optimisation is that it does not scale very well. In the worst case scenario, the complexity of a global optimisation method can grow exponentially with the size of the problem [25]. Even a small problem with tens of variables can take hours or even days to solve [25]. However, global optimisation has been used in the field of engineering design when the time to compute a solution is not critical and the value to find a global solution is highly significant [25]. One example is the worst-case analysis of a high-consequence system, which requires a critically high level of safety.

As mentioned previously, there is a difference between linear and nonlinear programming in terms of solution optimality. Clearly any global optimum is also a locally optimum but not vice versa. However in the case of linear programming, any local optimum is also a global optimum. More generally, local optima are also global optimums in convex programming, a generalisation of linear programming when both the objective function and constraints are convex. However, this is usually not the case for nonlinear programming and so in practice "good" locally optimal solutions are sought [23].

Single- and Multi-objective Optimisation.
When the problem has more than a single objective, it becomes harder to find a good solution. If there is only one objective function to be minimised or maximised, it is a *single-objective* problem. Traditional optimisation problems such as the travelling salesman problem (TSP) or the knapsack problem have a single objective function. However, many real-world complex optimisation problems, such as vehicle routing or portfolio optimisation often consider more than one objective to be optimised simultaneously. Such problems are called *multi-objective* optimisation problems.

In many cases, the objectives conflict with each other. A solution is *Pareto optimal* if none of the objective functions can be improved without worsening some of the other objective functions. A single best solution that simultaneously optimises all objectives usually does not exist. However, a number (possibly infinite) of equally-good Pareto optimal solutions can be discovered [26,27] when subjective preferences are not considered. The set of all Pareto optimal solutions is known as the Pareto front or Pareto set [28] (see Fig. 2.1). Multi-objective optimisation, compared to single-objective optimisation, has a new element: to select a solution from the Pareto front [27].

There exist many approaches to solve multi-objective optimisation problems. One of the most commonly used approach is the aggregating function approach, in which all objectives are aggregated into a single objective function with a weight parameter assigned to each objective. The problem is solved by finding the optimal value of the aggregated single objective function. However, how determining how much weight to assign to each objective can be tricky and subjective.

Continuous and Discrete Optimisation.
Another important distinction in optimisation methods involves the kinds of values the variables can take. In *continuous optimisation* problems, all variables are required to be continuous variables, which means the values of the variables are selected from an infinite set of real numbers [23]. However, it does not make sense to use a continuous variable when measuring the number of vehicles in a vehicle routing problem for instance. The variable can only take a non-negative integer value rather than a real number in this case. More broadly, in *discrete optimisation* problems, variables are restricted to take only discrete values. If there is a constraint requiring the values of variables to be integers, the problem is known as an integer programming problem [23]. In general, discrete optimisation problems are harder to solve than local optimisation problems. In practice, good locally optimal solutions are sought rather than true globally optimal solutions.

Deterministic and Stochastic Optimisation.
In deterministic optimisation problems, the optimisation model is fully specified [23]. This means the value of the objective function remains the same if the same variable inputs are given. However, sometimes the optimisation model depends on unknown parameters, which might be predicted or estimated rather than precisely specified. This type

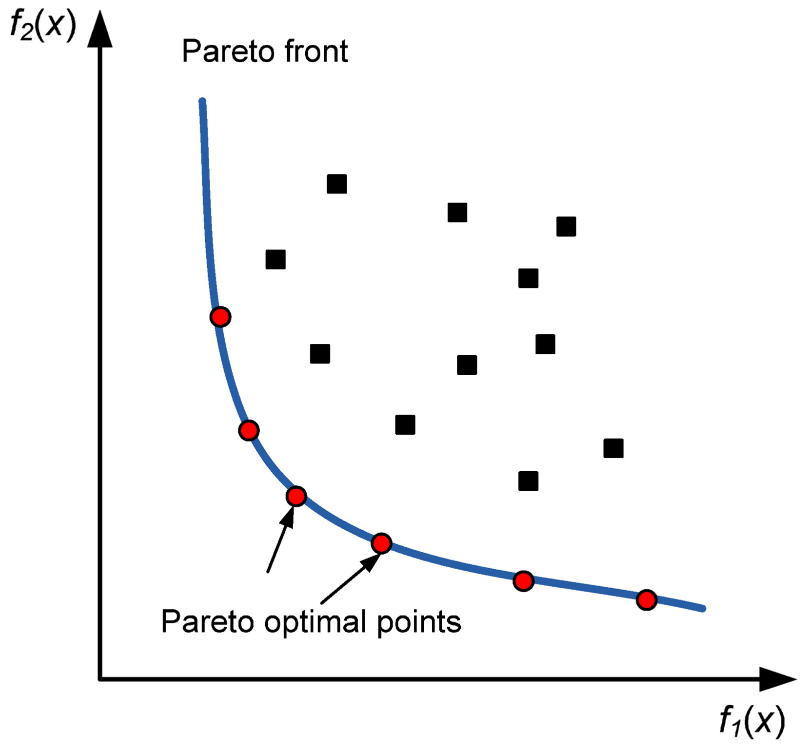

Figure 2.1: A demonstration of the Pareto front for an optimisation problem with two objectives: $f_1(x)$ and $f_2(x)$. Image from [2].

of problem is called a *stochastic optimisation* problem. Unlike deterministic optimisation problems, there is no guarantee to obtain the exact same objective each time in stochastic optimisation because of the uncertainty and possible randomness in variables or even objective functions.

2.1.2 Current Issues in Optimisation

Most algorithms for solving optimisation problems are black-boxes, and the typical solving processes are automated. The consequences of the black box nature lead to the limited use of fully-automatic optimisation.

As previously mentioned in Chapter 1, the first limitation of fully-automatic optimisation is due to the mathematical model. A mathematical model is an abstraction of a real-world problem, which means that not every single aspect of the problem can usually be modelled. The mathematical model only partially represents the problem [23, 29]. This creates a gap between the model and the problem. There are two main possible reasons for the existence of this gap.

First, it is not always possible to clearly define the objectives and constraints. Sometimes they are not quantifiable [30]. This means they cannot be precisely defined and mathematically modelled, such as risk management and trust measurement. It is possible that unquantifiable objectives and constraints may be approximated. However, the reliability and accuracy of such approximations need to be carefully tested and verified.

Second, the optimisation model is "incomplete". Limited resources such as limited time and budget can contribute to the incompleteness of the model [1]. This can also happen when optimisation experts have a limited understanding of the problem context or the problem is too complex to have a complete specification [1]. However, sharing knowledge among experts may improve the completeness of the optimisation model.

A second limitation of fully-automatic optimisation is solver performance. The quality and efficiency of the solving process are two crucial considerations in the use of optimisation tools. As discussed in Section 2.1.1, the underlying optimisation algorithms can quickly find locally optimal solutions but might not be able to find (or at least prove) a globally optimal solution in a reasonable time frame. Often, the performance of the solver depends upon parameters or the choice of solution the solver starts with. This means that the quality of solution or time of search to find a global solution can vary drastically from problem instance to instance.

The third limitation in optimisation is non-acceptance and misunderstanding of optimisation solutions and systems. As discussed in Chapter 1, building appropriate trust is the key to overcome this limitation. Inappropriate trust in optimisation solutions can be either lack-of-trust and overtrust.

David and Kottemann [31] investigated the optimisation system performance under different experimental conditions with two experiments. In the first experiment, the user perceived that the what-if analysis was more effective than unaided decision making even when there was no significant difference in performance. The "illusion of control" boosts users' trust in solutions. The results of the second experiment showed more evidence to support the above finding. Users still thought what-if analysis was superior to decision rules even though decision rules could significantly improve systems' performance if used, whereas what-if analysis did not. This leads to overtrust.

Both lack-of-trust and overtrust are equally dangerous. We include a more detailed discussion on this trust topic in Chapter 6.

2.2 Human-in-the-loop Optimisation

As a result of these three limitations there is a growing realisation by the optimisation community, driven by application requirements, of the importance of giving users greater control over the optimisation process. The human-in-the-loop approach to optimisation differs from the traditional fully-automatic optimisation by providing interactive interfaces for humans to control and explore optimisation systems and solutions. However, the idea of involving users in the optimisation solving process is not new.

In the early 1970s, Wallenius [32] investigated the performance of the human-in-the-loop approach for multi-objective optimisation problems and reported it was not ideal to cooperate human and machine together to solve this kind of optimisation problems. However, one year later, Wallenius [33] used interactions to elicit users' preferences on multiple objectives by answering yes and no questions to express such preferences. This time the human-in-the loop approach was superior. It also showed that users' acceptance of solutions improved.

A few years later, in 1985, the human-in-the-loop optimisation approach was tested on other kinds of optimisation problems. Glover *et al.* [34] developed a human-in-the-loop decision support system for architectural design and produced a high-quality solution in a much shorter time compared with traditional design methods. One year later, Jones and Maxwell [35] implemented a human-in-the-loop factory scheduling system to enhance the scheduling process. In the same year, Fisher [36] presented a survey about the successful use of human-in-the-loop optimisation in various application areas including vehicle routing, job scheduling and planning.

In the last 20 years, human-in-the-loop optimisation has been widely used in many different application areas to solve complex real-world problems. For instance, clinicians have used human-in-the-loop optimisation to produce better radioactive treatment plans for different cancers [37–39]. On the one hand, delivering a high level radiation dose to the cancer tissue is expected in order to maximise tumour control. On the other hand, it

Figure 2.2: A human-in-the-loop optimisation system supporting the HuGSS framework to solve the capacitated-vehicle-routing-with-time-windows (CVRTW) problem. Image from [3].

is very important to lower the radiation dose to surrounding healthy tissues. Clinicians need to be involved in the optimisation process in order to balance the treatment quality and treatment risk. Being able to interactively search for treatment planning proposals based on preferences and visualise different plans allows clinicians to find high-quality treatment plans in a shorter time compared with the traditional trial-and-error planning approach [37, 39] or other clinically approved planning programs [38].

Human-in-the-loop optimisation has also been used in interface design. Bailly *et al.* [40] conducted a user study to compare the performance of manual and human-in-the-loop optimisation in menu design and found that using human-in-the-loop optimisation can reduce the editing effort to achieve equally good menu designs.

Another example application area is vehicle routing. Anderson *et al.* [3] presented a human-in-the-loop optimisation system based on the human-guided simple search (HuGSS) framework to solve a particular kind of vehicle routing problem (see Fig. 2.2). The system allowed users to manually modify the current solution by assigning a customer to a route. It also allowed users to guide the optimisation search process with the use of priority settings for each customer. Specifically, the underlying algorithm only moved customers with high priorities to routes without low-priority customers. Moreover, users had other controls of the optimisation search process including determining how deep the search was and when to stop the search.

Including users in the optimisation process to explore and analyse optimisation makes it possible for human-in-the-loop optimisation to overcome problems from fully-automatic optimisation, such as the model-problem gap and solver performance limitations. We explain this together with the interaction approaches and visualisation focusses in human-in-the-loop optimisation in the following sections.

2.2.1 Interaction Approaches in Human-in-the-loop Optimisation

The success of human-in-the-loop optimisation not only relies on the performance of the underlying optimisation algorithms but also crucially depends on providing effective interaction techniques. Here we give a brief summary of interaction approaches in human-in-the-loop optimisation with a focus on the utility and purpose of the interactions. We also explain how human-in-the-loop optimisation can overcome some of the limitations from fully-automatic optimisation by the use of these interaction approaches.

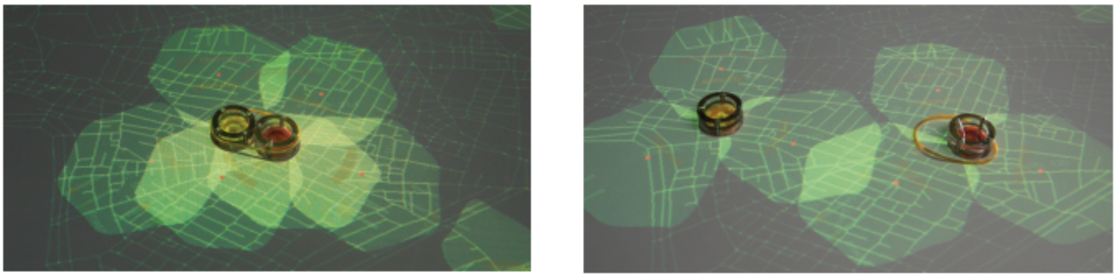

Figure 2.3: Constraints are added to the optimisation model using physical objects (pucks in this case). The model is used to solve a cellphone tower coverage problem. The image on the left explains the situation where two cellphone towers are constrained to stay together using a rubber band. The image on the right shows what happens after removing the constraint by releasing the rubber band. The underlying software guides both cellphone towers to move away from each other to gain better coverage. Image from [4].

Refine Optimisation Models.
The purpose of refining an optimisation model is to address the problem of conflicting models and incomplete models. Often, multiple objectives are conflicting (recall from Section 2.1.1). Interaction allow users to guide optimisation systems to balance the trade-offs between conflicting objectives using user preference feedback.

In terms of incomplete models, users are provided with interactions to add missing constraints or objectives, or to refine the existing ones to enrich the underlying model and make it more realistic. When constraints or objectives are not quantifiable, users can use interactions to rank solutions. We will discuss this specific situation later.

A number of human-in-the-loop optimisation systems support interactions to refine optimisation models [3,4,40–46]. In particular, a network layout editing tool "Dunnart" [41] allowed users to directly add constraints to a node in a network diagram with a constraint menu containing three types of constraints: alignment, separation and containment. Patten and Ishii [4] presented a human-computer interface called physical intervention in computational optimisation (PICO) to allow users to add constraints using physical interactions on a tabletop interaction surface (see Fig. 2.3). It was a very innovative and intuitive way to apply constraints to a problem. However, the sophisticated requirement for both software and hardware made this kind of interaction less applicable to other problems.

Control Optimisation Search Process.
Another kind of interactive approach is to support users' control of the optimisation search processes. This is useful to improve the solver performance because users are given control of when to stop the algorithm and in which areas to search for solutions.

Having direct control of the search process is a straightforward way to guide the optimisation algorithms. However, it requires users to have a certain level of expertise in the field of optimisation. In a network layout tool "HapStar" [47] for gene sequences, users can stop the algorithms and perform manual changes to the current layout such as repositioning overlapped long gene branches, after which users can continue the algorithms to further layout gene sequences. Moreover, users can set up the parameters of the underlying layout algorithm to control how the layout looks like. Ugur and Aydin [5] presented a simulation tool to solve the travelling salesman problem (TSP) using the ant colony optimisation (ACO) algorithms, a particular kind of nature-inspired population-based optimisation algorithm. See Figure 2.4. Users were given great control to setup

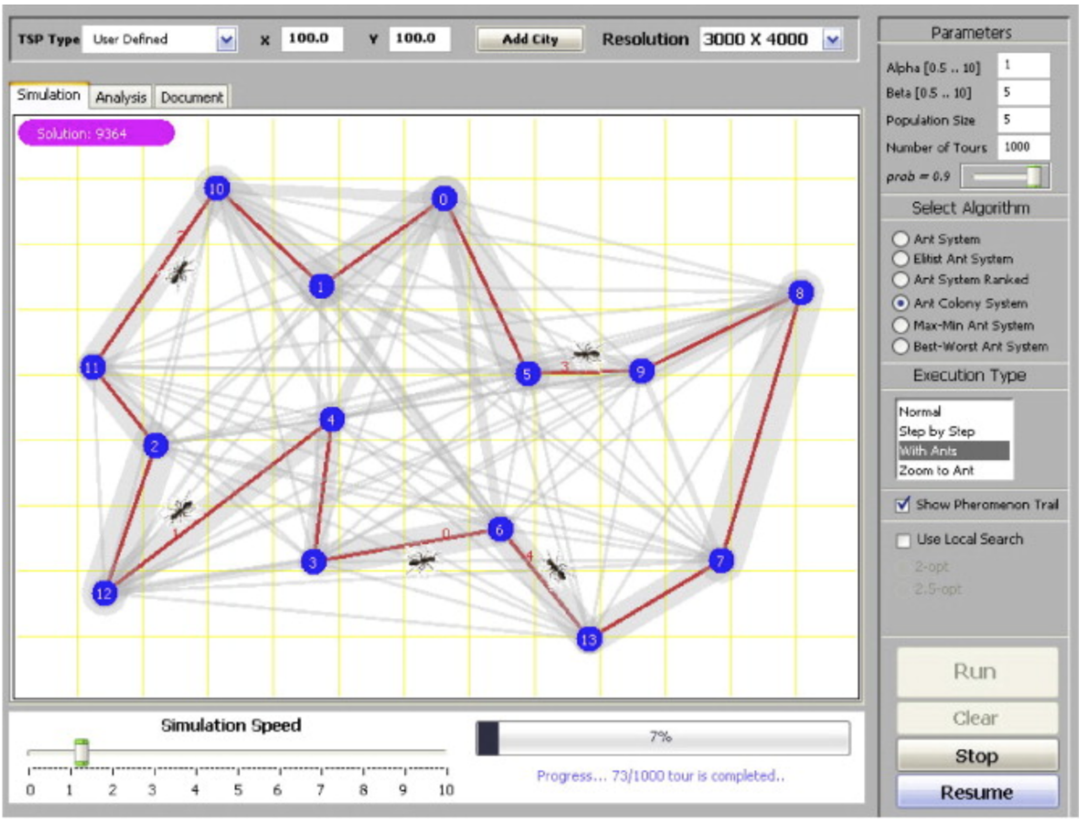

Figure 2.4: The simulation and analysis tool for solving TSP using ACO algorithms. Image from [5].

algorithm parameters, choose which algorithm to run and specify how to run it. As discussed in Section 2.2, the human-in-the-loop optimisation system supporting the HuGSS framework also allowed users to control and guide the optimisation search process. Often, after user guidance, solution re-optimisation is performed.

Rank Solutions.

As mentioned previously, ranking solutions is a useful interaction approach especially when constraints or objectives cannot be precisely defined. Typically it takes a number of iterations to produce a satisfying solution. During each iteration, optimisation algorithms automatically adjust the values of variables to produce a new generation of solutions based on users' solution rankings from the previous iteration. In such a way, the unquantifiable constraints or objectives can be captured in the final satisfying solution. The above process of ranking solutions belongs to the class of optimisation methods called interactive evolutionary computation (IEC), a family of optimisation algorithms inspired by biological evolution in which the evaluation of solutions is provided by humans [15].

Such interactive evolutionary computation approaches have been intensively used to solve various design problems. Cho [48] used IEC to solve a fashion design problem with the objective of producing cool-looking clothing designs. Different parts of clothes were encoded with two categories: colour and design style. During each iteration, a number of solutions were presented using 3D models. Users were asked to evaluate the solutions and to provide ratings. The solutions ranked the highest were kept as the basis for new solutions for the next iteration until cool-looking clothes were produced by the system. Brochu *et al.* [6] used a similar different approach to design satisfying animations. However their system did not use the IEC approach to generate new solutions. Instead, the system used machine learning to automatically capture users' animation preferences after

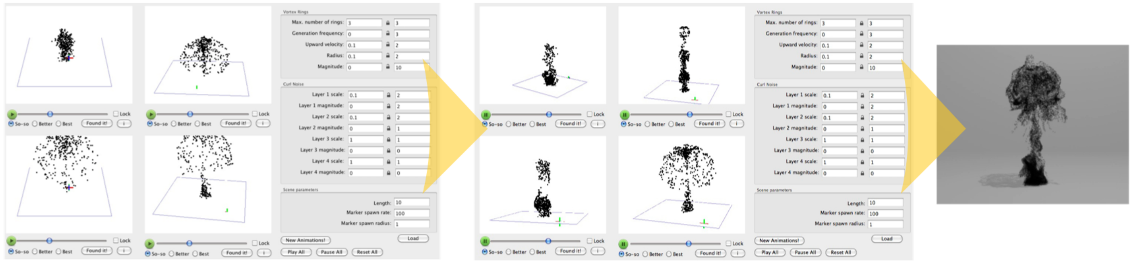

Figure 2.5: A machine-learning based optimisation system for animation design. Four smoke animations are represented in the gallery each time with a ranking option attached below each animation. The panel on the right can be used to manual adjust animation parameters if needed. The new animations displayed next in the gallery will be based the updated animation parameters. Image from [6].

using the system (see Fig. 2.5). The more the system is used by a particular user, the better and more accurate the system can learn the user's preference.

Modify Solutions.
Modifying solutions is one of the most straightforward interaction approaches to involve users in the optimisation process. It has two main purposes. First, it allows users to directly apply their domain knowledge to improve a solution produced by an optimisation system. This is one way to address the solver performance issue from fully-automatic optimisation. The other approach is to support user guidance of the optimisation search process as mentioned before. Second, it provides users with the ability to analyse the solution by making changes.

Many human-in-the-loop optimisation systems support this kind of interaction to allow users to directly modify a solution [4,6,7,17,40,41,47,49,50]. In particular, Belin *et al.* [7] developed a human-in-the-loop optimisation tool for urban city design (see Fig. 2.6). Users were free to perform changes to the solution while the optimisation solver was kept alive to re-optimise the solution. Users could directly move an urban shape from one location to another. A urban shape represented the basic building block for a city. For example, it could be a park or a school. However, to prevent the solver from re-optimising the entire solution, a radius around the urban shape was defined to constrain the degree of re-optimisation caused by users' changes.

Many scheduling and planning applications allow users to make changes to a solution [30,43,51,52] such as assigning new tasks and moving existing tasks.

2.2.2 Current Issues in Human-in-the-loop Optimisation

Human-in-the-loop optimisation has been used in a wide range of applications: facility layout [4,53,54], vehicle routing [3,5,55,56], user interface design problems [40,50], animation design problems [6,57], radiotherapy treatment planning [38], vehicle scheduling and planning problems [43,58,59], image segmentation [18,60] and composition [61], daylight performance problems [62], environmental management [63] and many more.

Most of these human-in-the-loop optimisation application papers are focussed on algorithms or systems and tool designs. For instance, a recent survey [1] focusses on the reasons and role of interactive visualisation, solver techniques and user preference gathering, but is almost completely silent about visualisation and interaction techniques, the user experience or user studies evaluating systems. However, there are some exceptions. An early review from Jones [64] looked at visualisation usage in optimisation while Miettinen [65] surveyed visualisation used in multi-criteria optimisation.

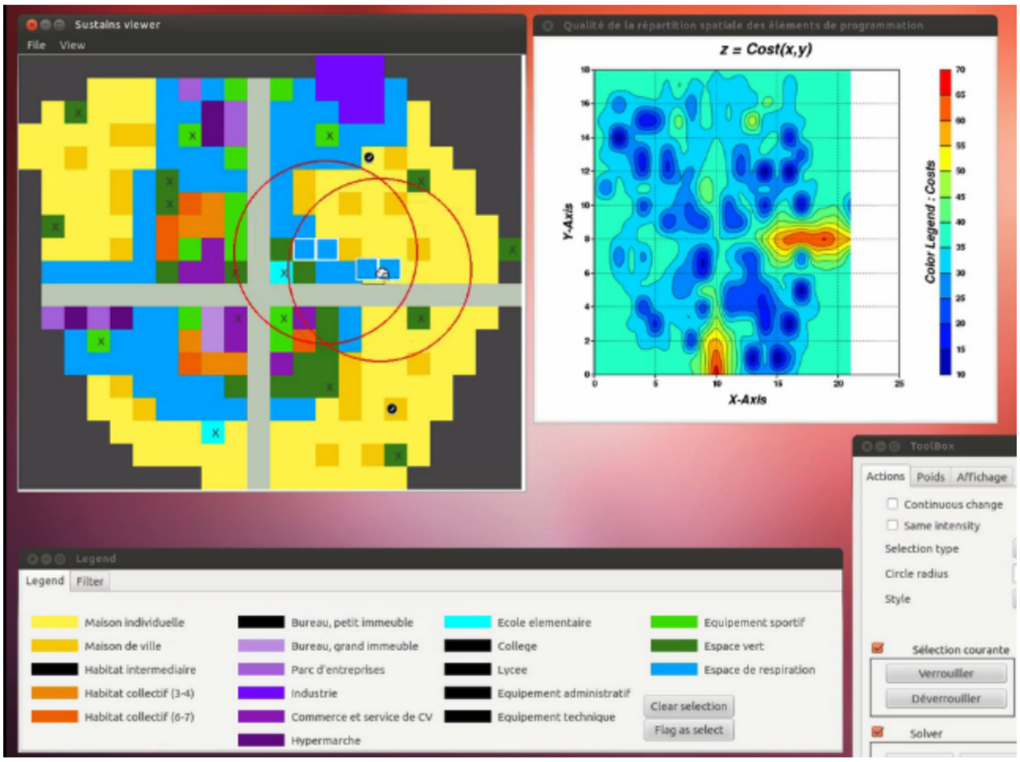

Figure 2.6: The human-in-the-loop application for urban city planning. It allows users to swap urban shapes and a heatmap is used to give feedback to the users about the impact of the swap changes. Image from [7].

One area that has received attention is the use of visualisation when debugging and profiling optimisation models. For instance, Goodwin *et al.* [66] investigated requirements for visualisation in profiling of search in constraint programming (a specific approach to optimisation), thus attempting to open up the 'black box' of the optimisation search process. Closely related are interactive visualisation tools for determining the best choice of parameters for image segmentation [67], simulation-based design [68], animation [69], weather forecasting [70] and sensitivity analysis for car engine design [71]. Sedlmair *et al.* [72] provided a conceptual framework for visual parameter space analysis for simulation and design.

A number of user studies have evaluated human-in-the-loop optimisation systems [3,4, 6,19,30,38,40,51,56,57]. In particular, Anderson *et al.* [3] reported that better routes could be found by using the human-in-the-loop optimisation system HuGSS to solve a vehicle routing problem. Bailly *et al.* [40] conducted a user study to compare the performance of manual and human-in-the-loop optimisation in menu design and found that using human-in-the-loop optimisation can reduce the editing effort to achieve equally good menu designs. Thieke *et al.* [38] developed an optimisation system that allows doctors to explore the Pareto front for multi-criteria treatment plans. They concluded that faster treatment planning is achieved.

However, virtually all user studies have evaluated human-in-the-loop optimisation in terms of solution quality and time spent to find solution aspects: usability and user satisfaction is not considered. An exception is the study by Patten and Ishii [4] described above evaluating the usability of their tangible human-computer interface Pico.

An even bigger gap is that the possible effects of interaction on *trust* have, to the best of our knowledge, never been explored in a user study for human-in-the-loop optimisation. This is in contrast to studies investigating user trust in recommendation systems [73, 74], adaptive agent systems [75], information security classification [76] and classification based on machine learning [77].

Another limitation of previous user studies is that very few use experts from a particular problem domain. Exceptions are Butler *et al.* [51] which asked professionals to develop flight-line maintenance schedules and Thieke *et al.* [38] which used clinical planners to evaluate treatment plans for two clinical test cases: a paraspinal case and a prostate case.

2.3 Visual Analytics

While user experience has not been a focus of attention in human-in-the-loop optimisation, it has been for visual analytics. Visual analytics combines automatic analysis with interaction and visualisation techniques to gain knowledge from data [9]. It is now a widely used tool for decision support.

Back in the early 1970s, visualisation and interaction techniques were not used in traditional statistics and data mining research. Then in 1977, Tukey [78] suggested moving from the traditional confirmatory data analysis which focussed on visually presenting results to what is called exploratory data analysis to support users' interactions with results. Later on, researchers in the field of information visualisation realised the potential use of visualisation techniques to enhance the knowledge discovery and sensemaking process by allowing users to visually interact with data.

Then in the early 2000s, in responding to the threat of terrorism, new interactive visualisation technologies were developed to aid intelligence analysts and then scientists and business analysts. In particular, there was a realisation of the need to develop more effective technologies to handle the problem of information overload, which led to a tremendous loss of time and money in industry and business because of the incapability to deal with the over-sized data volumes in a proper way [9]. Also, such technologies were required to support human judgement and communication with others.

Even though there existed powerful fully-automatic tools for data analysis, these can only work reliably when the problems were well-defined [9], which was not always the case in reality. When the problems were ill-defined or needed further understanding or clarification, human judgement plays an important role in handling such situations. Moreover, fully-automatic tools lack the ability to communicate the results and knowledge to users. This ability is critical in data analysis.

In 2004, the National Visualisation and Analytics Centre (NVAC) was supported by the US Department of Homeland Security (DHS) to define an agenda to facilitate more advanced analytical insight [20]. The fast-growing research field of information visualisation had caught their attention. It had shown novel techniques to support useful visual data presentations. However, information visualisation was not enough to address the requirements of big data analytics.

A new science called Visual Analytics was proposed by Thomas and Cook with the agenda of "analytical reasoning facilitated by interactive visual interfaces" [20]. Thomas and Cook defined visual analytics as a multidisciplinary research field focussing on four areas: analytical reasoning techniques; visual representations and interaction techniques; data representations and transformations and techniques to support production, presentations; and dissemination of the results [20]. First, the use of analytical reasoning techniques enables human judgement and analysis to gain insights from data. Second, the idiom "a picture is worth a thousand words" suggests the use of visualisation to allow users to see information by taking advantage of human vision. As mentioned above, interactions are necessary to support users' exploration of information. Third, data needs to be aggregated and transformed to eliminate conflicts and ambiguity to better support visual representation and analysis. Last, generating and understanding the results is not the final step.

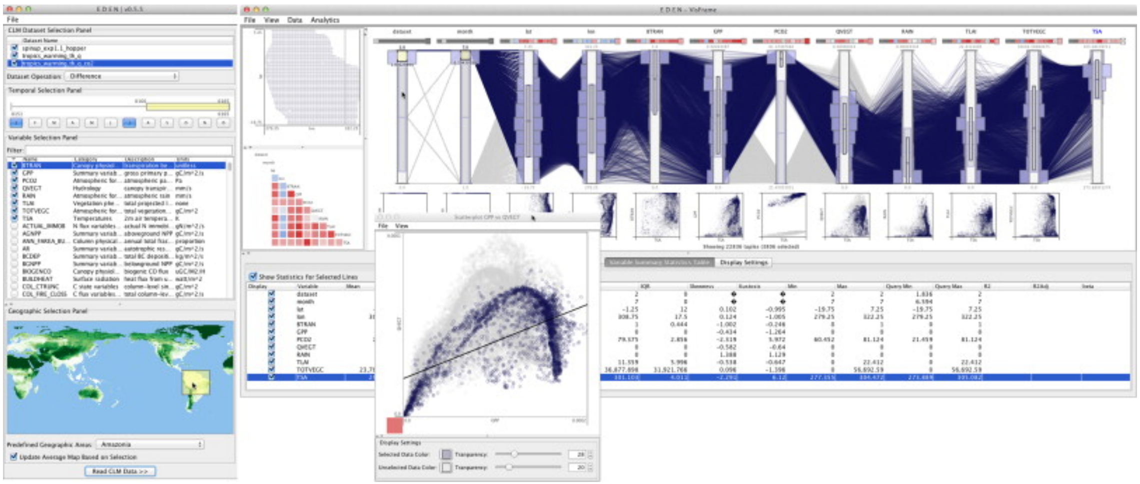

Figure 2.7: The Exploratory Data analysis ENvironment (EDEN), a unique visual analytics system to support climate scientists to address the need for exploratory analysis of "big data" in today's climate science. Image from [8].

It is critical to share and communicate the results with stakeholders and other audiences from a variety of different backgrounds.

By now, visual analytics has been intensively studied and there has been considerable progress. It is now widely used in application areas, in which a large amount of information has to be processed and analysed. For instance, climate and weather monitoring is a domain in which huge amounts of data are collected by sensors and satellites. Steed *et al.* [8] developed a novel visual analytics system, called the Exploratory Data analysis ENvironment (EDEN) to analyse sophisticated earth system simulation data. The system is shown in Fig 2.7. The system is specialised in supporting visual filtering and exploratory data analysis. Specifically, it allows users to interactively explore and filter the data using parallel coordinates, which is complemented with scatterplots to provide additional details for the identification of patterns in the data as well as a correlation matrix to offer visual feedback of the relationship between variables. Diehl *et al.* [70] proposed an interactive visualisation interface for weather forecasting to allow users to detect trends and identify errors in forecast models.

Visual analytics has also been widely used to solve design-based parameter tuning problems. This is closely related to design optimisation. Coffey *et al.* [68] developed and presented an interactive interface to support engineers in simulation-based design. Instead of using the conventional parameter fine-tuning method, the interface offered users with a more intuitive and user-guided approach to solve the design problem. Users can directly navigate simulation design to see the effects of simulation changes while the system correspondingly adjusts the underlying parameters. In animation design, Bruckner and Moller [69] implemented a visualisation system to help graphic artists design animations with instant visual feedback to open up the "black box" of typical design simulation parameter adjustment. In car engine design, Berger *et al.* [71] proposed a technique to facilitate multiple parameter driven sensitivity analysis to allow interactive exploration of parameter regions.

The Relationship between Visual Analytics and Human-in-the-loop Optimisation.

Both visual analytics and human-in-the-loop optimisation aim to leverage the complementary strengths of humans and computers to solve difficult real-world decision problems.

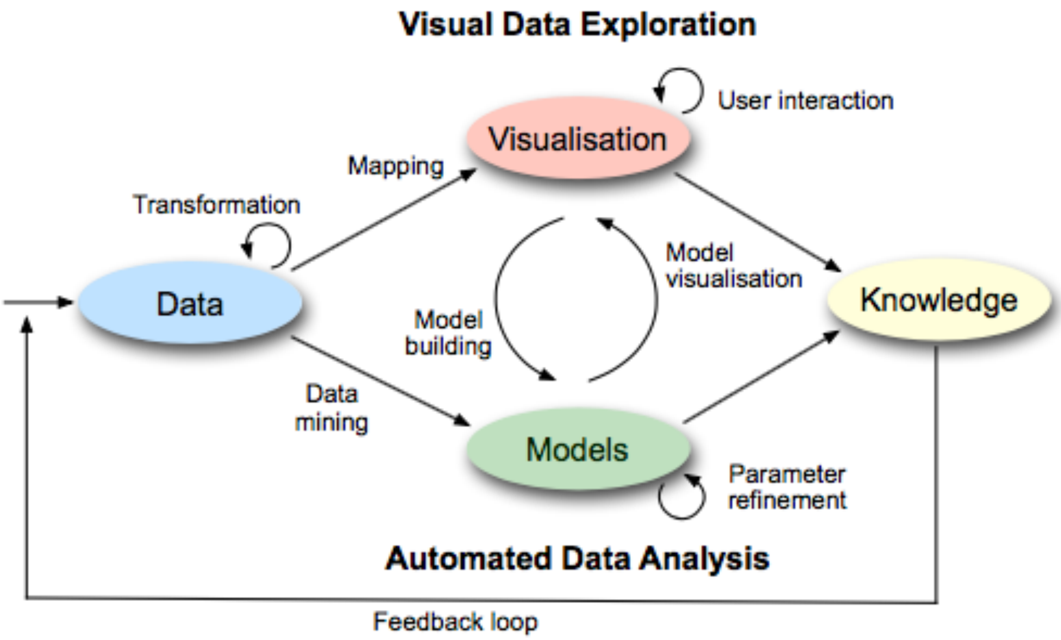

Figure 2.8: The visual analytics process presented by Keim *et al.* [9]. It is a knowledge discovery framework which involves the users, interactions, visualisation and data models. Image from [9].

This similarity of intent brings opportunities to share and transfer experience and knowledge between the two fields. In particular, creating effective visual interfaces for humans to control and explore optimisation can be seen as an application of visual analytics. Yet, as we discuss above in Section 2.2.2, this is a subject that has received little attention from optimisation researchers.

Many similarities exist between visual analytics and human-in-the-loop optimisation. Visual analytics aims to handle the information overload problem and to make information processing transparent [9]. In optimisation as well, many candidate solutions can be generated when problems get complex. However, effective human-in-the-loop optimisation allows users to explore only the ones that interest the user (the ones refers to candidate solutions from the previous sentence). Also it can open the “black box” of the optimisation solving process, which potentially gives users further insight into the solution space.

Keim *et al.* [9] presented the visual analytics process shown in Fig. 2.8. It combines automatic and visual analysis by providing user interactions to gain knowledge from data [9]. After initial data processing, automatic analysis and visual analysis can be combined in a complementary and powerful way. Automatic analysis allows analysts to refine the parameters of a model built through data mining. Visual analysis gives the freedom to analysts to explore the data through interactive visualisation.

The combination of machine processing and human inspection in human-in-the-loop optimisation can be seen as analogous to the visual analysis process. In human-in-the-loop optimisation, the initial solutions are usually generated by fully-automatic optimisation solvers. With the use of interactive visualisation, users can modify constraints and objectives from the mathematical models and ask optimisation solvers to re-solve the problem. This is similar to the role of the human analyst in visual analytics.

The high-level user goals and tasks supported by interaction are also similar in both disciplines. The user may:

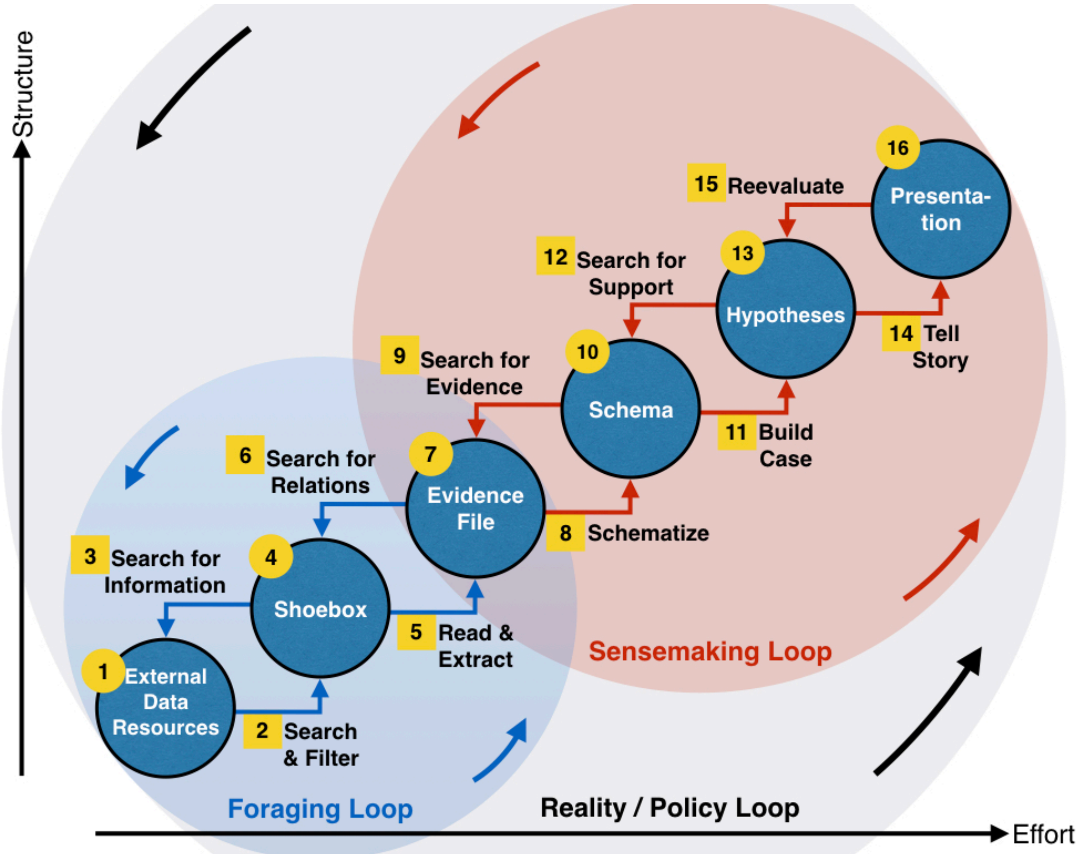

Figure 2.9: Pirolli and Card's sense-making loop.

- interact with the system to bridge the gap between the inherent limits of a mathematical model and the real world problem;
- guide the search, either for a solution or for the right model and parameters;
- use interaction and visualisation to build their trust in the model or solution;
- use interaction and visualisation to find patterns and to understand complex and large datasets (in optimisation this is the solution space, fitness landscape or the solver search space);
- communicate the result to managers, policy makers and other stakeholders.

Therefore, human-in-the-loop optimisation can be seen as an application domain for visual analytics. Conversely, model fitting and classification can be regarded as a particular kind of optimisation problem. Visual analytics applications utilising machine learning or data mining may therefore be regarded as a kind of human-in-the-loop optimisation.

Pirolli and Card's Sense-making Loop.
Visual analytics researchers have carefully examined the underlying tasks and goals of analysts in order to guide the design of better visual analytics tools. As previously mentioned in Chapter 1, *sense-making* is a widely used theoretical framework for understanding visual analytics tasks. It identifies the following steps in the analytical reasoning process:

- Information gathering (or foraging) to find relevant information
- Reformulation of the data to aid analysis

- Development of insight by interactive exploration of the data
- Formalisation of this insight by fitting schemas or models to the data
- Generation of hypotheses based on these schemas or models
- Presentation of the findings

For instance, Pirolli and Card's sense-making loop (see Fig. 2.9, also presented in Chapter 1), captures the processes employed by intelligence analysts when making sense of information. Blue circles in the figure represent data whereas arrows indicate processes. An important observation is that the sense-making process is not a straight progression from one to another. It has many loops. In each step of the process, the analyst may revisit earlier steps in light of new insights. For example, after collecting evidence and constructing an evidence file, the analyst may need to go back to earlier steps to search for more information and collect new evidence. There are two particularly important loops: the foraging loop and the sensemaking loop. In particular, the foraging loop is about data collection and data processing, while the sensemaking loop focusses on building hypotheses and presenting the results. To the best of my knowledge, there is no analogue of the sense-making loop for human-in-the-loop optimisation.

2.4 Conclusions

As an important decision support tool, optimisation has been used in a wide variety of application areas including rostering, scheduling, routing and design. Many application areas use fully-automatic optimisation techniques. In practice, many applications are nonlinear or discrete optimisation problems for which finding the global solution is impractical. In such situations, the solution quality or solving speed of a fully automatic system can be unpredictable. Moreover, many real-world optimisation problems have more than one objective. Determining how to balance these typically conflicting objectives is problematic in fully automatic systems. More generally, the mismatch between the mathematical model and the real-world problem is a significant limitation. Yet another limitation of fully-automatic optimisation is non-acceptance and misunderstanding of optimisation solutions and systems by the user because of its "black-box" nature.

Human-in-the-loop optimisation can address these limitations by involving users in the optimisation solving process. To overcome the solver performance limitation, human-in-the-loop optimisation allows users to guide the optimisation search and to modify solutions. In order to overcome the second model-problem limitation, human-in-the-loop optimisation provides users with the ability to add and adjust constraints or objectives. When the objectives cannot be precisely defined, letting users subjectively rank solutions may work. Interaction may also allow the user to better calibrate their trust in the solver.

Usability, user experience and trust are clearly key factors affecting the uptake of human-in-the-loop optimisation systems. However, such considerations and other HCI aspects in general such as identifying high-level goals and tasks have received little attention from optimisation researchers. Furthermore, another limitation of most human-in-the-loop optimisation research is that the evaluation does not include expert users.

In the remainder of this thesis, we address these two shortcomings. Our evaluation of a prototype human-in-the-loop optimisation tool for prostate brachytherapy in Chapters 3 and 4 includes seven domain experts from the field of radiation oncology. Based on their workflows, we develop a problem-solving loop which summarises the high-level user goals and processes in human-in-the-loop optimisation. Based on this, we present nine design guidelines in Chapter 5, which provide the foundation for building effective human-in-the-loop optimisation interfaces. Our investigation of trust in Chapter 6 provides insights into which factors affect trust, and how to engender the right amount of trust in a real-world optimisation tool.

Chapter 3

Prostate Brachytherapy: Context and System Design

As discussed in the introduction, our research methodology is to explain HCI aspects of human-in-the-loop optimisation through two case studies. The first case study is treatment planning for prostate brachytherapy. This is a treatment approach for prostate cancer. Prostate cancer is the development of abnormal cancer cells in a man's prostate gland. Prostate brachytherapy controls tumour cells in the prostate gland by the use of radiation. According to the World Cancer Report [79], prostate cancer is the second most common type of male cancer.

Prostate brachytherapy treatment planning determines the locations in the prostate gland to receive radiation. Current treatment planning is usually a careful, mostly manual process involving multiple radiation oncology professionals. We develop a human-in-the-loop optimisation tool for prostate brachytherapy that goes beyond current practice by supporting *focal* brachytherapy, a new treatment approach for prostate cancer. It is an alternative to the conventional method called *whole gland brachytherapy*.

In this chapter, we introduce prostate brachytherapy. Then we discuss the development of our human-in-the-loop optimisation tool in detail, including various design decisions during the development period, several big interface upgrades and so on. In the next Chapter 4, we evaluate the human-in-the-loop optimisation tool with several radiation oncology professionals using a two-stage semi-structured interview.

3.1 LDR Prostate Brachytherapy

There exist many different approaches to treat prostate cancer, such as surgery, hormone therapy, radiation therapy (also known as radiotherapy). Often, they are referred to treatment modalities by clinical professionals. Which treatment modality to use depends on various criteria, including the seriousness of the prostate cancer, the areas and organs affected by cancer cells, and patients' preferences. For instance, if tumour cells in the prostate cancer grow slowly and they do not spread from the prostate to other areas and organs, radiotherapy is probably more than enough to control the tumour cells. Whereas if tumour cells are more aggressive and have spread or are about to spread to other areas of the body, such as bones, a combination of multiple treatment modalities may be used together to eliminate tumour cells and to stop the cancer spreading. In this thesis, we only consider localised prostate cancer, in which tumour cells stay inside the prostate gland and have not spread to other parts of the body.

In localised prostate cancer, surgery (formally called radical prostatectomy), external beam radiation therapy (EBRT) and brachytherapy are common treatment modalities. EBRT and brachytherapy are two different forms of radiotherapy. Specifically, in EBRT, high-energy rays are directed by a machine to kill tumour cells from the outside of the patient's body. Whereas in brachytherapy, radioactive resources are implanted into the patient's body, the prostate gland, to kill tumour cells. This is why EBRT is also called external radiation and brachytherapy is named internal radiation. Clinical treatment results shows that the cancer-cure rates of brachytherapy are no worse than EBRT and radical prostatectomy for the use of a single treatment modality and the rates improves when brachytherapy is used in combination with other treatment modalities [80, 81]. In this thesis, we focus on brachytherapy.

There are two types of brachytherapy. Low-Dose-Rate (LDR) brachytherapy permanently implants well-sealed seeds containing radioactive resources into the patient‘s prostate gland using needle insertion. Because seeds permanently stay in the prostate gland, LDR brachytherapy is also named permanent brachytherapy. The other type of brachytherapy is High-Dose-Rate (HDR) brachytherapy, in which plastic catheters are temporarily inserted into the patient‘s prostate gland and a dedicated machine is used to deliver radioactive resources through catheters to the prostate. Catheters are removed after treatment. Unlike LDR brachytherapy, seeds are not left in the patient‘s prostate gland once the treatment is finished. Because of the temporary insertion of catheters, HDR brachytherapy is also called temporary brachytherapy. In this thesis, we focus on LDR brachytherapy.

3.1.1 Focal Therapy

Our application context is the development of planning tools for the treatment of prostate cancer using LDR brachytherapy in which radioactive sources are placed inside the patient's body close to or in tumours in order to control or kill the tumour cells. The radioactive isotope is encased and sealed inside a tiny cylinder-shaped titanium "*seed*". Multiple seeds are connected by a physical strand and assembled into a *needle*. Using the needle, the surgeon implants seeds into the patient's prostate gland where they remain permanently, though the radioactive strength decays exponentially over time. As shown in Fig. 3.1, a template grid is used to guide needle placement.

Whole gland therapy is the current standard clinical treatment method in LDR prostate brachytherapy. The aim is to give a good dose coverage to the entire prostate by uniformly distributing radiation to the prostate gland, while avoiding high doses to *organs at risk (OAR)*. In the case of the prostate, organs at risk include the urethra and the rectum. The urethra passes through the middle of the prostate; the rectum runs below the prostate (see Fig. 3.1). Whole gland therapy has great success in prostate tumour control and is supported by strong clinical evidence. However, whole gland therapy has the danger of over-treating patients if the number of tumours in the prostate is limited and the tumour size is relatively small.

For this reason, *focal therapy* has been proposed as an alternative to whole gland therapy. Instead of delivering a high dose to the entire gland, only those regions of the prostate with high likelihood of containing tumour cells are targeted for high-dose radiation. The potential benefit is a reduction of toxicity to other OAR especially the urethra and rectum, while still maintaining effective control of the prostate tumour cells. Approaches to focal therapy differ in the amount of radiation applied to regions of the prostrate deemed to be low-risk. Both the 'ultra-focal' and 'hemi-gland' approaches only partially treat the prostate gland with the rest prostate gland uncovered by any radiation. In the 'focussed' approach the entire prostate gland is irradiated by a small but sufficient dose, while the tumour regions are irradiated with high dose [82]. My research is part of an on-going project investigating the viability and effectiveness of focussed focal therapy. My role is to investigate how to best support radiation oncology professionals when creating a treatment plan with this therapy.

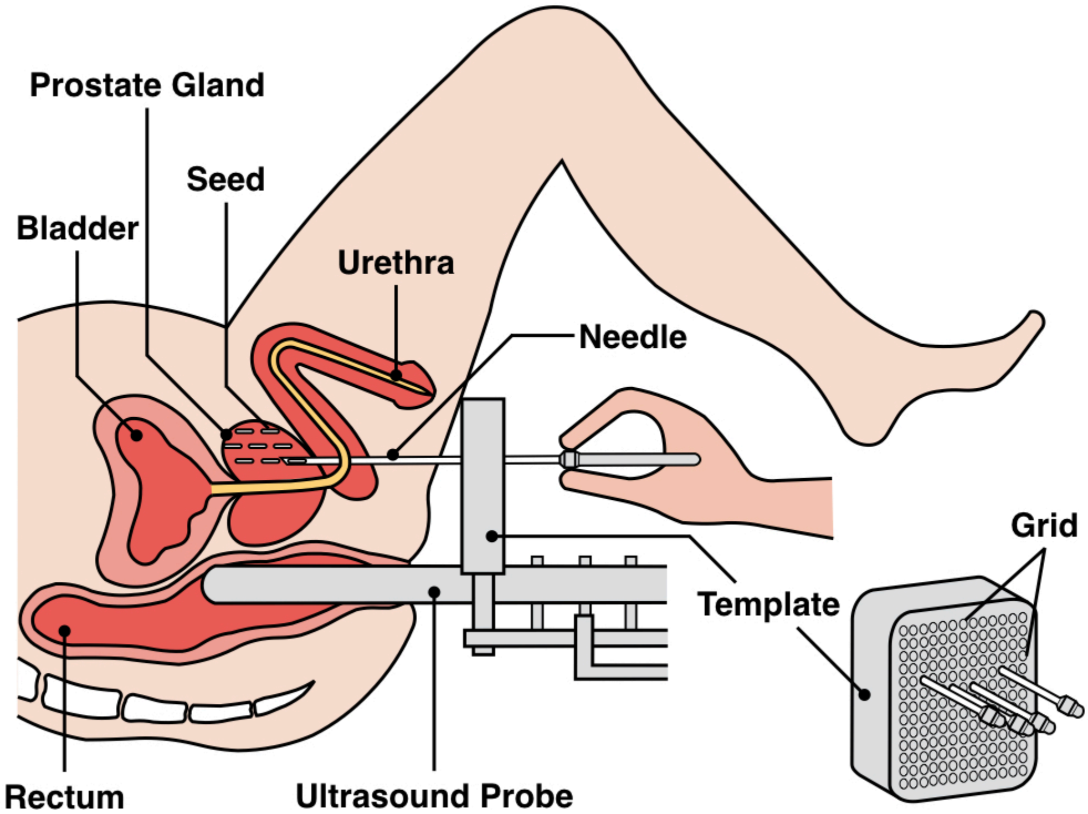

Figure 3.1: Needle placement in LDR prostate brachytherapy. Note the physical template grid used to guide needle placement. The front view of the template grid is displayed at the bottom right to provide more details.

Commercial treatment planning systems are used for treatment plan creation for whole gland LDR brachytherapy. *VariSeed* (Varian Medical Systems, Inc., Palo Alto, CA) is one of the most widely used. It provides manual planning of seed placement as well as a module for automatic planning. The manual approach, often referred to as *forward planning*, presents the treatment planners with an initial auto-seed-loading pattern based on a simple template automatically generated from a pre-defined planning protocol that specifies where seeds are placed on the grid points and also defines forbidden areas for the seeds. The planning protocols often vary between clinical departments. The treatment planners manually adjust the loaded template by adding extra seeds to increase dosage coverage, or removing or moving seeds to avoid overdosing of other sensitive structures. The automatic optimisation approach (also named *inverse planning*) requires the treatment planner to set planning criteria such as the dosage coverage and then automatically generates a treatment plan. However, at least within Australia, the manual approach is typically used for whole gland LDR brachytherapy planning. Automatic optimisation is used for intra-operative LDR brachytherapy, i.e., the plan is re-optimised while the patient is undergoing the treatment when the actual needle placement varies from the planned position. Automatic optimisation is used because of the requirement to update the plan in real-time making manual planning impractical.

Our approach to the development of treatment plans for focal therapy is based on a new mathematical model of biological characteristics of prostate tumour cells [83, 84]. The model introduces *tumour control probability (TCP)* as a biological planning objective that provides a relative measure of the likelihood of disease control following radiation. The calculation of TCP takes into account the radiation *dose* and the *tumour cell density (TCD)*, a special tumour characteristic measure, within the prostate volume as well as other tumour characteristics such as tumour aggressiveness and tumour hypoxia.

Because of the complex interaction between TCP and TCD, and the lack of clinical expertise, manually creating focal therapy treatment plans based on TCP had initially been regarded as impractical. As a result, automatic optimisation software based on the TCP model has recently been developed to produce focal treatment plans [85]. This aims to reach a desired TCP level while providing a safe level of dose to OAR. The optimisation software was effective, producing plans in a few minutes. However, as discussed in the introduction, unsupervised optimisation has a number of limitations including limitations of the underlying optimisation model itself, solving performance and lack of trust in automatically generated solutions. Therefore it was clear that a human-in-the-loop-optimisation approach potentially offered several benefits over a black-box approach:

- Treatment planning requires multi-criteria optimisation as high TCP conflicts with low dosage to OAR. Human radiation oncology professionals are better able to manage this tradeoff by taking into account additional knowledge, e.g. patient age or previous medical history.

- The model is not guaranteed to find an optimal solution as it uses an incomplete local search to find a solution above the desired TCP value. Humans can help guide the search to find better solutions.

- Finally, when the consequences of a poor optimisation can be extreme, clinicians are rightfully cautious in mistrusting the results of automatic optimisation. This is compounded by the fact that focal therapy and TCP are unfamiliar. Focal therapy is not the standard treatment approach for LDR prostate brachytherapy. Also, TCP is a new concept introduced into focal therapy to model the relative tumour control. Focal therapy is still under its development. It has not been widely used in clinical practises. Interaction potentially allows clinicians to build a better understanding of TCP and confidence in the underlying biological model as well as the underlying optimisation.

3.2 The Prototype human-in-the-loop Optimisation Tool

In order to evaluate whether human-in-the-loop optimisation is superior to manual or fully-automatic optimisation for TCP-based focal prostate brachytherapy, we developed a prototype human-in-the-loop optimisation tool. The tool builds upon a local-search-based optimisation solver previously developed to test a fully-automatic approach to focal therapy using TCP. The user interface for this new human-in-the-loop tool, and the required interfacing to the algorithm, took more than a year to develop using a participative design process with a medical physicist and bioengineer.

3.2.1 The First Version

In the early development of the tool interface, we focussed on building a 3D model for the prostate gland. We thought that with a 3D model, we could better visualise the radiation coverage of the prostate gland, from which clinical professionals could readily adjust needle positions and seed placement. Therefore, we used three.js, a cross-browser JavaScript library, to construct the 3D model of a prostate gland based on ultrasound biopsies provided by our project collaborators. Later on, we also built 3D models for other related structures to the prostate gland, including a urethra in the centre of the prostate and the *planning target volume (PTV)* defining the permit region for seed placements.

We were informed by our collaborators that it was still necessary to provide the standard 2D cross-sectional views of the prostate for treatment planning, similar to the biopsies collected from an ultrasound scan. The vertical views of the prostate were based on the provided ultrasound biopsies because the orientation of the prostate was the same. The front views and the side views of the prostate were based on the constructed 3D model. We were recommended to use the standard anatomical terms of location to name the cross-sectional views: the vertical view was named the transverse view or axial view; the front view was called the coronal view; and the side view was the sagittal view.

In order to visualise the radiation dose, we initially used voxel visualisation in the prostate 3D model and pixel visualisation in its cross-sectional views because both were easy to implement. We also provided interactive histograms to support the visualisation (See Fig. 3.2). For instance, the user can select the range of dose. Then both the 3D model and 2D views update the dose visualisation to only display the dose within the specified range. Therefore, the user can visualise where high doses are given to which parts of the prostate gland to assist the evaluation of the treatment plan. Later on, the visualisation was improved after the implementation of the marching squares algorithm to replace pixels with isolines representing the same level of radiation dose and the marching cubes algorithm to replace voxels with isosurfaces indicating the same amount of radiation dose in 3D space. This improved interface is shown in Fig. 3.3.

3.2.2 The Second Version

After we completed the first version of the prototype human-in-the-loop optimisation tool, on the one hand, our collaborators were excited about the visualisation and the 3D model. They were particularly interested in the visualisation because they had not seen it anywhere else before. On the other hand, however, they were very cautious and conservative, concerning the practical use and acceptance of the visualisation. At the end of the discussion, they suggested to only visualise the predefined radiation dose and to redesign the interface to include four viewports: one for the sagittal view, one for the coronal view, one for the axial view and the last one for the 3D model.

We reimplemented our interface based on the suggestions. We created four viewports and placed sliders beside each of them except the viewport displaying the 3D model. The

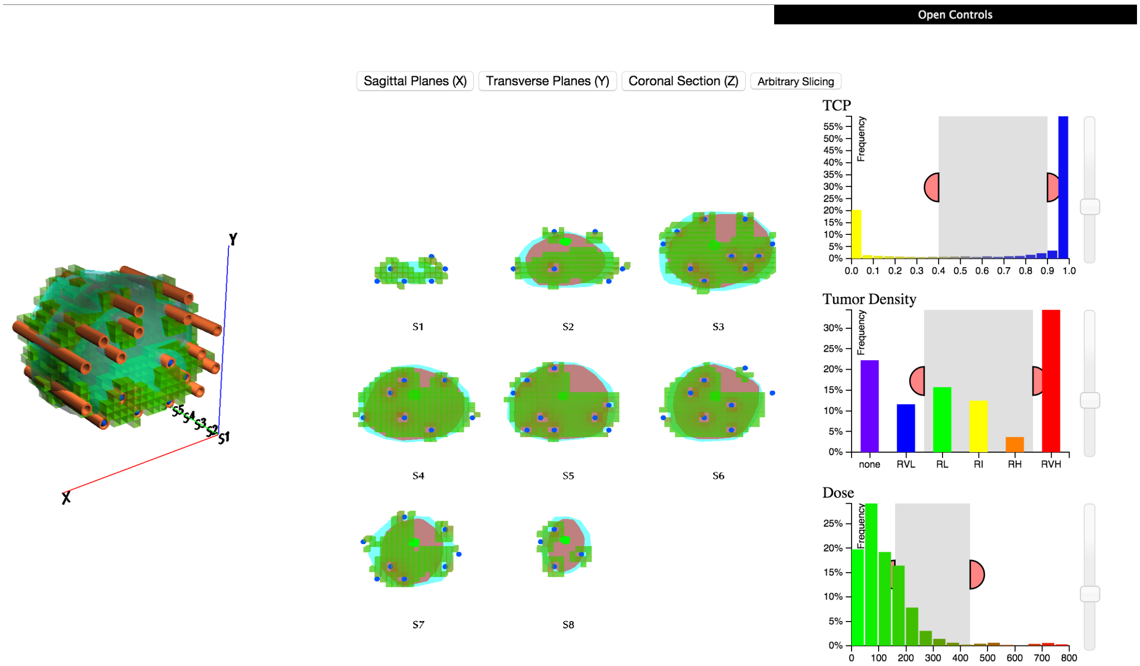

Figure 3.2: The early version of the prototype human-in-the-loop optimisation tool using voxel and pixel visualisation. The sliders beside the histograms are used to adjust the transparency of voxels and pixels.

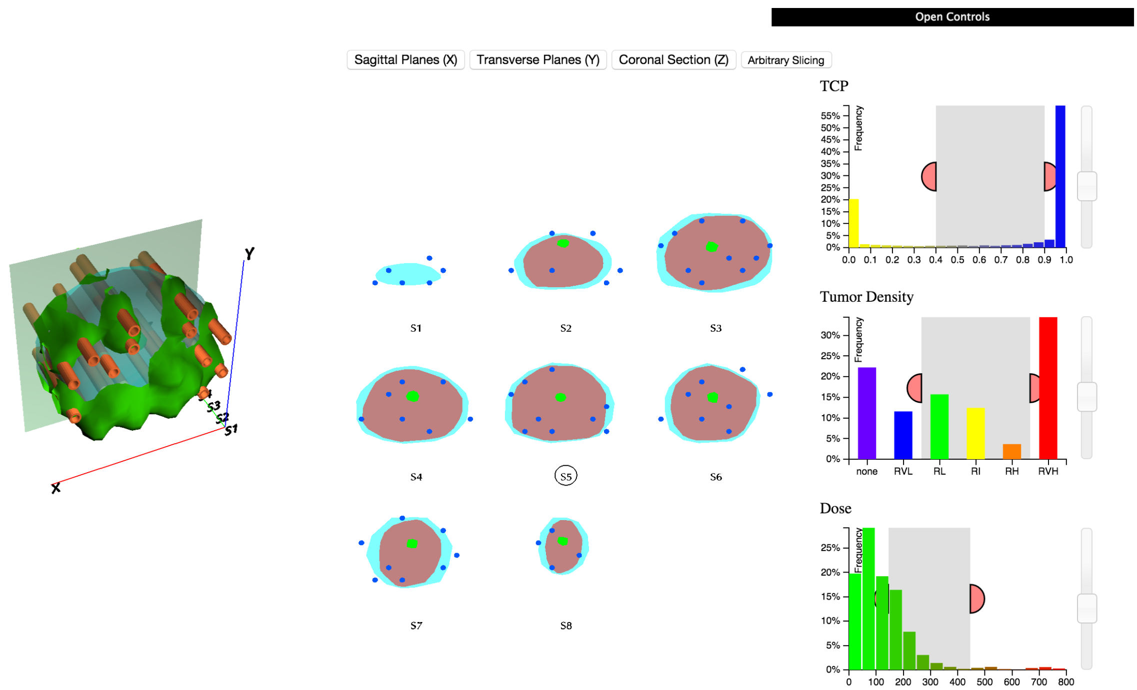

Figure 3.3: The improved version of the prototype human-in-the-loop optimisation tool using the marching squares algorithm and marching cubes algorithm. A cutting plane is added to the 3D model to indicate the location of a 2D slice.

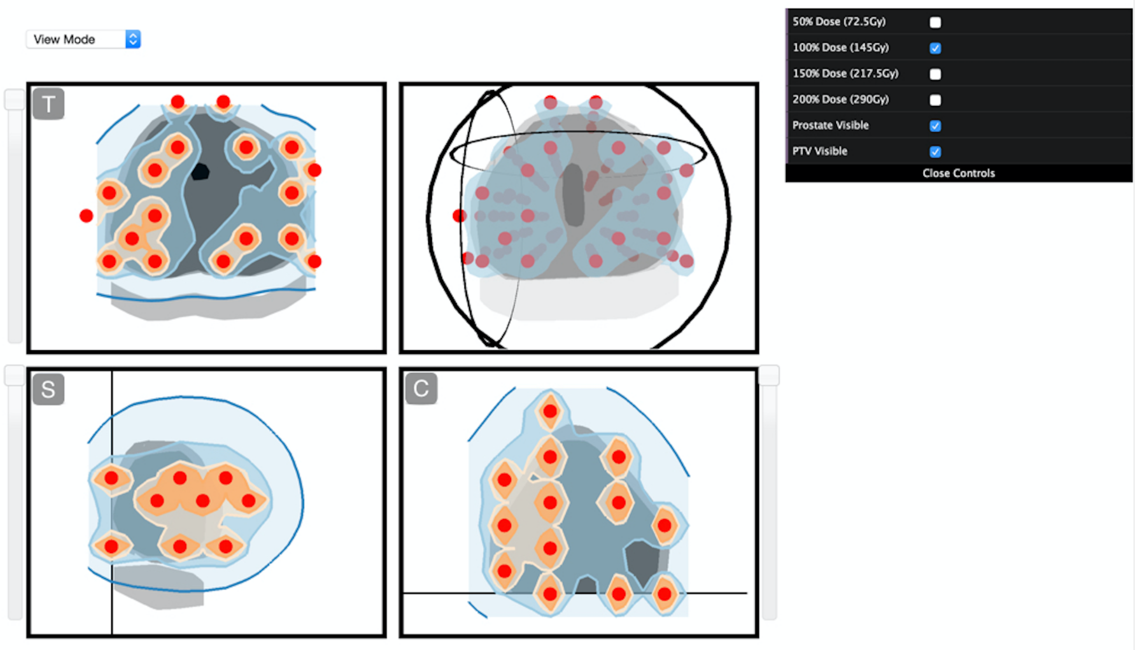

Figure 3.4: The updated version of our prostate human-in-the-loop optimisation tool. A single letter is placed at the top left corner of each viewport except the top right viewport for 3D model: T stands for transverse; S stands for sagittal; and C stands for coronal. The transverse view is also called the axial view. A panel is placed at the top right side of the interface to control the visibility of different radiation dose levels.

sliders were used to switch between different slices along the same direction. Because the axial view is the main view, we drew a black line in both sagittal and coronal views to indicate the current position of the axial view slice. Similarly, we drew three rings in the 3D model as the cutting planes to indicate the positions of the three 2D view slices. The updated interface is shown in Fig. 3.4.

After the completion of the interface redesign, we started to think about how to compare two treatment plans. The current interface could only present a single treatment plan using the four viewports. It did not work well to draw eight viewports for the purpose of comparison. As a result, we decided to use modes to differentiate the ways of using the interface. The four-viewport mode was the *presentation mode*, whose purpose was to visualise a single plan. While we called the new mode *compare mode* and redesigned the layout of the current interface to better support comparison of different treatment plans..

We used two viewports in the compare mode. One viewport was used to present different treatment plans. We called it *solution gallery*. It is interesting to think about the possibility to combine several solutions from the solution gallery. However, they may not be that useful for this particular prostate brachytherapy use case. The main reason is the symmetry property in a solution, which will be discussed in Chapter 4. We found that clinicians highly prefer to have seeds placed evenly and symmetrically on the left and right sides of the prostate gland. The line of symmetry is aligned with the urethra, usually located in the centre of the prostate gland. Also, there are constraints on seed placement. Two seeds cannot be placed too close to each other. Otherwise, the strong dose can hurt the patients tissue and cause a burning sensation. We considered the possibility of combining solutions in our second use case: interactive vehicle routing, which will be discussed in Chapter 5 in detail. However, we did not include and implement such interactions because of the complexity and time constraints.

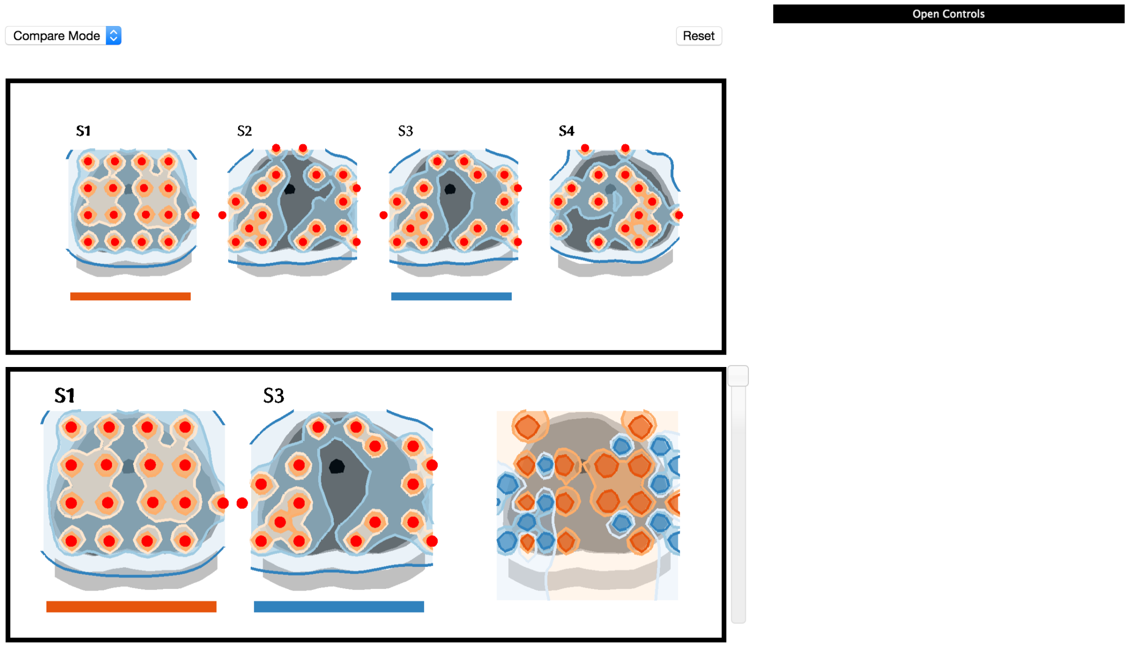

Figure 3.5: The compare mode of our prostate human-in-the-loop optimisation tool. The top viewport is the solution gallery showing different treatment plans. The bottom viewport contains the current slices from the two selected plans. The slices from the two plans been compared are synchronised. An orange rectangle is drawn under the slice of the first plan and a blue rectangle is drawn under the second plan. The dose difference map is coloured orange when the radiation dose level of the first plans is higher than the second. The map is coloured blue if the second plan has a higher dose level than the first. Colour saturation is used to indicate dose differences. If the colour is more saturated, the larger the dose difference is.

The other viewport was intended to show the similarities and differences between two plans. Specifically, we calculated the radiation dose level differences of both plans and drew contours to represent the differences using colour coding. When the first plan had higher radiation dose level than the second plan, the part of the prostate was coloured orange. Or when the radiation dose level of the second plan was higher than the first, the part was highlighted in blue. The bigger the difference was, the more saturated the colour was. We called the visualisation the *dose difference map*. It is shown in Fig. 3.5 and Fig. 3.6.

When we demonstrated the dose difference map in the compare mode, both our collaborators were impressed and very excited about it. They thought it would be very useful for radiation oncology professionals for treatment planning as they were always comparing multiple plans in order to select the best from the many. They had not seen similar visualisations in any other treatment planning systems that they used so far. However, the thresholds for the radiation dose level in the dose difference map needed to be carefully selected.

The new interface supported limited ways to interact with a treatment plan. Therefore, we created a *planning mode*, similar to the *presentation mode*, to allow clinical professionals to modify a plan by directly manipulating the position of a seed or the location of a needle. The movement of a needle resulted in all seeds within the needle moving to the new location. Because of the movement of the seeds, the radiation dose changed. We therefore updated all isodose contours in all four viewports in real-time to reflect the movement. A screenshot of the plan mode is shown in Fig. 3.7.

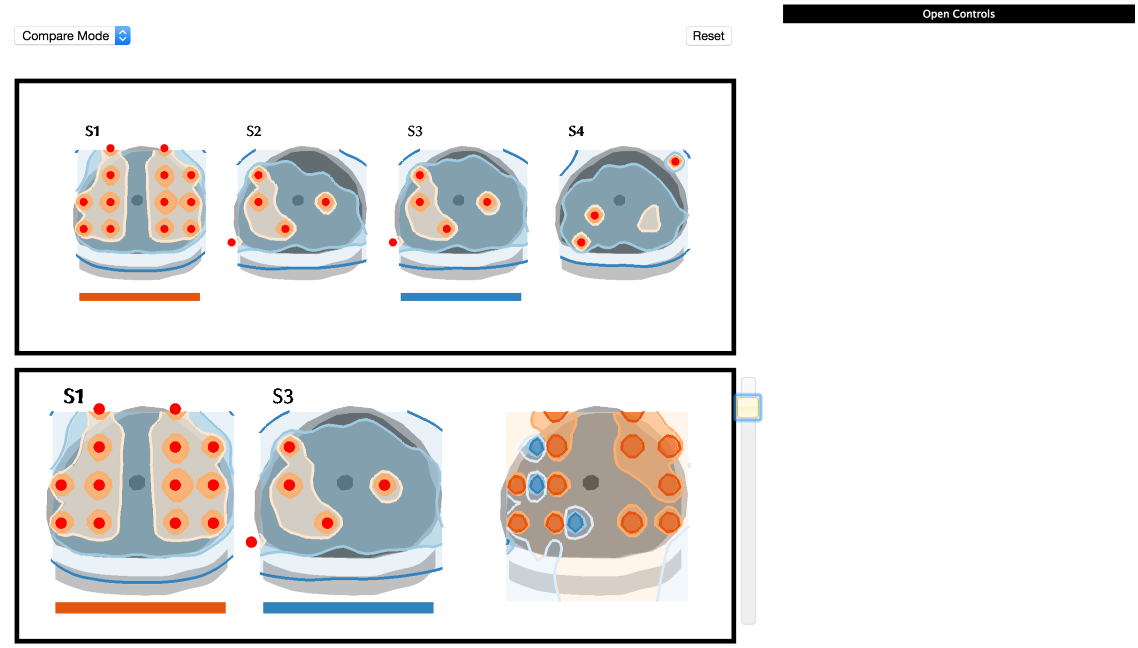

Figure 3.6: The same plans S1 and S3 are selected to compare with the dose difference map calculated and visualised based on another slices from both plans.

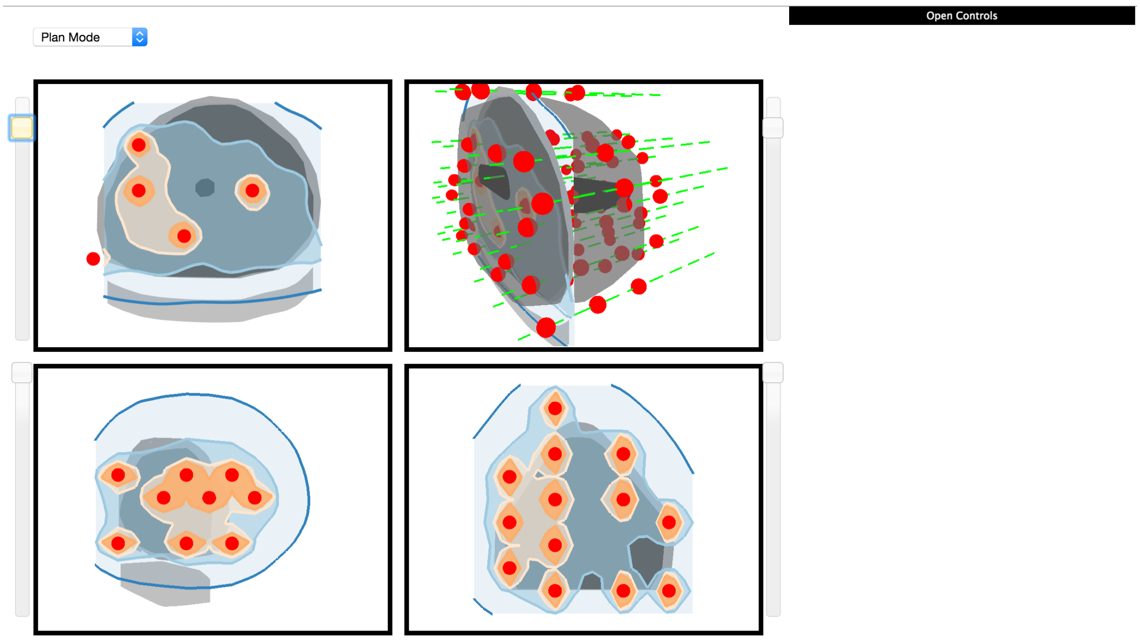

Figure 3.7: The plan mode of our prostate human-in-the-loop optimisation tool. Clinical professionals can adjust a seed placement by dragging the red sphere along the needle represented as a green dash-line in the top right viewport. The location of a needle can also be changed via drag-and-drop when hovering the mouse on the needle.

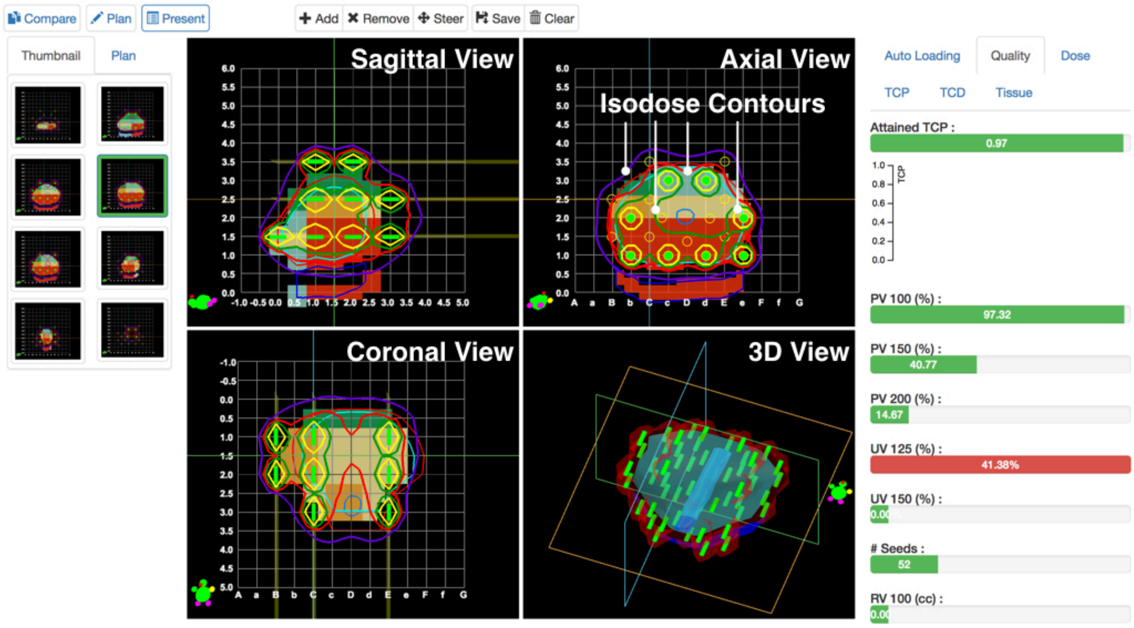

Figure 3.8: A sample treatment plan shown in presentation mode after Study 1 improvements. The white text is not part of the interface but added for explanation purpose.

3.2.3 The Third and Final Version

After discussion with our collaborators, they were satisfied with the three mode reimplementation. We were invited to join in a yearly brachytherapy workshop organised by the Alfred hospital in Melbourne, Australia to see clinical treatment planning practices and needle insertion simulations, which was beneficial for our interface design and implementation.

Based on our observations and communications with clinical professionals in the workshop, we redesigned our interface. We now describe the final tool in detail. This final tool also included participants' feedback from our Study 1 (details in the next Chapter 4).

Presentation Mode (Fig. 3.8): It took a number of attempts to produce a satisfactory visualisation of a single solution. In the first version, two view ports were used: one for the 3D prostate model and the other for *axial* views presented in small multiples. In the second version, after discussion with our collaborating medical physicist and bioengineer, *sagittal* and *coronal* views (longitudinal and cross section views) of the prostate were added to give a complete overview of a treatment plan. The eventual solution was to provide an anatomical plane of the prostate (the *axial*, *sagittal* and *coronal* views) and the 3D model in four linked view ports on a single pane. Grids were added in all anatomical views to show the physical grids (see Fig. 3.1) used for treatment delivery. As the axial view is the primary view used by radiation oncology professionals when planning treatments, we also included a complete gallery of axial view slices on the left.

Contours are provided for the prostate, urethra, rectum and the PTV defining the treatment margin around the prostate. The user can choose to overlay this with TCD, TCP or physical dose. We settled on an isodose contour representation (implemented using marching squares and marching cubes), similar to that used in commercial prostate brachytherapy planning tools. We also provided histograms on the right to summarise the dosage and planning objectives in our final version.

Compare Mode (Fig. 3.9): Being able to review and verify a single treatment plan is essential. However, we felt that it would also be useful to be able to *compare* and *rank* candidate solutions. This is an activity that is typically not well supported in optimisation

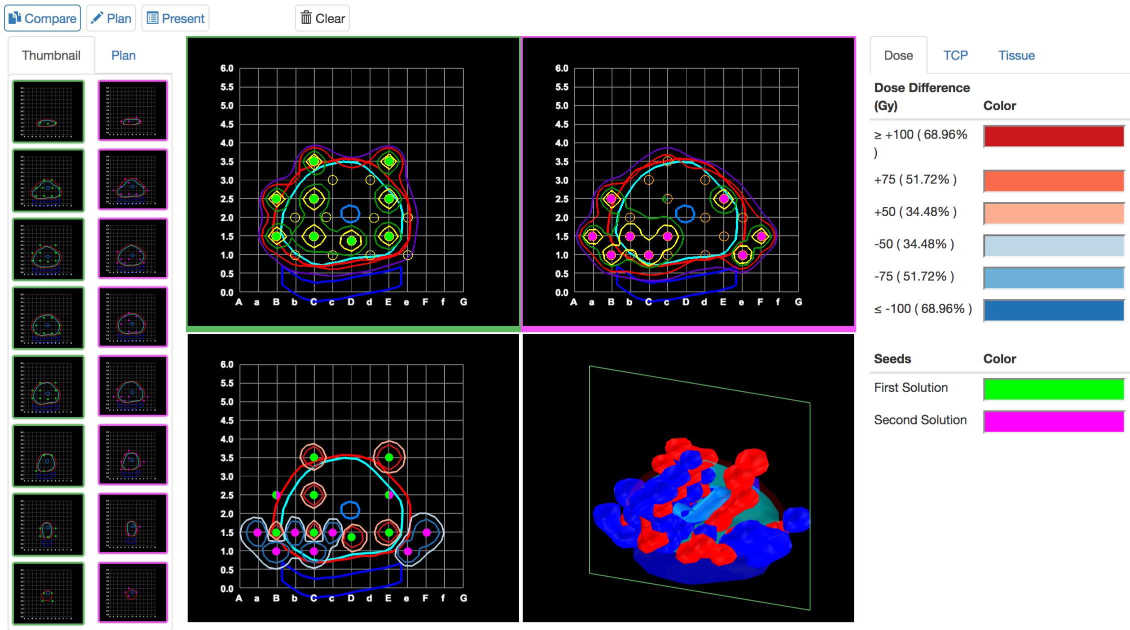

Figure 3.9: The compare mode. Top two view ports show the axial view slices of the two plans to be compared. Bottom two view ports show the difference map in 2D and 3D.

tools. Our tool provides a novel comparison mode in which the user can scroll through synchronised axial view slices of two solutions, with a third view port showing the difference between the two slices and a fourth showing a 3D visualisation of the difference. Like the presentation mode, the user can choose to show TCD, TCP, physical dose or a combination.

Initially, in the second version, we designed two view ports: a solution gallery for browsing all candidate plans and below a view of two candidate plans and their absolute dose differences. Our collaborators were very excited about this. However switching between slices caused significant delay because of the colour-washed difference representation. We replaced it with pixels to reduce the delay and keep the whole interface responsive. Later we changed the layout to more closely resemble the presentation mode, providing a more uniform interface and more flexible comparison with sagittal, coronal and 3D views as well as the difference in axial views. We used isosurface for the 3D visualisation of difference.

Planning Mode (Fig. 3.10): The ability to generate new solutions or improve an existing solution must be supported by any true human-in-the-loop optimisation tool. In our tool this is supported by the *planning mode*. Fig. 3.10 shows the planning mode before any plans have been created. In the figure, the user is viewing the TCD. They can use the pane on the right to generate a new solution. They can set values for the minimum desired TCP and the maximum allowed dose to both the urethra and the rectum, defined as three constraints ('prescribed physical dose' is declared as $145Gy$, *Gray (Gy)* being the unit used to measure the total amount of radiation a patient is exposed to):

1. *UV125*: Volume of the urethra receiving 125% of the prescribed dose ($145Gy \times 125\% = 181.25Gy$);
2. *UV150*: As above, but 150% of the prescribed dose ($145Gy \times 150\% = 217.5Gy$);
3. *RV100*: Volume of the rectum receiving the prescribed dose ($145Gy$).

By varying these parameters they can *guide* the solver by forcing it to generate solutions that have different tradeoffs between higher TCP and less irradiation to OAR. When the solution is generated a visualisation shows the solver's progress: this provides an indication of how difficult it is to find a solution meeting the specified criteria.

The planning mode also allows users to improve a solution by *adjusting/suggesting* changes to the placement of seeds and needles in the solution. Our tool provides both

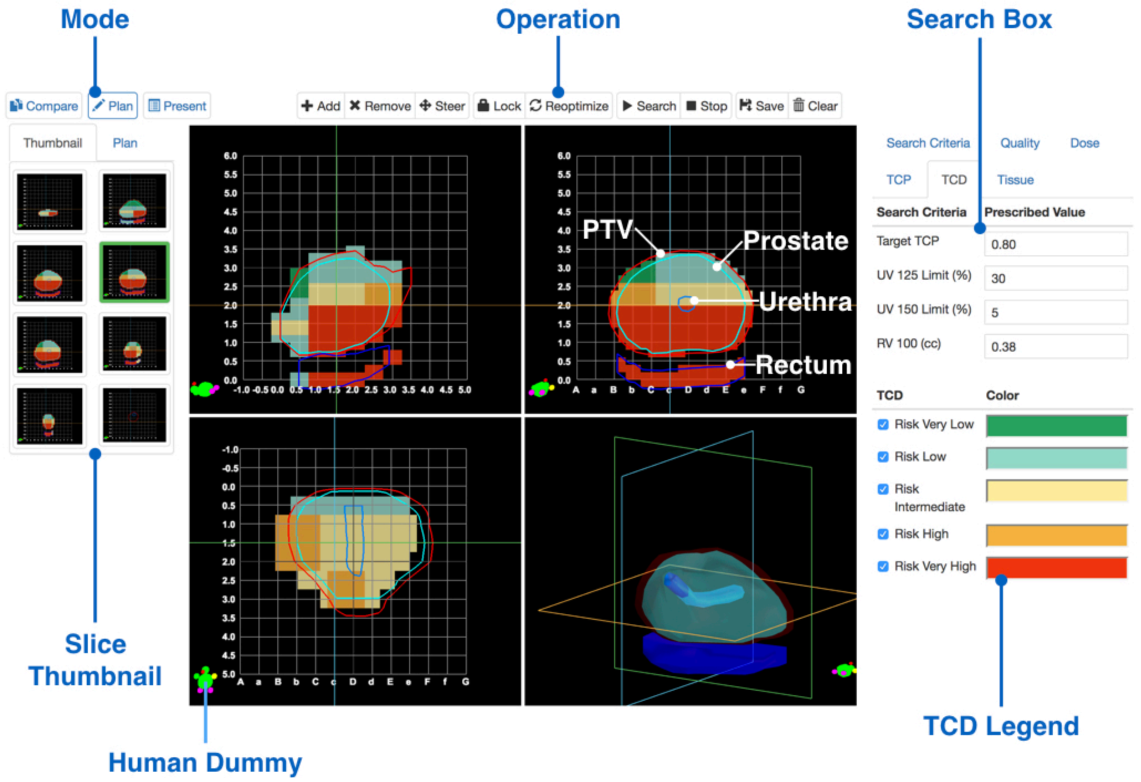

Figure 3.10: The final prototype human-in-the-loop optimisation tool. The prostate gland is divided by a grid with colour coding representing the TCD in each grid cell. A Human Dummy is used to indicate the patient‘s orientation.

manual and semi-automatic adjustment of solutions as we were interested in finding out which techniques would be preferred by radiotherapy team members. The following adjustments are possible:

Add: Users can add a new seed to a grid point in the current slice. If there is no needle at the target grid point a new needle is created.

Remove: Users can also remove seeds. If the seed to be removed is the last in the needle, removing the seed also removes the needle.

Move: Users can select and move a needle which also moves the seeds on the needle.

Lock & Re-optimise: Users can lock the position of seeds and needles whose position they are happy with. They can then run re-optimise which starts from the current solution and tries to improve it by moving the unlocked seeds. In combination with manual placement of seeds this is a powerful way of *steering/guiding* the solver to generate better solutions. By locking the position of some seeds the solver search space is reduced, speeding up the search for better solutions.

After users perform any of these operations, the treatment plan is refreshed and either physical dose or TCP contours are updated immediately to give users instant feedback.

3.3 Conclusions

We introduced our first case study: treatment planning for prostate brachytherapy. In particular, we focussed on the new treatment approach focal therapy, an alternative to the conventional whole gland therapy approach. Moreover, the focal therapy approach used a new mathematical model of biological characteristics of prostate tumour cells by introducing TCP, a relative measurement of disease control following radiation and TCD, a tumour characteristic measure.

We presented our prototype human-in-the-loop optimisation tool for treatment planning in three versions. The first tool was intended as a proof of concept tool to explore the possible visual representations for treatment planning. The second version focussed on redesign and reimplementation based on clinical requirements and practices. On the one hand, clinical professionals were open to new visualisations and interactions. On the other hand, they were also very conservative about its practical use. This made the development of the tool very challenging and time-consuming. Importantly, the success of developing such a tool was highly depending on the engagement with domain experts. We even participated in a clinical training workshop with doctors and practitioners. The final version was redesigned with two purposes: First, the overall looking of the tool as well as the details needed to look similar to what professionals have used to reduce their resistances. Second, the operations of the tool were enriched to include more interactions based on clinical requirements.

In the next chapter, we will present our evaluations of the tool using two studies.

Chapter 4

Prostate Brachytherapy: User Studies

We conducted two exploratory, formative user studies with radiotherapy team members to evaluate and inform the design of our prototype tool. We had originally planned a single study but based on feedback from our medical collaborators we decided to break the study into two parts because participants would have been overwhelmed by the need to understand focal therapy (recall from Chapter. 3.1 that the current standard method is whole gland therapy), TCP (proposed in [84] and not yet part of clinical procedure) as well as our novel and therefore unfamiliar human-in-the-loop optimisation tool.

The study protocols were reviewed and approved by the Monash University Human Research Ethics Committee (MUHREC). The user study project was titled "Investigating Human-in-the-loop Optimisation to Facilitate Solution Construction, Understanding and Trustworthiness". The project ID was CF16/2314.

4.1 Study 1: Whole Gland Brachytherapy

The first study was designed to present the tool in the context of treatment planning for whole gland therapy, a process they were extremely familiar with. Based on feedback from this study we could refine the design of the tool before conducting the second study investigating use of the tool for focal therapy planning.

This meant that we needed to modify the tool for the first study as traditional planning for whole gland therapy does not use TCP or TCD, only physical dose. The underlying solver was modified to optimise dose coverage rather than TCP and the interface of the tool was also appropriately modified to consider only physical dose.

Study 1 was designed to achieve three main aims:

1. Build a better understanding of the workflow of treatment planning;
2. Understand why radiation oncology professionals use a manual approach to create treatment plans rather than optimisation in whole gland therapy;
3. Evaluate and improve the prototype human-in-the-loop optimisation tool, in particular the visualisations and interaction design.

4.1.1 Study Design

The study was in the form of a semi-structured interview. Therefore, it was appropriate to make a few changes to the study based on users' feedback. We described the adjustments and explained why we did so in the next section Results & Discussion.

Apparatus & Materials: The modified human-in-the-loop optimisation tool was used throughout the study. Two whole gland treatment plans were prepared for the study.

The first was produced using automatic optimisation while the second was a plan created manually by a radiation therapist (not part of the study).

Participants: Seven participants were recruited for the study: 3 *radiation therapists (RT)*, *2 medical physicists (MP)* and 2 *radiation oncologists (RO)* after we gave an introductory presentation about this project at the Alfred Hospital in Melbourne, Australia. All participants completed the study except for one RO who did not do the plan improvement (due to time constraints). An initial pilot was conducted with a radiation oncologist who was a member of the research team. We have included the results from the pilot study (indicated as such) where relevant.

Procedure: Interviews were conducted at the participants' workplace. The study took about one hour on average. It had four parts:

1. *Introduction Activity:* The participants were asked about their experience in LDR brachytherapy, their current roles and responsibility in the brachytherapy team and the overall workflow, as well as their opinion on manually produced and optimisation generated treatment plans.

2. *Training Activity:* Next, our prototype tool was presented to participants. The operations of the tool were explained and demonstrated and participants were encouraged to ask questions and play with the tool until they were familiar with its operations.

3. *Planning Activity:* In the main part of the study participants were asked to evaluate two different types of treatment plan and to "think aloud" to explain the reasons and the evaluation criteria. They were not told how the plans had been produced. The two different types of treatment plan were:

   (a) A manually-produced plan by another radiation oncology professional using their usual procedure;

   (b) An automatically-produced plan (without interaction) using our solver software to optimise for dosage.

Then the participants were asked to improve the automatically produced treatment plan by using the operations introduced in the Training Activity. As before, they were asked to evaluate it and to give their reasons and evaluation criteria. The participants were also asked to give a rank to each treatment plan after evaluation.

4. *Recap Activity:* Finally, the participants were asked open-ended questions about the advantages and disadvantages of both manual and optimisation approaches to producing a treatment plan and to provide general comments about the prototype tool and suggestions for improvement. They were specifically asked to provide feedback on the comparison mode and the difference map visualisation as this was the most novel aspect of the tool.

4.1.2 Results & Discussion

Aim 1 (Understanding the treatment planning workflow) was investigated by asking participants to describe their roles and responsibilities in radiation treatment planning and to explain the process in the Introduction Activity.

The first step is that an RO contours the prostate gland and its surrounding organs and establishes the treatment margin based on the patient's ultrasound images. [1] This is the 'Volume Study' process. The initial plan is usually created by an RT and reviewed by an MP before sending it to RO for the final review. However, to balance workload, sometimes an MP will create the initial plan that is reviewed by an RT before the final review. Both MPs mentioned they were in charge of the treatment plan quality assurance process and indicated that it was common to ask for revision or for another plan to be produced. On occasion more than one plan is produced when it is unclear how best to trade-off dose

[1] A normal prostate gland is only roughly symmetric and a symmetric treatment plan may not be strictly the optimal solution. Furthermore, if treatment is post surgery then the remaining parts of the prostate gland are almost certainly highly asymmetric.

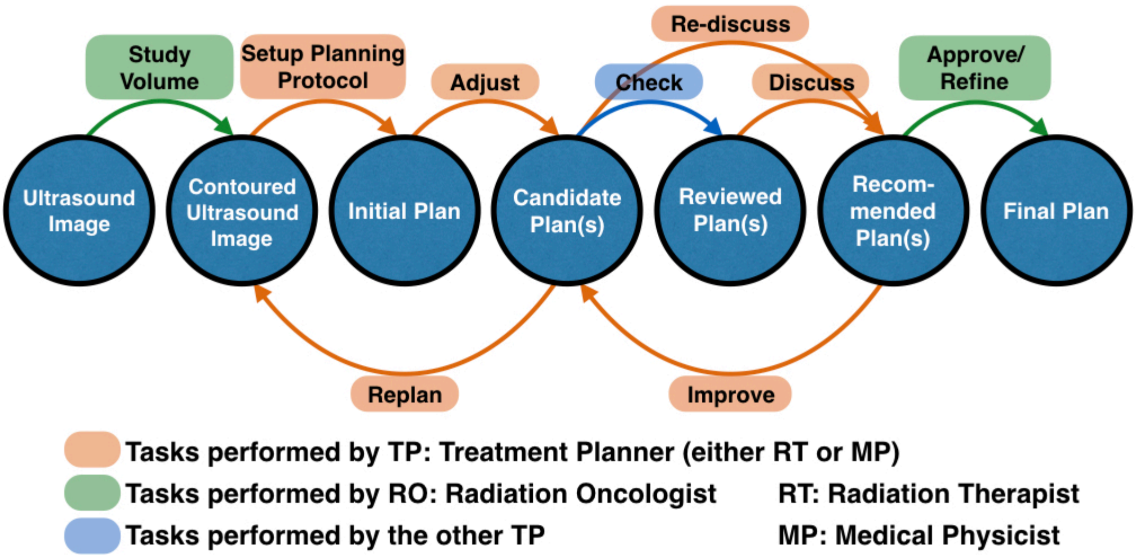

Figure 4.1: The workflow to create and approve a treatment plan.

coverage and protecting OAR. What is striking about the workflow is the importance placed on checking and validation of the plan. This requires significant collaboration between the clinical professionals. Fig. 4.1 summarises the treatment planning workflow and the different roles.

Aim 2 (Understanding why radiation oncology professionals use a manual approach to create treatment plans rather than optimisation in whole gland therapy) was investigated through questions in the Introduction and Recap Activities as well as the "think aloud" ranking and planning in the Planning Activity.

Our study revealed a number of reasons why manually produced plans were preferred. One unexpected reason was a desire for plans to be symmetric. During the pilot the RO strongly criticised the automatically produced plan for not placing seeds and needles symmetrically on either side of the prostate gland. This was intriguing so we slightly modified the study procedure to further investigate symmetry.

The tool is initially developed for the use of focal therapy, in which seeds are not symmetrically distributed. The tool is adapted later on to do whole gland therapy by evenly distributing the tumour probability across the entire prostate gland. At the early stage of the first whole gland therapy study, we did not realise the importance of seeds symmetry. But based on users feedback, we did refine our tool to add symmetric constraints so that seeds can be roughly placed symmetrically on both sides of the prostate gland.

Subsequent participants where asked 'Do you like to keep the plan symmetric? If so, why?' and we modified the optimisation solver to produce more symmetric plans. For the last **4** participants we added an additional automatically produced plan that was more symmetric to the study and asked the participants to rank four plans (the original asymmetric automatically generated plan, the more symmetric automatically generated plan, the manually modified automatically generated plan and the manually produced plan).

All participants agreed on the importance of symmetry and this was reflected in the rankings with the manual plan (which was symmetric) and the symmetric automatic plan receiving higher ranks by most users. The importance of symmetry was also reflected in participants' improvements to the optimisation plan with some participants replacing existing seeds and needles with new ones placed evenly on both sides of the prostate.

A number of reasons were given for preferring symmetric plans. Aesthetic preference was one. Symmetry of the prostate was another. One participant said:

> "*if it [the prostate] is symmetrical then it makes sense to have sort of the same kind of needle placement on both sides of the prostate.*"

A third reason was treatment speed and simplicity. It was felt that surgeons liked symmetric plans because they were easier to remember, meaning that treatment could be faster and less error prone. Symmetry was also said to make the plan more robust. Finally, we suspect that symmetric plans make it easier for planners to understand the plan and build a mental model from the slices. It also reduces the search space when creating a plan manually.

In addition to lack of symmetry, fully automatic generation of treatment plans was criticised for a number of other reasons. One reason reflects the multi-criteria objectives in whole gland therapy planning. There is a conflict between delivering sufficient dose coverage to the prostate gland and reducing the dose toxicity to OAR. Participants felt this balance required human judgment and decision making. Another reason is that different clinical centres can differ in the planning protocol for dose constraints as well as allowed placement of seeds and needles. This makes it difficult for automatic optimisation to produce acceptable plans for all clinical centres. Yet another reason was that the underlying mathematical model restricts seed placement to grid points. While needles are inserted through a grid template, in practice surgeons may insert them at an angle so as to steer seeds away from OAR or to take into account physical constraints such as the pelvis. Thus seeds may actually be placed off the grid. Other comments reflected that participants were simply comfortable with the current processes:

> "*[Forward planning (manual planning) is] the robust way of planning. [We've] got years of results behind it.*"

The responses clearly show that the current preference for manual planning over automatic planning in whole gland therapy is because of the mismatch between the mathematical optimisation model used in automatic planning and the actual real-world problem. This provides strong support for using human-in-the-loop optimisation rather than automatic optimisation in focal therapy planning.

Aim 3 (Evaluating and improving the prototype human-in-the-loop optimisation tool, in particular the visualisations and interaction design) was achieved by asking questions in the Recap Activity and from comments made by participants during planning. Changes were made to visualisations, interaction and to the underlying optimisation model.

We improved the visualisations as the study progressed in light of participants' feedback. The first change was to use the clinic-specific colour scheme for treatment planning. Next, we rearranged the four view ports in the presentation mode so as to accord with the layout provided in Variseed (the commercial prostate brachytherapy planning tool used in the clinic). We also adjusted the grid labelling and alignments of images in each view port to better fit with the current treatment planning workflow as well as to synchronise with the alignment of the physical template for treatment delivery.

One of the most novel aspects of our prototype tool is the compare mode. Generally participants were positive: one participant said:

> "*I think the comparison [where the difference map is used] could be really useful actually. That is one thing we are missing in VariSeed.*"

However some participants found it initially confusing:

> "*[It is] hard because [I have] never seen this before.*"

Another commented:

> "*I do not think it is hard to understand. I think it takes a little bit of getting used to looking at it.*"

In response to user feedback we: moved the axial view slices of the two plans to be side-by-side in the top two view ports for easier comparison; replaced the pixel-based display of differences with isoline contours because of participant familiarity with isodose line usage in treatment planning and, for consistency, replaced the point cloud 3D difference visualisation with isosurface contours. We also reduced the use of colour and emphasised large differences in the 3D difference model. However, while participants liked the 2D difference view they did not make much use of the 3D difference model. In general we found they relied on the 2D views much more than 3D models.

2D models are preferred because of participants' past experience in brachytherapy. They tried to use the 3D model, but they did not like it because they questioned its usefulness. The only usage of the 3D model was to check the dose coverage around the prostate gland at the end of treatment planning to make sure enough given to the entire prostate. This is related to the conventional whole gland therapy treatment approach, which does not seek to treat specific tumour locations. Whole gland therapy treats the entire prostate gland instead to make sure no tumours are missed out. This assumes the cancer is localised to the prostate, which means no tumours are located outside of the prostate gland.

Responsiveness was an issue raised by several participants. We removed the sliders along each anatomical view and replaced with scrolling and key stroke to support faster browsing of anatomical slices and improved responsiveness during planning by reducing refresh frequency for the axial view slices gallery, as this was viewed less frequently.

Finally, as discussed previously, we modified the underlying constraint model to produce more symmetric plans and also allowed the user to manually steer a needle away from a grid point to allow fine-tuning, better reflecting actual practice.

4.2 Problem-solving Loop

As discussed in the related work, visual analytics and human-in-the-loop optimisation share the intent to leverage the strengths of humans and computers to solve difficult problems. One way to understand the differences and similarities between visual analytics and human-in-the-loop optimisation is to compare the high-level processes and aims of the user in these two endeavours. Informed by the first study, we can now attempt this.

Sense-making is a widely used theoretical framework for understanding visual analytics tasks. It identifies the following steps in the analytical reasoning process:

- Information gathering (or foraging) to find relevant information
- Reformulation of the data to aid analysis
- Development of insight by interactive exploration of the data
- Formalisation of this insight by fitting schema or models to the data
- Generating hypotheses based on these schema or models
- Presentation of the findings

For instance, Pirolli and Card's sense-making loop (Recall from Fig. 1.2) captures the processes employed by intelligence analysts when making sense of information. To the best of our knowledge there is no analogue of the sense-making loop for human-in-the-loop optimisation.

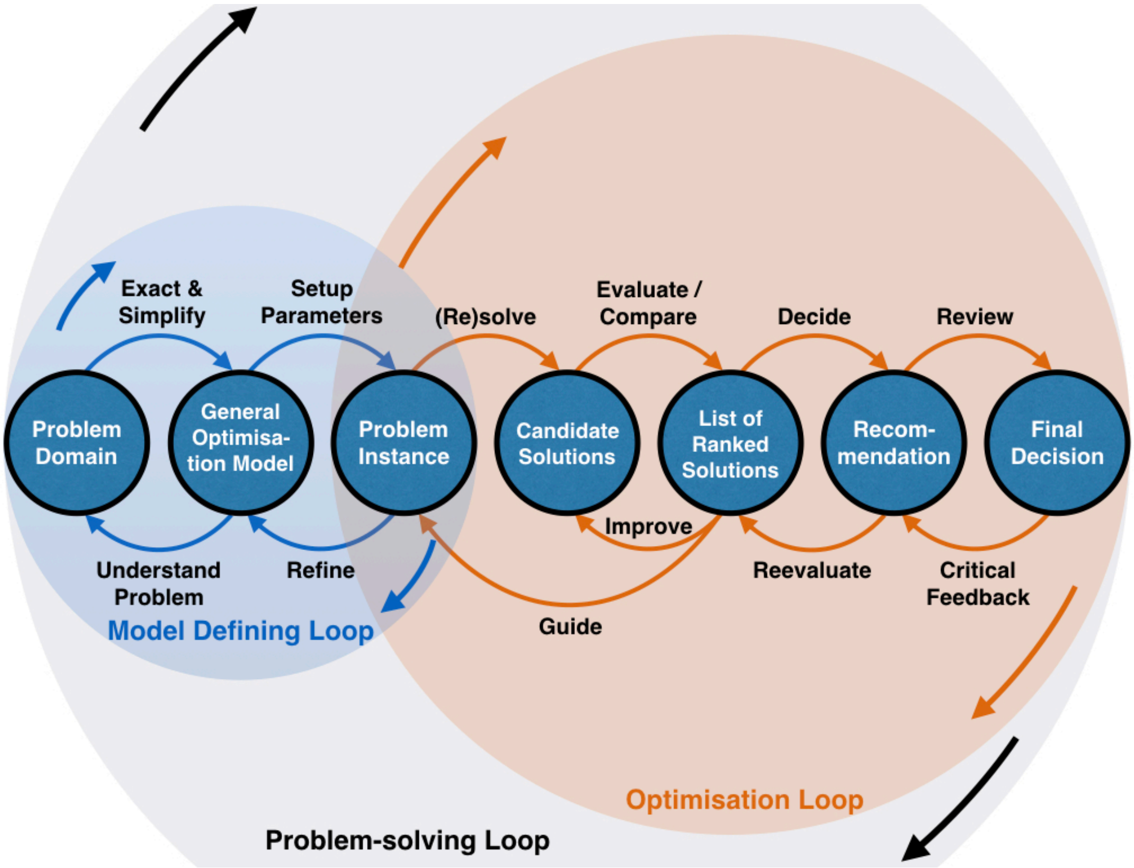

Figure 4.2: The problem-solving loop.

The *problem-solving loop* shown in Fig. 4.2 is our attempt to provide a similar theoretical framework for understanding the high-level user goals and processes in human-in-the-loop optimisation. There are two main loops: the *model-defining loop* captures the development of the generic mathematical optimisation model, typically by an expert in optimisation, while the second loop, the *optimisation loop* captures the use of the model by the end-user, typically the domain expert, to support their decision making. We will start from explaining the model-defining loop. First we will explain the three products represented in blue circles as follow:

- Problem Domain: the very original real-world problem in its domain before any simplification and extraction;
- General Optimisation Model: the mathematical optimisation model built based on the original problem with certain simplifications;
- Problem Instance: the detailed representation of the original problem using the mathematical optimisation model with constraints and objectives clearly defined.

There are four steps in the model-defining loop (two going forwards and two backwards):

- Extract & Simplify: Extracting the key components from the original real-world problem and modelling them into a general optimisation model. Other unimportant factors, as well as unnecessary details of the problem, are simplified and excluded from the optimisation model;
- Setup Parameters: Specifying and assigning the parameter values to form a detailed instance of the problem;

- Refine: adding new constraints and/or objectives into the optimisation model, or removing/changing existing constraints and/or objectives from the model;

- Understand Problem: Reevaluate the optimisation model to better understand the original real-world problem as well as identifying any overlooked components existed in the original problem but not included in the model.

An important observation of the model-defining loop is that at each step of the process the analyst may revisit earlier steps in light of new insights. For example, an optimisation expert develops a mathematical model based on the real-world problem. A solution is produced after the expert creates the problem instance by assigning values to parameters. However, a domain expert is not satisfied with the quality of the solution. As a result, certain constraints need to be adjusted and a new constraint has to be introduced to the model. The domain expert is not necessarily an optimisation expert. Therefore, it is impossible for the domain expert to refine the model and to include the new constraint in the traditional optimisation approach.

However, this is different in the human-in-the-loop approach for optimisation. With a well-constructed model, human-in-the-loop optimisation offers the flexibility to make changes to constraints and objectives. It is no longer difficult for the domain expert to make such model modifications. This also opens the possibilities for the domain expert to, later on, extend the model by including other constraints and to create other problem instances. This model refinement process can take multiple iterations. This is beneficial for the domain expert to build a better understanding of the optimisation problem. The mathematical model is also improved so that the gap between the model and the problem becomes closer.

Our focus in this thesis is on the optimisation loop. It has the following steps:

- Solving the generic model with the problem specific data to generate a pool of candidate solutions

- Evaluation and comparison of these candidates to give a ranked list of solutions

- Decide upon a recommendation which may be a single solution or solutions with an analysis of the tradeoffs between them

- Review by relevant stakeholders and decision makers to determine the final decision

Like the sense-making loop the problem-solving loop is interactive and the analyst will frequently return to earlier steps. For instance, after evaluating and comparing the solutions in order to rank the candidate solutions she may decide that none of them are good enough and try to generate better candidate solutions, either by refining the general model by guiding the solver, or by improving an existing solution manually by changing the values assigned to some of the decision variables. Thus, steering of the needle away from a grid point is an example of manual improvement.

Another important loop results from critical feedback from stakeholders and the final decision makers on the recommendations which may lead to reevaluation of the ranked list. human-in-the-loop optimisation is regularly employed to solve problems where the wrong decision can lead to significant loss or even death. As we have seen in the first study robust review is an important part of the problem solving process in such high-stake applications and will often lead to refinement or even rejection of the initial recommendation. This differs from many visual analytics applications in which findings of the analyst tend to be presented as a *fait accompli* rather than being the subject of robust collaborative review.

Examination of the optimisation loop and of the prototype tool developed for our case study clarifies the tasks for which interactive visualisation may be useful:

- *Evaluate:* Showing a single solution and its associated constraints and objective values in order to understand and evaluate it.
- *Compare:* Comparing multiple candidate solutions to rank them and to better understand the tradeoffs between them (this is especially important in multi-criteria optimisation).
- *Improve:* Manually manipulating a solution to improve it.
- *Guide*: Guiding the optimisation solver to search for new solutions in "interesting" regions of the search space of the problem instance.
- *Review:* Presenting recommended solutions including alternatives to decision makers and stakeholders to review.

There are a number of different kinds of multidimensional data that may need to be visualised. The first is a single solution and its "fitness". The second is the solution space, the set of allowed values for the decision variables. The third is the fitness landscape: the set of possible values for the objective criteria while a fourth is the search space being explored by the solver.

4.3 Study 2: Focal Brachytherapy

In our second study we investigated the use of human-in-the-loop optimisation for focal therapy. Our aims were fourfold:
1. Observe participants' exploration of focal therapy planning;
2. Determine the participants' preferred method for creating treatment plans for focal therapy (manual, fully-automatic optimisation or human-in-the-loop optimisation) and the reasons for their preference;
3. Investigate whether experience with a human-in-the-loop optimisation tool will increase participant trust in focal therapy and in optimisation-based treatment planning for focal therapy;
4. Obtain feedback on the design of the prototype tool.

4.3.1 Study Design

Apparatus & Materials: We used the human-in-the-loop optimisation prototype tool for this study after we rolled back the changes required to support whole gland rather than *focal therapy* and also integrated the improvements from the first study. We used the tool to prepare two automatically generated focal plans with different TCP values: **F2**: with TCP = 0.87 and **F4**: with TCP = 0.96.
Participants: Participants were recruited at a second presentation given to the brachytherapy team at the Alfred. This reported the findings from the first study, introduced focal therapy and explained the purpose of the second study. Participants from the first study were again asked to participate to ensure they were familiar with the tool and the project methodology and aims. One could not participate but two new clinical professionals were recruited and were provided with additional instruction in the tool (to compensate for their lack of experience of Study 1). This gave a total of 7 participants (one RO, three MPs and three RTs). As in the first study, because of the low number of participants we have also included the results from the pilot study with a member of our research team (an experienced RO).
Procedure: Our second study followed the same structure as the first:
1. *Introduction Activity:* We asked participants questions about their preferences in treatment planning methods (whole gland, focal) for prostate brachytherapy, and preference

and trust in methods (manual, fully-automatic, human-in-the-loop (or semi-automatic)) for creating a focal plan.

2. *Training Activity:* We presented our prototype tool and explained its operations. Participants were asked to briefly evaluate both TCD and TCP visualisations and provide comments.

3. *Planning Activity:* The main part of the study was to evaluate four focal treatment plans (one produced manually (**F1**), two produced fully automatically (**F2** and **F4**) and one produced using human-in-the-loop optimisation (**F3**)):

   (a) Participants were asked to manually create a focal plan **F1** with the tool and then score it. They were provided with the choice of two auto-seed-loading templates as the starting point, so as to reduce the time required for the study and to mirror the manual planning approach for the whole gland therapy approach they were familiar with.

   (b) Next, focal plan **F2** was presented. Participants were asked to evaluate and then score it.

   (c) Then participants were asked to improve focal plan **F2** using the tool and score the resulting focal plan **F3**.

   (d) The final focal plan **F4** was presented and evaluated and scored by participants.

They were then asked to rank the four focal plans.

4. *Recap Activity:* We repeated the questions about participants' preferences of both treatment methods and approaches to create a focal plan. Suggestions for improving the prototype tool were solicited.

4.3.2 Results & Discussion

One participant (P5 as short for Participant 5) did not wish to manually create or improve a focal plan because this participant thought it was outside their area of expertise. As a result, only focal plans **F2** and **F4** were included for this particular participant. We included the participant's other responses.

Aim 1 (Observation of participants' exploration of focal therapy planning) was based on the Planning Activity. To this end we asked participants to "think aloud" as they created a focal plan.

We observed they used different planning strategies. However all participants started with dose coverage of the prostate gland. Because of the unfamiliarity of focal therapy and TCP many participants were not sure about what a clinically acceptable focal plan should look like as they have never done any focal treatment before. They were unsure about the right TCP threshold and the interpretation of TCP contours.

One participant finished planning without using either TCP or TCD visualisations at all because of unfamiliarity. Another participant refined the focal plan by using the TCP contours "*as a second assessment*", whereas the other three radiotherapy team members fine-tuned plans based on the extra tumour information provided from TCD visualisations without looking at TCP contours at all. The remaining two participants adjusted their focal plans according to the TCD visualisation and finally double checked and refined plans by using TCP contours. Five of seven utilised the TCD visualisation to further adjust the seeds' and needles' positions to give a better coverage around the high tumour-risk areas. Most participants said the TCP and TCD visualisations were useful and that they did affect the way they do treatment planning, for example:

> *"Having contours showing your [tumour] control probability is really good from a planning point of view." P3*

and

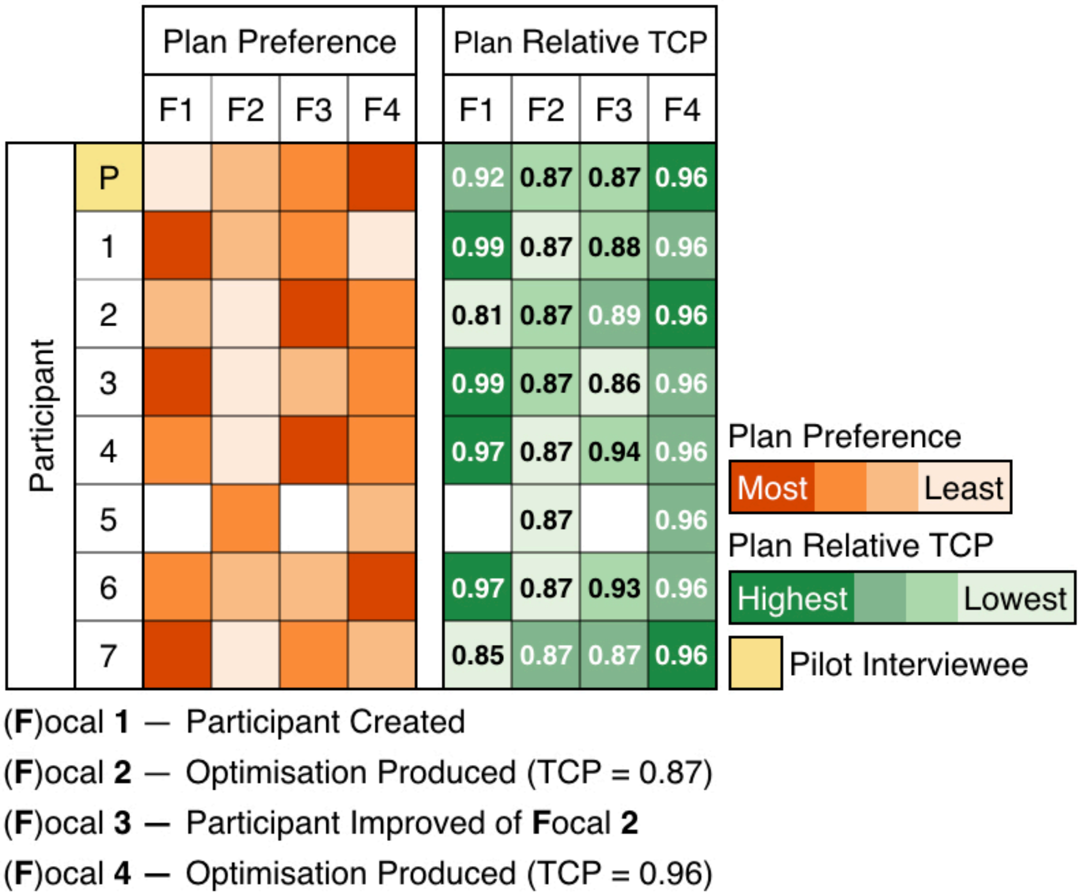

Figure 4.3: The relative ranks of all four focal plans and their associated TCP scores and relative TCP ranks. The ranks of focal plans (F1 & F3) are left blank for participant 5 (P5) because of the unwillingness to create and improve the focal plans.

> *"I liked it (TCD visualisation)." P1*

However, participants were curious and somewhat sceptical about the reliability of the data underlying the TCD calculation, for instance:

> *"I definitely think it (TCD visualisation) will be useful. If this information (TCD) is reliable, that will guide the treatment." P1*

Participant ranking of the focal plans (see Fig. 4.3) was difficult to interpret. Focal plans **F1**, **F3** and **F4** received roughly equal ranks which indicated that focal plans produced by manual, fully automatic optimisation (with high TCP setting) and human-in-the-loop optimisation are acceptable. However, the automatically produced plan with low TCP was not liked. It is also clear that overall TCP score is not the only quality that participants were using to evaluate the plans. Other factors, such as dose coverage, needle positions and patterns, were also important and considered in the evaluation. This provides support for human-in-the-loop optimisation rather than automatic optimisation. One participant commented:

> *"The optimisation does not take a lot of the clinical information you have and there is still some kind of knowledge that we have but the machine does not." P2*

To address Aim 2 (User preferences for creating treatment plans for focal therapy) we asked participants to rank the methods (manual, fully automatic optimisation, human-in-the-loop optimisation) in order of preference from most preferred (rank 1) to least preferred

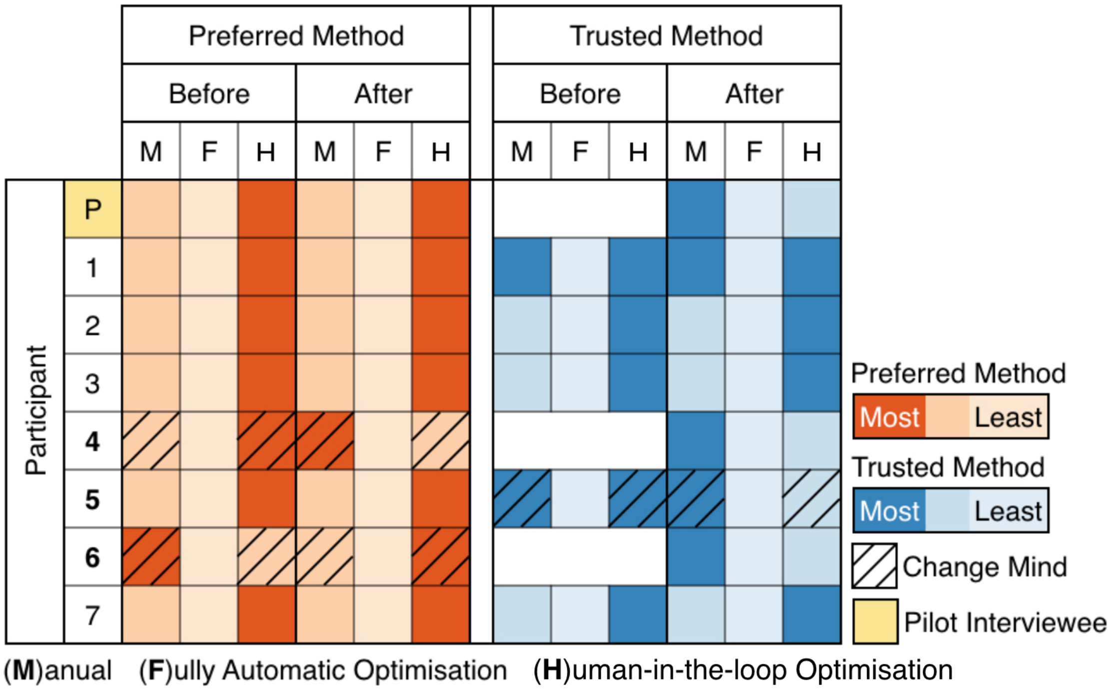

Figure 4.4: Participant preferences and level of trust in different methods for creating a focal treatment plan including **M**anual, **F**ully automatic optimisation, and **H**uman-in-the-loop optimisation. They were asked to rank the methods before and after experience with the tool.

(rank 3) for creation of a focal therapy treatment plan and in terms of trust. They were asked at the beginning and the end of the interview so that we could see if their experience with the tool changed their response. The results are shown in Fig. 4.4.

Most participants (6 of 8) preferred human-in-the-loop optimisation from the beginning to the end. Two changed their preferences. One changed from human-in-the-loop optimisation to manual because the participant was not happy with the focal plan produced from optimisation (poor dose coverage and needle loading pattern, lack of symmetry) and felt that fine-tuning an optimisation produced plan cannot completely fix these problems or would take too long to do so. The other one preferred manual at the beginning but human-in-the-loop optimisation in the end. After trying out the prototype tool they thought human-in-the-loop optimisation has more potential in treatment planning because of the speed advantage from optimisation and the flexibility to make adjustment with interactions. The responses strongly support that human-in-the-loop optimisation is the preferred method for focal therapy planning.

Aim 3 was to evaluate the effect of human-in-the-loop optimisation on trust in planning method and treatment method. In the case of optimisation-based treatment planning half of the participants trusted human-in-the-loop optimisation the most from the start to the end. There were 3 participants who did not trust any of the methods before trying and seeing each themselves. These cells were kept empty in the trust preference table (see Fig. 4.4). All 3 of them trusted manual planning the most at the end of the interview. Two of them were very sceptical about focal therapy during interviews. It is perhaps not surprising that they trusted the manual approach more in the end because they had a strong belief in whole gland therapy and manual treatment planning approach is the standard in whole gland therapy. One participant changed their mind from equally trusting both manual and human-in-the-loop optimisation to trusting manual more than human-in-the-loop optimisation. The main reason for this change was concern about

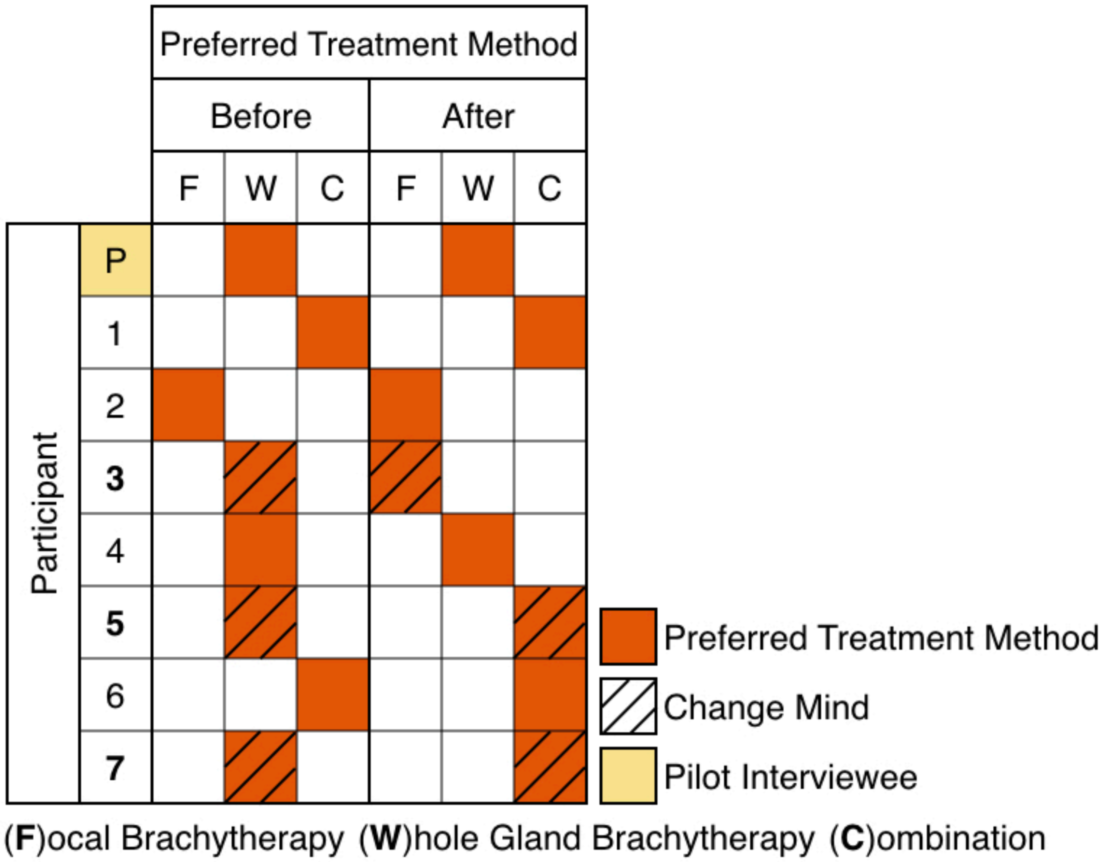

Figure 4.5: Participant preferences for treatment methods in prostate brachytherapy, before and after experience with the tool.

prostate dose coverage and the belief that the coverage from both optimisation produced focal plans was not satisfying while using the manual approach could give better coverage to the prostate gland.

Overall, by the end of the study half of the participants trusted the manual approach and the other half trusted the human-in-the-loop optimisation approach the most. What is clear is that the fully automatic optimisation approach is the least trusted, all 8 participants ranking it last. One stated:

> "*Optimisation is like a black-box and I do not trust any black-box.*" *P5*

Others also pointed out the importance of manual adjustment. We also heard many times that clinicians will very rarely accept a treatment plan from optimisation without any changes and more often than not they would like to make some manual adjustment to the plan.

In Study 1, experience with the tool for whole gland planning did not appear to increase or decrease participants' level of trust in human-in-the-loop optimisation. However, it is a different story for focal therapy (see Fig. 4.5). Participants were asked 'If both focal brachytherapy and whole gland brachytherapy were available which would you prefer to use? Why?' at the very start and the very end of the interview. At the start of the study, 5 of 8 participants preferred whole gland therapy, another two preferred the combination of whole gland and focal therapy and only one preferred focal therapy. By the end of the study, 3 of the 5 who preferred whole gland therapy changed their minds: one preferred focal therapy and the other preferred the combination. The participants who preferred either the combination or focal therapy did not change their answers. One participant who changed their preference from whole gland therapy into focal therapy stated after using the tool that providing extra information about prostate tumour cells from TCD visualisation was really useful and she had become used to seeing the TCP contours. This is a strong indication that experience with the human-in-the-loop planning tool and the planning exercises did increase participants' trust in focal therapy.

From our interviews we learned that many professionals at the moment are resistant to the adoption of focal therapy due to its current immaturity as a treatment modality. It might be the future for radiotherapy, but its widespread use will take time. Professionals still trust the whole gland therapy approach, with which they have extensive experience and versatile knowledge. A few professionals are rather open-minded. They are willing to accept new technologies. But they still prefer to have a human intervention to override the seed locations. The professionals rely upon specific patient details and have knowledge that machines do not in treatment planning. Therefore, fully-automatic approaches to produce focal treatment plans are currently the least preferred and trusted. In the short-term, human-in-the-loop approaches, such as those studied here, are therefore more likely to gain traction than fully automatic approaches.

Aim 4 (Feedback on the design of the prototype human-in-the-loop optimisation tool) was addressed in the Recap Activity and by observations of tool use in the Planning Activity.

There were few suggestions, suggesting they were now comfortable with the design of the tool and visualisations. Some participants suggested overlaying the TCP contours with the TCD visualisation, and the isodose contours with the TCD visualisation in order to better support the visual representation of a focal plan. We made the suggested changes and used them in the subsequent studies. When modifying plans they used *Add* (a seed), *Remove* (a seed) and *Steer* (a needle) but did not use *Lock & Re-optimise*. When queried about Lock & Re-optimise most responded it would be useful and some of them gave situations where it could be used such as in early stage of planning after loading seeds from templates to adjust the plan followed by manual fine-tuning using the other operations afterwards.

4.4 Conclusions

We conducted two studies to evaluate our prototype human-in-the-loop optimisation tool described in previous Chapter 3. It provided an in-depth analysis of the potential use of human-in-the-loop optimisation for treatment planning in both whole-gland and focal LDR prostate brachytherapy. These two studies, as well as the design and implantation of the prototype human-in-the-loop optimisation tool we used in the studies, add significantly to our body of knowledge about the potential use of human-in-the-loop optimisation in real-world applications and how to integrate interactive visualisation into such tools.

One big difference between our studies and other studies in human-in-the-loop optimisation is that we used domain experts to evaluate the tool. Specifically, we recruited seven radiotherapy team members from the Alfred Hospital in Melbourne, Australia in each study, which made it possible for us to clearly understand the typical workflow to create and approve a treatment plan in our first study. Enlightened by the workflow, we developed a new theoretical framework for understanding high-level user goals and tasks in human-in-the-loop optimisation called the problem-solving loop, analogous to the sense-making loop framework widely used in visual analytics mentioned in both Chapter 1 and 2.

Our second study suggested that users' trust is increased in focal therapy with the experience of the prototype human-in-the-loop optimisation tool. It also showed that radiation oncology professionals overwhelmingly prefer to use human-in-the-loop optimisation to do focal brachytherapy treatment planning.

In next Chapter 5, we will explain the design guidelines generalised from the problem-solving loop explained in this chapter, as well as introduce our second case study and present the results of a heuristic evaluation of the human-in-the-loop optimisation tool developed for the case study.

Chapter 5

Design Guidelines for Human-in-the-loop Optimisation

In the previous chapter, we introduced the *problem-solving loop*, a theoretical framework for human-in-the-loop optimisation, which is an analogue of the sense-making loop in visual analytics. It answers the research question **RQ 1-1: What are the high-level goals and processes in human-in-the-loop optimisation?**

In this Chapter 5, we address the other research question **RQ 1-2: What are effective visualisation and interaction techniques to support human-in-the-loop optimisation?** To make practical use of our theoretical framework—the *problem-solving loop*—we have developed general design guidelines for designing effective interfaces for human-in-the-loop optimisation that are informed by the various steps in the loop.

After the introduction of the design guidelines, we will discuss how these guidelines have been reflected in the interface design for our first use case—Prostate Brachytherapy. Then we will introduce our second use case—Interactive Vehicle Routing Problem (VRP)—after which we will give details about our interface design for a human-in-the-loop optimisation tool for solving this second use case. We will focus on explaining how we have used the design guidelines to guide the design process.

At the end of this chapter, we will provide a heuristic evaluation [86] of the interface for the second use case to verify the usability of the interface and the usefulness of the design guidelines.

5.1 Design Guidelines

We have developed the following design guidelines based on our *problem-solving loop* (see Fig. 5.1):

- ***Guideline 1*: Appropriate visual representations of solutions & constraints.** Representations of solutions and constraints should, where possible, make use of visualisations currently used by domain experts so that they can understand and make sense of them. Importantly, the visual representations of solutions and constraints should be tightly coupled. Specifically, when presenting a solution the constraints need to be represented as well to help users make sense of the solution and to understand the solution space immediately around it. It may look obvious to represent both solutions and constraints, but often this has been overlooked by optimisation researchers [87].

  This guideline is central to supporting the *problem-solving loop*. Without appropriate visual representations of solutions, *Evaluating/ Comparing* solutions as well as

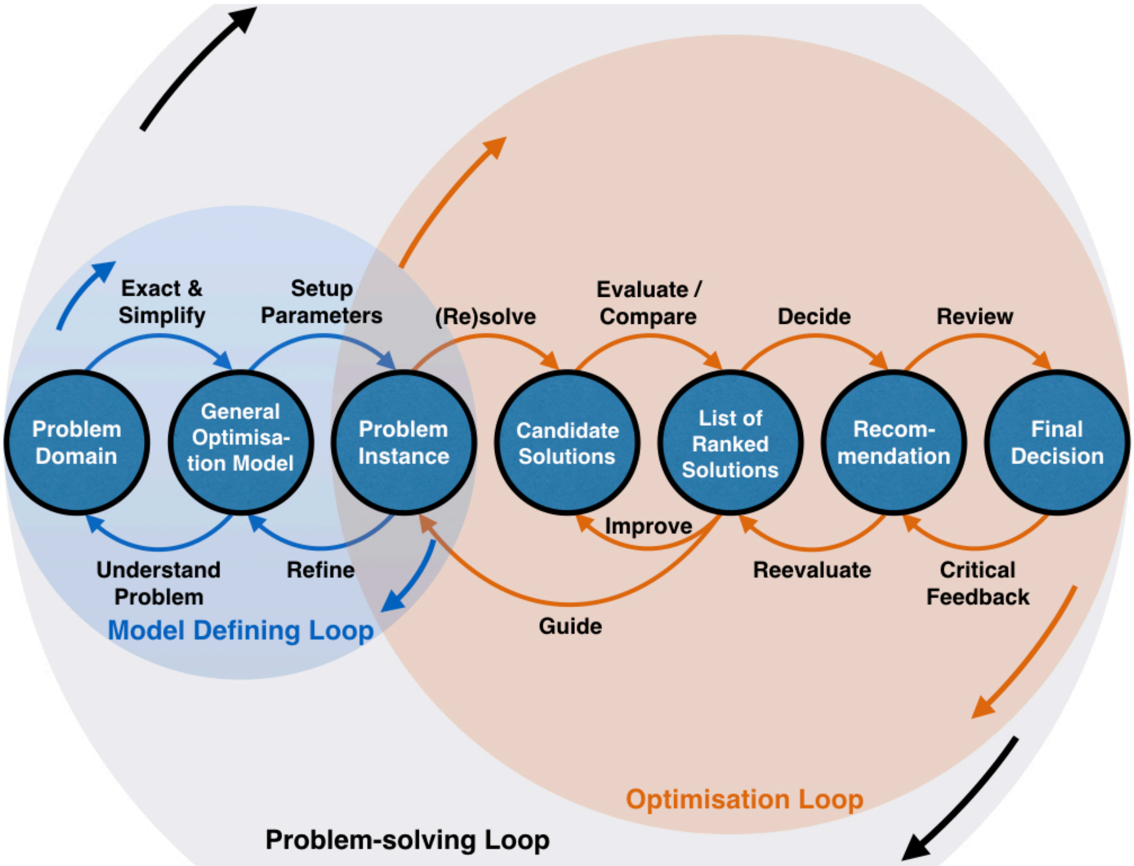

Figure 5.1: The problem-solving loop.

Guiding optimisation solvers to find new solutions will be very difficult. Without appropriate visual representations of constraints, *Improving* a solution becomes more like a "blind" solution manipulation and *Refining* an optimisation model is difficult to achieve.

- ***Guideline 2*: Modifiable optimisation models.**
  The interactive visualisation should be designed flexibly enough that users can not only modify existing constraints and objectives but also specify new constraints and add them to the optimisation model.

This guideline is to support the *Setup Parameters* and *Refine* steps in the *model-defining loop*. Recall from Chapter 2, sometimes constraints are missing or the optimisation problem can change dynamically. Therefore, it is critical for the user to be able to adjust the optimisation model. Specifically, modifiable optimisation models make it possible to refine values of parameters to update the existing problem instance or even create a new one. Importantly, being able to refine optimisation models can reduce the gap between optimisation models and real-world problems and produce more realistic and satisfying solutions.

However, adding constraints on the fly could be limited for two reasons. First, the new constraints and the type of optimisation solver need to match. For instance, the user cannot add non-linear constraints to a problem solved by a linear solver without changing the solver. Second, the capacity of the user to add constraints by him/herself may be limited, especially when the user does not have a computer science background. Perhaps, constraints acquisition would be a useful approach to elicit constraints from the user [88–90].

- ***Guideline 3*: Direct manipulation of solutions.**
  Users should be able to directly manipulate a solution and make whatever changes they like to the variable values.

  This guideline allows users to *Improve* an existing solution. Direct manipulation is a natural approach to change a solution. Often, users are sceptical about solutions even if they look good. Being able to manipulate and change a solution allows users to test the solution and verify its quality in their own way. With the support of appropriate representations of solutions and constraints, the user can explore the trade-offs between solutions, which is helpful for solution evaluation as well as perhaps finding a more satisfying solution.

- ***Guideline 4*: Gallery of solutions.**
  Multiple solutions should be collected and represented in the form of a gallery. This should provide the navigation interface for exploring the solutions. The gallery interface should also allow users to manually and automatically (re)order solutions based on their preferences and to annotate solutions.

  The solution gallery supports *Comparison* of multiple solutions, allowing them to create a list of ranked solutions based on the preferences. Manual re-ordering allows flexibility while automatic re-ordering based on say the value of the objective function speeds comparison and ranking. Moreover, with the ability to add annotations of solutions, users can make the ranking list more readable and informative for themselves or other users. The solution gallery also gives the user an overview of all solutions, which can be useful for a quick comparison of solutions or to navigate between solutions with the gallery image and annotations providing a memory aid.

- ***Guideline 5*: User controlled re-optimise.**
  We believe it is necessary to provide users with the ability to re-optimise part of a solution based on their preferences. Often, users are satisfied with part of a solution but not all. It is a good idea to provide the ability for the user to "lock" the values of variables in the good part of the solution and improve the remaining parts of the solution using the re-optimise option.

  In such a way, users can *Guide* the optimisation solver to explore the search space that they are interested in. Specifically, optimisation solvers should make minimum or no changes to the parts of a solution that are locked and apply any modifications to the unlocked parts without breaking any constraints. This has the benefit of narrowing down the current search space because of the locked parts of the solution and so speeding up the search process so that optimisation solvers can find a better solution in a shorter time. It also provides *stability* in the solving process which helps to preserve the user's mental model [91, 92].

- ***Guideline 6*: Side-by-side comparisons of two (or perhaps more) solutions.**
  To support better *Comparison* of solutions the interface should allow two (or more) solutions to be viewed side-by-side. One important aspect is to allow users to verify details from either solution. Another is to identify the similarities and differences

between the solutions. Visual cues may be used to highlight these.

Using side-by-side comparisons, users can more readily rank solutions and select the best one. This is necessary because solutions with equally-good values of the objective function may differ significantly. Seeing and comparing the trade-offs between these solutions is necessary in order to find the best solution. As a result, users can *Compare* solutions in a more meaningful and powerful way.

- ***Guideline 7*: Generating different solutions.**
  We believe an ability to automatically generate a new solution that is significantly different to any of the solutions in the gallery, but which is still of high qualities with respect to the objective function will support the user finding better solutions.

  The purpose of finding different solutions is to better *Solve* the optimisation problem. This has the benefit to enrich the solution gallery so that the solutions have diverse characteristics. Seeing different solutions can help users better understand the problem by knowing possible trade-offs between different criteria.

- ***Guideline 8:* Feedback on solving process.**
  Often, optimisation software is silent about the solving process. However, finding a solution can take minutes, even hours or days. Keeping users waiting and staring at the screen with nothing happening is confusing and disquieting. Even something as simple as a progress bar can show users that the optimisation software is still properly functioning, perhaps increasing their trust in the optimisation software.

- ***Guideline 9*: History of solution manipulations.**
  As discussed in *Guideline 3 & 7*, it is important to allow users to directly manipulate solutions and generate new solutions. During this process it is likely that the users will wish to return to previous solutions. Providing users with the history of solution manipulations facilitates this.

  The history of solution manipulations presents the provenance of the solution because all manipulations are recorded and presented to users [93]. It also encourages the user to *Manipulate* solutions as there is no need to be afraid of making a mistake—they can always go back to a previous solution and start over again.

The above nine guidelines were informed by our *problem-solving loop* framework. Certainly, other HCI and user interface guidelines should also be used when developing interfaces for human-in-the-loop optimisation such as Nielsen's ten usability heuristics for user interface design [94] including preventing errors to maintain good functionality of the system and ensuring fast system response time to users.

Guideline 6 is closely related to the ***Guideline 4***. When applying the guidelines, it is worthwhile to consider how to combine them during the interface design. However, both guidelines focus on different aspects. ***Guideline 4*** recommends having a solution gallery to allow users to select multiple solutions for solution comparison, evaluation or improvement. ***Guideline 6*** focusses on how and why to do a side-by-side comparison with enough details provided from multiple solutions.

Guideline 7 suggests to generate different solutions. As mentioned in Chapter 2, randomness is one approach to make the solution pool diverse. Another approach is to define a minimum difference threshold as part of the objective for the new solution and add a model specific function (solution diversity matrix) measuring the difference between the two solutions [95–97].

5.2 Reflections of problem-solving loop in Prostate Brachytherapy Use Case

The prototype tool for human-in-the-loop prostate brachytherapy treatment planning introduced in Chapter 3 was developed before the nine design guidelines described above. An interesting question to ask is:

> How well does the treatment planning tool follow the design guidelines?

We address this particular question here. Another question is:

> How can we use the guidelines to inform the development of interfaces for other human-in-the-loop optimisation tools?

We will address this question later on in Section 5.4.

We now examine the prototype tool following the order of the guidelines as below:

- ***Guideline 1*: Appropriate visual presentations of solutions & constraints.** This guideline is well supported. In the prototype interactive optimisation tool for focal prostate brachytherapy, treatment plans are represented as a number of slices with seed positions, organ contours, radiation ranges and strengths all clearly displayed on each slice. As requested by the oncology professionals our treatment plans look very similar to plans produced by the standard commercial treatment planning system *VariSeed*. Also, the background of each slice is made black with grids drawn on top of it so that it looks very similar to a slice of an ultrasound scan. The radiation strengths are indicated by displaying contours of different dose levels using the exact same colour scheme based on the feedback from radiation oncology professionals. This clearly reflects *Guideline 1*. Using the interface to present a solution and constraints is shown in Fig. 5.2 Getting the solution representation right is the fundamental step to make human-in-the-loop optimisation work effectively.

- ***Guideline 2*: Modifiable optimisation models.** This is partially supported. In our prototype tool, we allow users to define the target TCP as well as three different urethra dose volumes (see Fig. 5.3). Also, users can choose from the two initial treatment planning protocols to decide the initial seed placements. This flexibility to modify objectives, dose constraints as well as seed placement constraints allows users to very easily produce treatment plans with different qualities and constraints settings. However, users cannot add new constraints to the optimisation model.

- ***Guideline 3*: Direct manipulation of solutions.** The guideline is well supported. One important feature of our prototype tool was the ability to modify seeds of a treatment plan. In clinical treatment planning professionals would like to change seed positions after the automatic seed placements to adjust radiation dosage around different organs. We allow professionals to add and remove seeds (see Fig. 5.4 & Fig. 5.5). We also support moving an existing seed off a grid point. This is necessary for the situation when the grid point is outside of the prostate gland, but still within the treatment planning margin. This is because it is safer to implant seeds into the prostate gland so that seeds stably stay at the insertion location as seeds placed outside the prostate gland can move away because of the softness of the outside tissues.

- ***Guideline 4*: Gallery of solutions.** This guideline is only partially supported. We designed a solution gallery to store

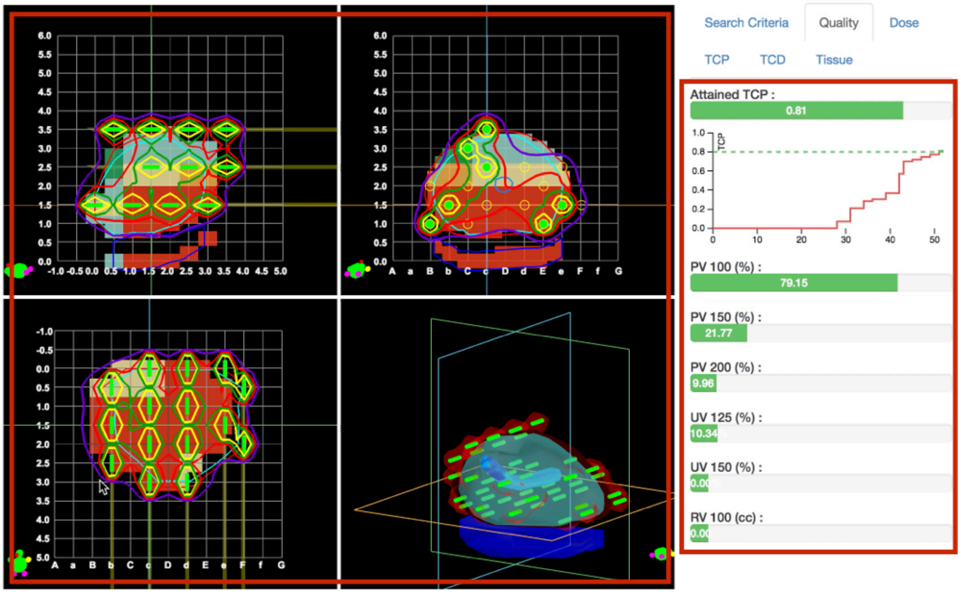

Figure 5.2: A treatment plan is presented using the interface. The four viewports handle the presentation of the plan, and the panel on the right shows the constraints such as TCP and UV 125 (%) (recall from Chapter 3). The reflection of the guidelines is indicated by a red rectangular outline.

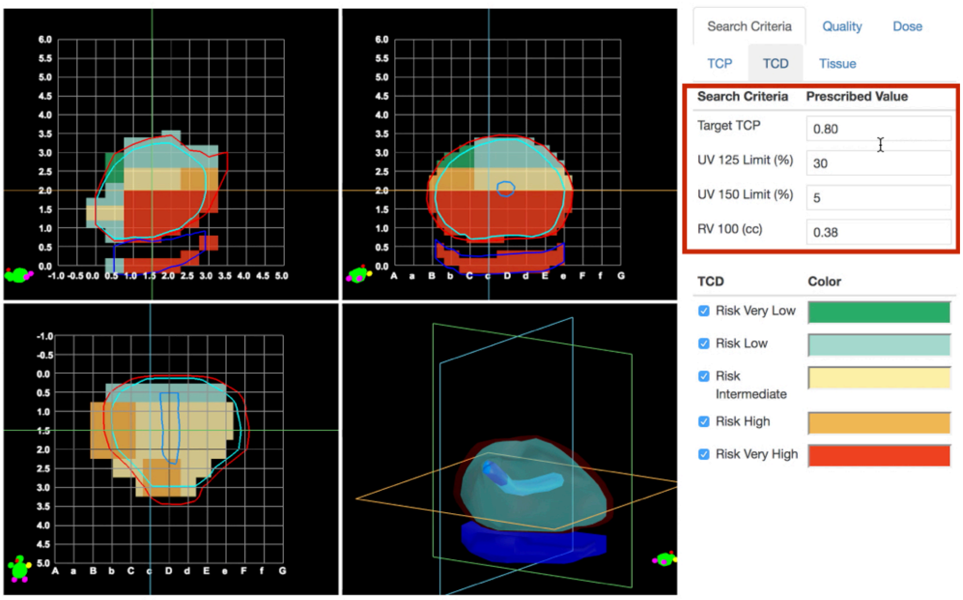

Figure 5.3: The panel on the right allows users to define the objective TCP and three dose constraints: UV 125 limit, UV 150 limit and RV 100. Check Chapter 3 for a detailed explanation.

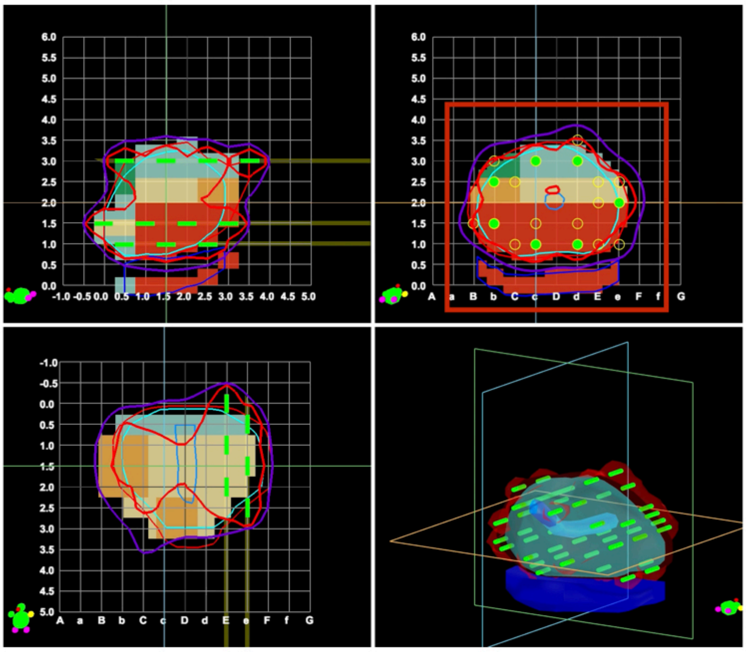

Figure 5.4: The plan before the user's manipulation.

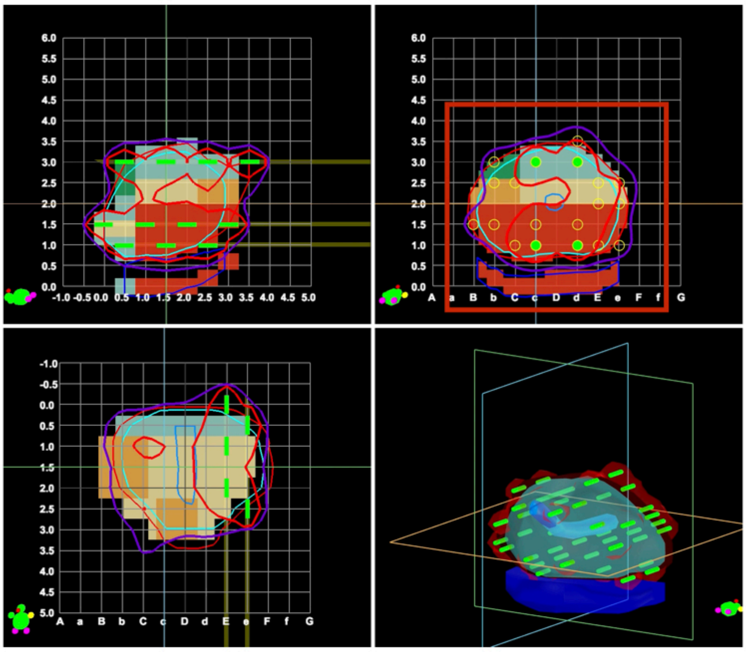

Figure 5.5: The plan after the user's manipulation. Two seeds on the left and one seed on the right are removed by the user.

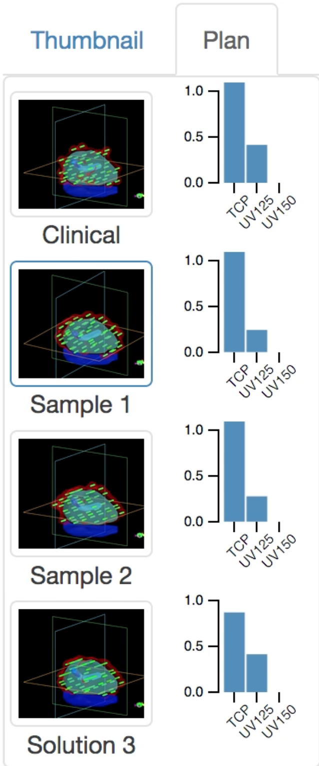

Figure 5.6: The solution gallery with four plans stored. The second plan, Sample 1, is current selected by the user.

and display all treatment plans (see Fig. 5.6). Users can access any plan at any time by selecting the plan from the gallery. They can rename a plan. However, manual or automatic re-ordering of treatment plans is not supported in our prototype tool, which we regretted afterwards. Therefore, *Guideline 4* is partially reflected in the interface.

- ***Guideline 5*: User controlled re-optimise.**
  The guideline is well supported by the lock and re-optimise operation. The operation takes two parts: First, the lock operation provides users with the ability to fix the locations of needles that they are satisfied with (see Fig. 5.7). When the location of a needle is fixed, every seed placed on the needle is also fixed so that the optimisation solver cannot change any of them when re-optimising the plan. Second, the re-optimise operation allows optimisation solvers the improve the current plan by reallocating the locations of all unlocked seeds to achieve the target TCP value without breaking any constraints (see Fig. 5.8). The lock & re-optimise operation allows users to guide the optimisation solver to specifically search the "interesting" area to improve the current solution close to what they want. This is exactly the idea behind the *Guideline 5*.

- ***Guideline 6*: Side-by-side comparisons of two (or perhaps more) solutions.**
  This is also well supported. In the compare mode of our prototype tool, we updated the solution gallery to support displaying two plans slice by slice with each pair of slices placed side-by-side (see Fig. 5.9). We calculated the dose differences from two plans and drew dose difference contours in all three sectional views to show where and how each plan differs from the other. We also displayed seeds from both plans

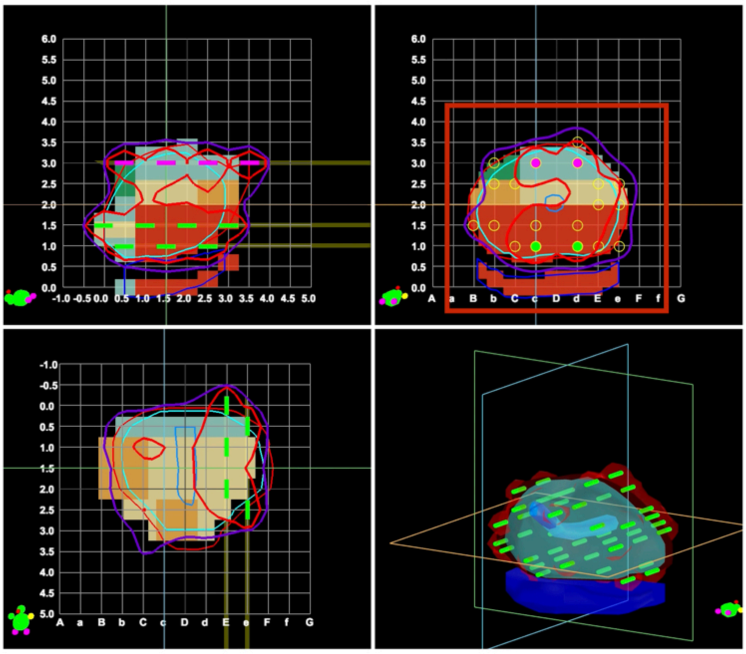

Figure 5.7: The plan before re-optimisation. The user has locked the top two seeds. They are marked in pink.

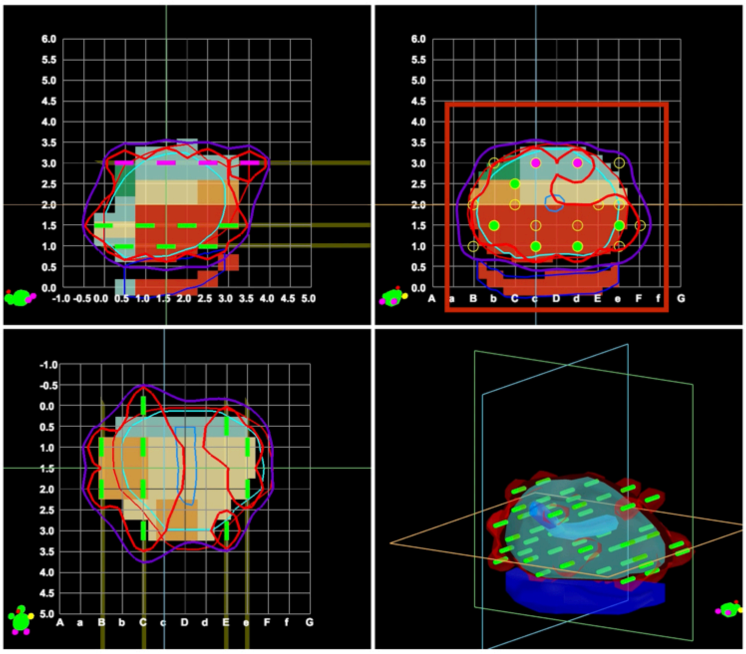

Figure 5.8: The plan after re-optimisation. The user locked two seeds remains unchanged.

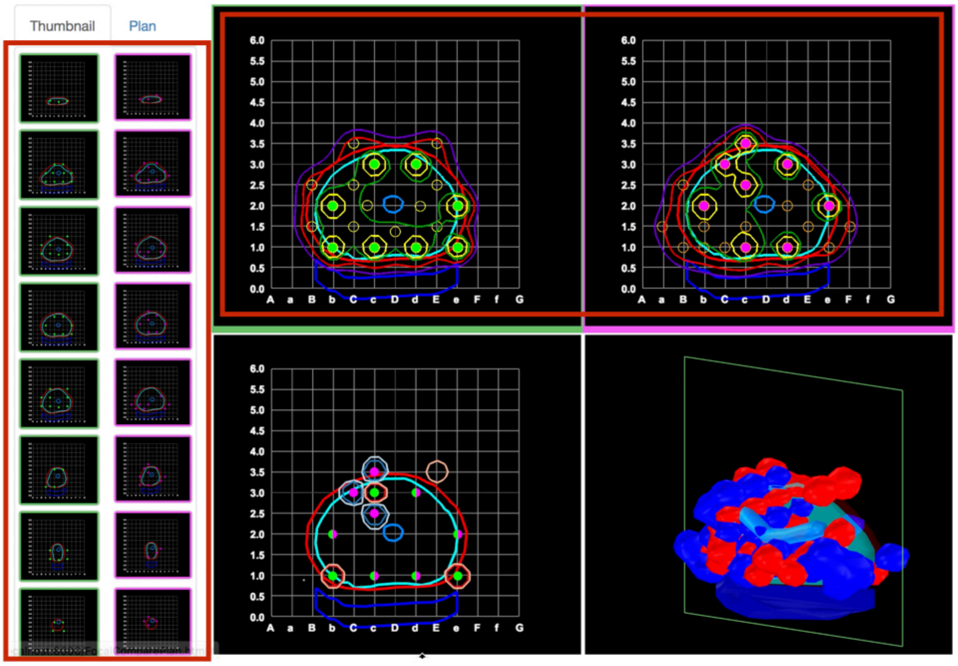

Figure 5.9: Two plans are presented side-by-side. The thumbnails of both plans are also presented side-by-side.

to help users better understand the dose difference contours. The compare mode design follows the *Guideline 6* very well.

- ***Guideline 7*: Generating different solutions.**
  This is poorly supported. This guideline is only partially reflected in our prototype tool design. Users can run the exact same settings for the TCP objective, dose constraints and seed placement constraints to get another treatment plan. This is typically different from previously produced plans because of the randomness of the underlying algorithm. However, the degree of differences between the two plans produced using the same setups is not controlled and they may look very similar or completely different. Thus *Guideline 7* is not completely reflected.

- ***Guideline 8*: Feedback on solving process.**
  The guideline is well supported. There is an objective line chart placed on the right-side panel of the interface to show the search progress (see Fig.5.10). When users start the search operation after setting up target TCP value and other dose constraints, each sectional view is updated based on new seed placements returned from the optimisation solver each time. The objective line chart is also updated to show how the TCP value has changed during the search process. As discussed in the *Guideline 8*, it is important to give users such feedback so that they are aware of the current system status and so are more confident about the system's functionality and they are more likely to believe the system is doing its work properly.

- ***Guideline 9*: History of solution manipulations.**
  This is not supported. Unfortunately, we do not have any measurement to track the solution manipulation history suggested by the *Guideline 9*. In hindsight, this was a mistake.

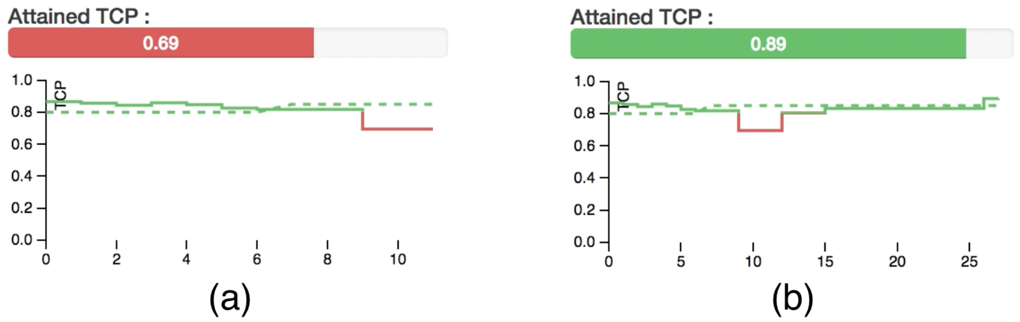

Figure 5.10: The search progress: (a) in the early period; (b) at the final stage.

5.3 Vehicle Routing Problem with Time Windows (VRPTW)

In Chapter 3 and 4, we have discussed our first use case prostate brachytherapy in great detail. Our human-in-the-loop prototype tool for supporting prostate treatment planning has been evaluated by radiation oncology professionals. Because of the specialised expertise required for this use case, the number of participants joining the study was limited. Therefore, we decided to investigate another use case, this time one in which less specialised expertise was required, so that we can generalise our findings to other applications and build a more complete understanding of the human-in-the-loop optimisation.

5.3.1 Use Case Selection

When choosing the problem domain for the second use case, we decided to use a more traditional optimisation problem so that techniques for its solution would be well studied. Also, we wanted a problem that could be understood by the general public.

We chose the Vehicle Routing Problem (VRP). This was first introduced by George Dantzig and John Ramser [98] in 1959 as the truck dispatching problem. The context of VRP is to pick up and deliver goods from a central location called the depot to customers who have placed orders for the goods. The objective of VRP is to minimise the total cost of all routes for a number of vehicles to deliver to all customers.

However, VRP problems are a little too easy to solve. We therefore chose a harder variant of the problem in which there are time windows in which each order must be delivered to a customer.

5.3.2 Problem Description

Here we formally introduce the vehicle routing problem with time windows (VRPTW) with details. Before we look into the terminology, we give a general problem example:

> Assume that you are the manager of an electrical repair service company in Melbourne. Your job is to assign a number of electricians from your company to deliver lighting fixing services to customers around Melbourne. Each customer has a fixed time window at home in which electricians can come and deliver the service. Late services are not allowed. If an electrician arrives earlier and there is no one at home, the electrician has to wait until the customer backs home so that the service can be delivered. As a service manager, you are aiming to minimise the total distance travelled by all electricians when you are planning how to assign customers to electricians.

We will use the following terminology:

- *Service*: the work of fixing the lighting problem.
- *Customer*: the person who has a lighting problem at home and requests a service to fix the problem.
- *Electrician*: the person who drives a vehicle to the customers' home to deliver the lighting fixing service.
- *Depot*: the central location of the electricians' warehouse.
- *Service Period*: the actual total amount of time (in minutes) required for an electrician to fix the lighting problem of a customer.
- *Time Window*: the allowed period of time (in minutes) to deliver a service to a customer.
- *Violation*: the situation in which the service delivered is later than the allowed period of time.

To simplify the complexity of the problem and maintain the right amount of difficulty, we have made the following **assumptions**:

- *Consistent Service Period*: the amount of time to fix a lighting problem is the same for all electricians.
- *Consistent Vehicle Travel Speed*: the speed of any electrician's vehicle is the same for all roads (with road conditions disregarded).
- *Fixed Single Depot*: There is only a single depot, and its location is fixed.
- *Fixed Customer Location*: All locations of customers are known and fixed in the problem instance.
- *Realistic Time Window*: The allowed period of time from a customer must be longer than the actual time required to deliver a service.
- *Shortest Route between Customers*: The route between two customers is pre-computed. This is guaranteed to be the shortest route.

There are a number of **constraints** in the problem. They are defined as following:

- *Route Loop*: Each electrician must depart from the depot and return there when all services are finished.
- *Single Customer Visit*: Each customer can only be serviced by a single electrician at any given time.
- *No Early Service*: No service can start sooner than the earliest time in the time window.
- *No Late Service*: No service can finish later than the latest time in the time window (No violation).
- *Complete Service*: Each electrician must finish all services for their assigned customers.

The single **objective** of the problem is to minimise:

- *Total Travel Distance*: The sum of the distances travelled by all electricians.

Figure 5.11: A screenshot of the overall interface.

5.4 Interface Design of VRPTW

We spent considerable time developing a browser-based visual interface for human-in-the-loop optimisation tool to solve the VRPTW problem. The guidelines proposed at the beginning of this chapter informed the interface design. This section, in particular, therefore provides a case study which answers the question we raised at the start of Section 5.2:

> How can we use the guidelines to inform the development of interfaces for other human-in-the-loop optimisation tools?

5.4.1 Front-end Interface Design

The interface is shown in Fig. 5.11. We first identify its main viewports:

- The view of a solution & objectives and constraints;
- The gallery & comparison;
- The history;
- The statistics & solver progress.

The view of a solution & objectives and constraints.
The first viewport is used to display a solution. A node-link diagram was our first thought, within which customers can be represented as nodes and the route from one customer to another can be represented as a straight line linking the two nodes. As a result, a single

route with multiple customers might look like something in Fig. 5.12 (a). If we consider multiple routes with multiple customers and also include the depot represented as a special node, the routes might look like this in Fig. 5.12 (b). However, there is one main drawback with this representation. A route between two customers is presented as a straight line. Realistically, it is very unlikely that there is a straight road to travel from a customer to another. Consequently, when we calculate the route distance between the two customers, using the Euclidean distance can be very misleading even when the geographical locations of the two customers are accurate. Such a node-link presentation of routes violate our ***Guideline 1*. Appropriate visual representation of solution & constraints**.

Therefore, we need to find a better representation. Why not just draw routes on a map? Using OpenStreetMap (OSM) is our second thought. OSM provides users with a free editable map of the world. The problem of the unrealistic distance of a route is completely eliminated. Not only we can get the accurate distance of a route between two customers, but also the street map itself provides a clear visual context to the problem. The solution to a VRPTW can now be appropriately presented (see Fig. 5.12 (b)). The new presentation conforms well to our ***Guideline 1*** now. From a technical aspect, we choose *Leaflet*, a leading open-source JavaScript library for interactive maps, to display routes on a map.

It is possible to have some routes overlapping between different trucks. It becomes confusing to read routes. A good way to reduce the confusion is to add an offset of each overlapped route. But we found this fiddly and impractical to implement fully within the time constraints of this experiment. Instead, when the user hovers the mouse over a particular truck route, all the routes are brought to the very front to override any other overlapped same routes. This is not the best way of doing so. But it sufficiently reduces the confusion.

It is also possible that a route needs to be re-routed for many reasons, such as heavy traffic congestion, vehicle accidents or even users' preferences. We have designed waypoints to enable the re-route operation. A single-click on a route will attach a new waypoint to it, after which users can drag and drop the waypoint so that the existing route will automatically adjust based on the current location of the dragged waypoint. A double-click on a waypoint will detach and remove it from a route. This provides users the ability to change a solution, which is all about the ***Guideline 3*: Direct manipulation of solutions.**

The automation of the re-routing relies on the Open Source Routing Machine (OSRM) server built by researchers in our institution. Each time when a route is re-routed, the back-end of our interface sends a query to the server asking for a route update. Then the front-end redraw the route based the update returned from the OSRM server. Sometimes it is necessary to add new customers or move existing customers to new locations. Therefore, by dragging and dropping a customer, the customer is re-located. Consequently, all routes associated with this customer are updated.

This is exactly the same interaction as re-routing by dragging a waypoint. Adding a new customer is done by a single click on the map. A double-click on an existing customer will remove it from the map. As a result, all routes attached to the removed customer will also be removed from the map. The operations of adding, removing or moving customers makes the optimisation model flexible and modifiable, which is the point behind the ***Guideline 2*. Modifiable optimisation models**.

After designing what a solution looks like, we consider how to show the time window constraints. Because the problem includes time, it is natural to consider representations related to scheduling such as a timetable. However, a traditional timetable is a grid with tasks and time stamps placed in each cell. This can hardly be regarded as a visual representation. Also, it is difficult to present the relationship between different time

Figure 5.12: (a) A solution of a VRPTW. Each location is represented as a circle and routes are represented as straight-lines connecting locations. The start location is marked in white whereas other locations are coloured in orange. (b) The same solution is represented using OSM with realistic travel routes drawn on the map. A home icon is used to represent the depot.

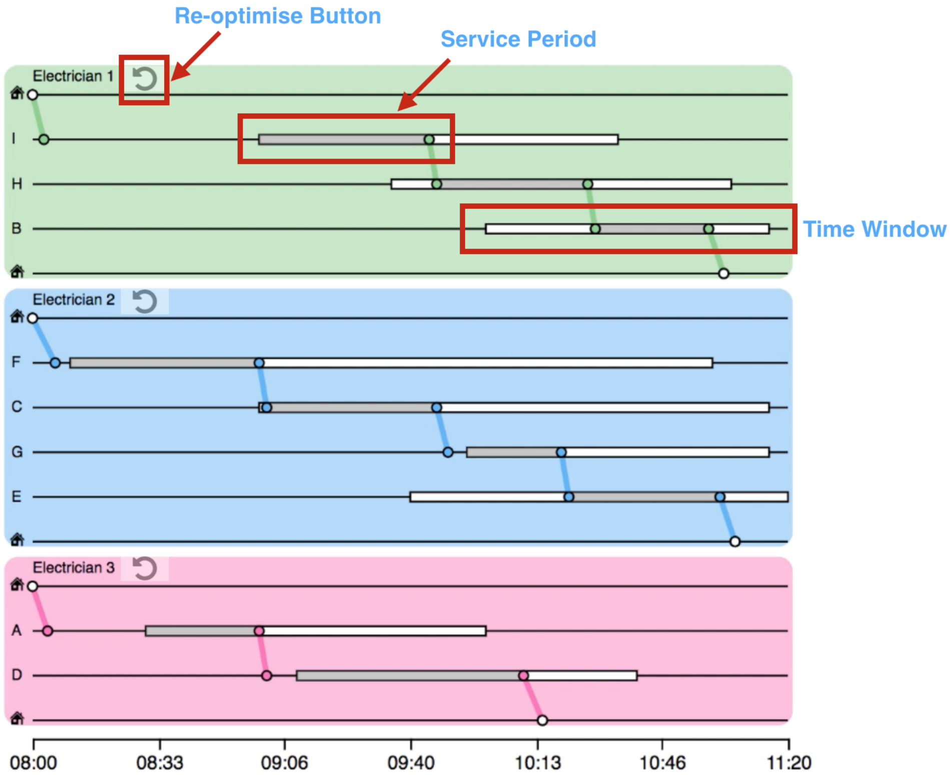

Figure 5.13: The representation of the overall timeline using a Marey diagram. The time window is represented as a white rectangle with a black border. The service period is drawn as a grey rectangle overlaid on top of the time window.

windows using a timetable. Gantt charts might be a good choice. A perhaps even better choice is Marey diagrams (or Marey charts) [99] delicately designed to visualise train movements in respect to times. It provides a visual presentation of train timetables, which fits well with our problem as electricians use vehicles to travel from customers to customers. Time windows from customers can be seen as a special kind of timetables for electricians. Therefore we choose Marey diagrams to represent time windows, which is appropriate for constraints representations in this case (see Fig. 5.13). This representation concurs with ***Guideline 1***. Specifically, each time window is represented in a rectangular shape with the length of the rectangle indicating the range of the time window.

Another important constraint is imposed by the service period from electricians. This is the amount of time required by an electrician to deliver a service to a customer. Therefore, an electrician's service period has to stay within the range of a customer's time window constraints. We overlay another rectangle on top of a time window rectangle to represent the service period constraint (see Fig. 5.13). However, it is possible that a service period exceeds the range of a time window, which means the service cannot be delivered on time to a customer, perhaps because the electrician must arrive after the start of the time window because of a previous job. We call this a time window constraints violation (constraints violation for short). We inform users by highlighting the time window rectangle in red when such a violation happens (see Fig. 5.14). This is also a reflection of ***Guideline 1***.

In order to distinguish the customers assigned to different electricians. All customers assigned to the same electrician are grouped together and represented using the same colour. Electricians are represented using big rectangles containing all assigned customers (see Fig. 5.13). We allow users to re-assign customers to electricians by drag-and-drop. When a customer is re-assigned to an electrician, the timelines of both affected electricians

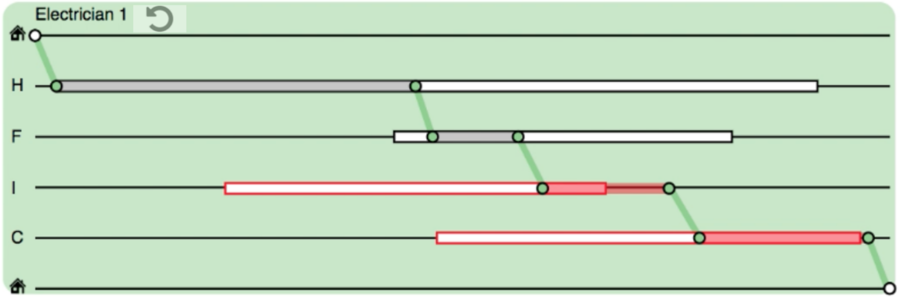

Figure 5.14: The time window constraints of the customer I and G (shown on the left) are violated. The violations are highlighted in red.

will be updated. Consequently, the routes that both electricians travel are also updated. Figure 5.15 shows an example of the re-assignment of a customer. This customer re-assignment is another approach to make changes to a solution. Therefore, it conforms to the ***Guideline 3***.

In order to distinguish the customers assigned to different electricians, customers assigned to the same electrician are grouped together and represented using the same colour. Electricians are represented as a big rectangle containing all assigned customers (see Fig. 5.13). We allow users to re-assign customers to electricians by drag-and-drop. When a customer is re-assigned to an electrician, the time lines of both affected electricians will be updated. Consequently, the routes that both electricians travel are also updated. Figure 5.15 shows an example of the re-assignment of a customer. This customer re-assignment is another approach to make changes to a solution. Therefore, it supports ***Guideline 3***.

It is possible for the user to add or remove customers and to adjust both time window constraints and service period constraints. This provides considerable flexibility in tailoring the model to deal with unforeseen real-world events. For instance, if an electrician wishes to visit a particular cafe at lunch we can model this using a new customer "lunch" at the cafe's location with time window and service period set to the lunch break. We might need to adjust a customer's time window if the customer has to leave home early. Or to adjust an electrician's service period to model dynamic events. For example, if an electrician's vehicle is broken down on the road, we can model this by adding a new customer at the break-down location and adjusting the electrician's service period to the time required to fix the vehicle. This prevents the electrician from delivering services to any assigned customers until the vehicle is repaired. We add handles to both sides of a time window and service period to enable such adjustments. The flexibility of adjusting constraints like this is designed to follow ***Guideline 2***.

Even though the users can only modify existing constraints in the interface, constraints modification is still an important aspect of making the optimisation model modifiable. It is true that we do not provide the operation to add new constraints. However, we have carefully considered it. For example, we can support the users to lock a specific customer to a specific truck. Maybe this is a loyal customer and the customer trusts the trucker driver more than the others. Another possible constraint is the relative order between customers, say customer A must be served before customer B for some reason. Supporting the users to add such constraints is not hard. However, it reminds challenging to design interactions as well as visual supports to make these constraints understandable and meaningful to the users. This is the main reason that we do not support the users to add constraints.

In order to support user interactions with the underlying optimisation algorithm, we have implemented the re-optimise operation following the ***Guideline 5.* User controlled re-optimise**. Specifically, we attach a re-optimise button to each electrician to enable a

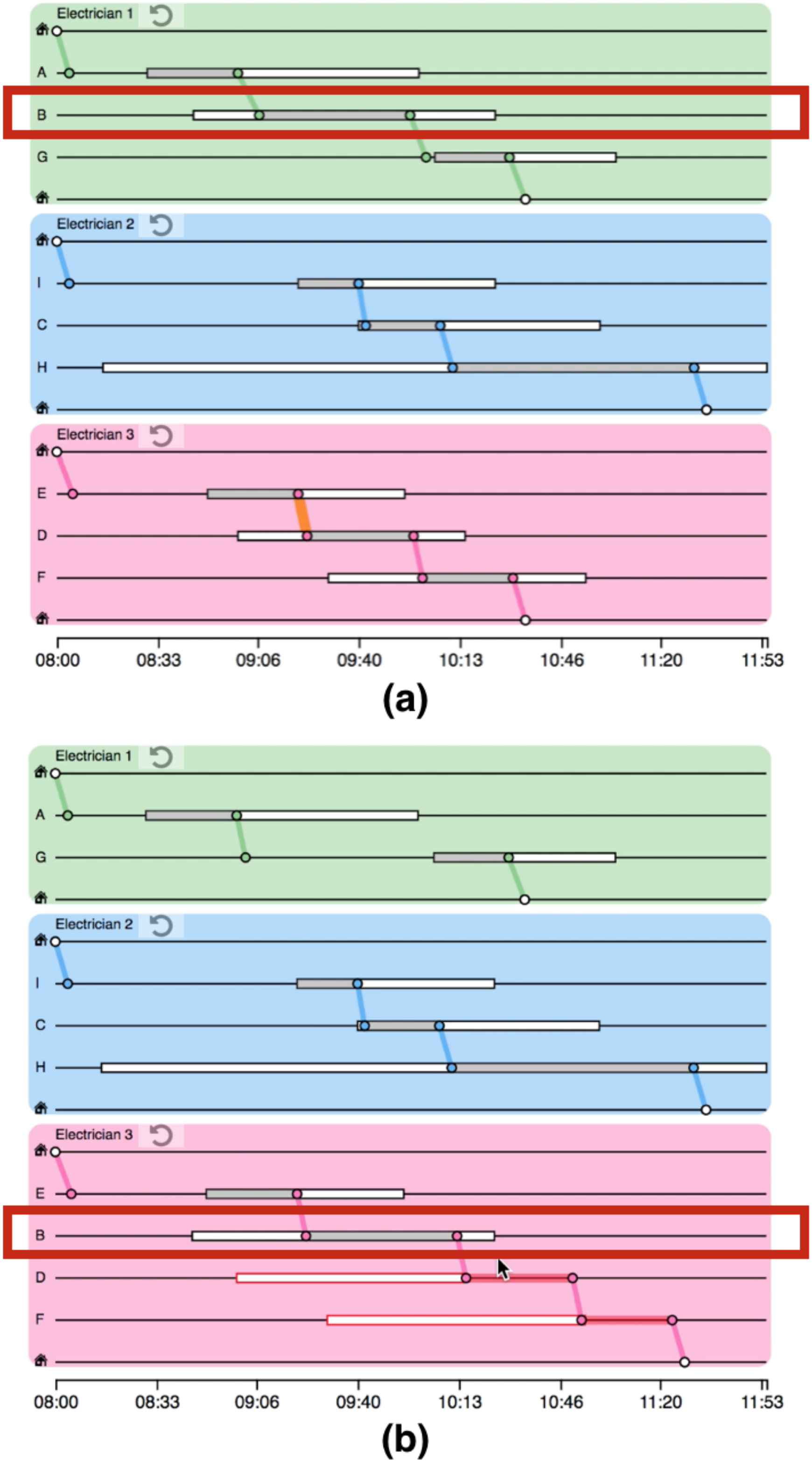

Figure 5.15: An example of the re-assignment of a customer. (a) before Customer B is re-assigned to Electrician 3; (b) After Customer B is re-assigned to Electrician 3. As a result, the timeline is updated after the re-assignment.

Figure 5.16: The solution gallery containing four solutions. The user can rename any solution. The plus sign in the fifth slot can be used to generate different solutions. The gallery can also sort solutions based on a particular criterion.

local re-optimisation of customers within a single electrician to minimise travel distance by that electrician. The re-optimise button is greyed out if the current customer arrangement for the electrician is optimal in the sense that the travel distance is minimised (see Fig. 5.13). In this case, the greyed button indicates the local optimality of the travel distance by an electrician. The button can also grey out if any constraints violation happens and it cannot be automatically resolved by the optimisation solver (see Fig. 5.14). In this situation, the greyed button in combination with the violated time window highlighted in red suggests the violation has to be addressed by the users before any re-optimisation.

The gallery & comparison.
Following the ***Guideline 4*. Gallery of solutions**, the second main viewport is designed for this. The solution gallery is used to manage solutions (see Fig. 5.16), just like a folder storing different files. One basic feature of the solution gallery is editing solution names. Users can type and assign any name to a solution, such as using the value of the objective to name a solution or simply typing the rank of a solution. Another feature is the ability to re-arrange solutions via drag-and-drop. Users can use it to rank all solutions in the gallery based on their objective values. In such a way, users can quickly select the most satisfying solutions and carry out further detailed analysis to verify and eventually find out the best solution.

The gallery also allows users to generate different solutions, following the ***Guideline 7:* Generating different solutions**. [1] There is an empty slot in the gallery placed on the right side of the last solution with a plus sign on it. When the user clicks the plus sign, the underlying optimisation solver will produce a solution with very different characteristics compared with solutions in the gallery. When the user is not satisfied with a particular solution, it can be removed by clicking the minus sign on the top right corner of the solution thumbnail.

In order to compare two solutions in detail, we design the interface to be able to present two solution side by side as recommended by the ***Guideline 6:* Side-by-side comparisons of two (or perhaps more) solutions**. A screenshot of the interface is displayed in Fig. 5.17. Each solution includes the map for routes display, the timeline to present time windows constraints and the statistics for the objective and high-level constraints display. The statistics for both solutions are merged to give a more compact display.

The history.
In accord with ***Guideline 9*: History of solution manipulation**, there is a viewport to keep track of the user's interaction history. We designed a histogram to record all

[1] This is a prototype, rather than a functional interface. In this prototype, we assume that different solutions can be generated. They are stored in the solution gallery.

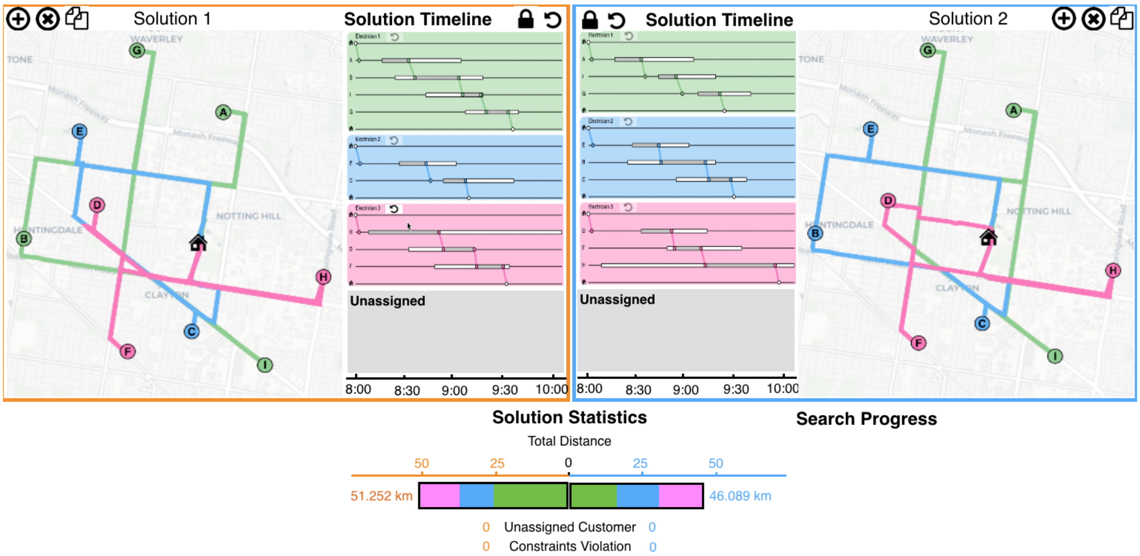

Figure 5.17: The side-by-side comparison of two solutions.

users' interactions (see Fig. 5.18). A single solution is presented as a bar. The height of the bar shows the objective value. The width of all bars is the same. Whenever user interaction finishes, a new bar will be drawn to reflect changes caused by the interaction. If an interaction makes a solution invalid by introducing constraints violations, the bar is highlighted in red. Hovering over a bar temporarily loads the corresponding solution to the interface with all viewports updated to display the solution. When the hover is finished, the interface reloads the latest solution before the hover. Single-clicking on a bar replaces the current solution and reloads the selected solution to the interface.

Temporarily loading a solution via hovering reduced users' memory load so that they do not have to remember the objective value and how routes look like. Reloading a solution via clicking allows user to go to any solution without worrying about making mistakes. Also the histogram itself provides the provenance of all interactions of a solution. Users can see how a solution is changed and where the changes come from. Users can go back a solution and explore different possibilities by making different changes. This provides users with the opportunity and freedom to better focus on solution evaluation and solution improvement tasks.

The statistics & solver progress.
We have also designed a viewport to record how the objective values have changed for each representative interim solution during the search process. The solver outputs a solution every time a configuration with a lower objective than the previous solution is reached. We show each of these representative interim solutions in an animated sequence in the map viewport, together with an *objective line chart* that graphs the tour length for each of these interim solutions as they decrease over time as the solver finds better solutions (see Fig. 5.19). The whole idea of designing the objective line chart viewport is to make the optimisation solving process more transparent to users suggested by the ***Guideline 8*: Feedback on solving process.**

5.4.2 Back-end Solving Technique

The problems are modelled in MiniZinc [100], a free and open-source constraint modelling language. Each problem instance is modelled in a way to combine multiple routes into a grand route. [2× the number of routes] depots are created and duplicated so that 2 depots

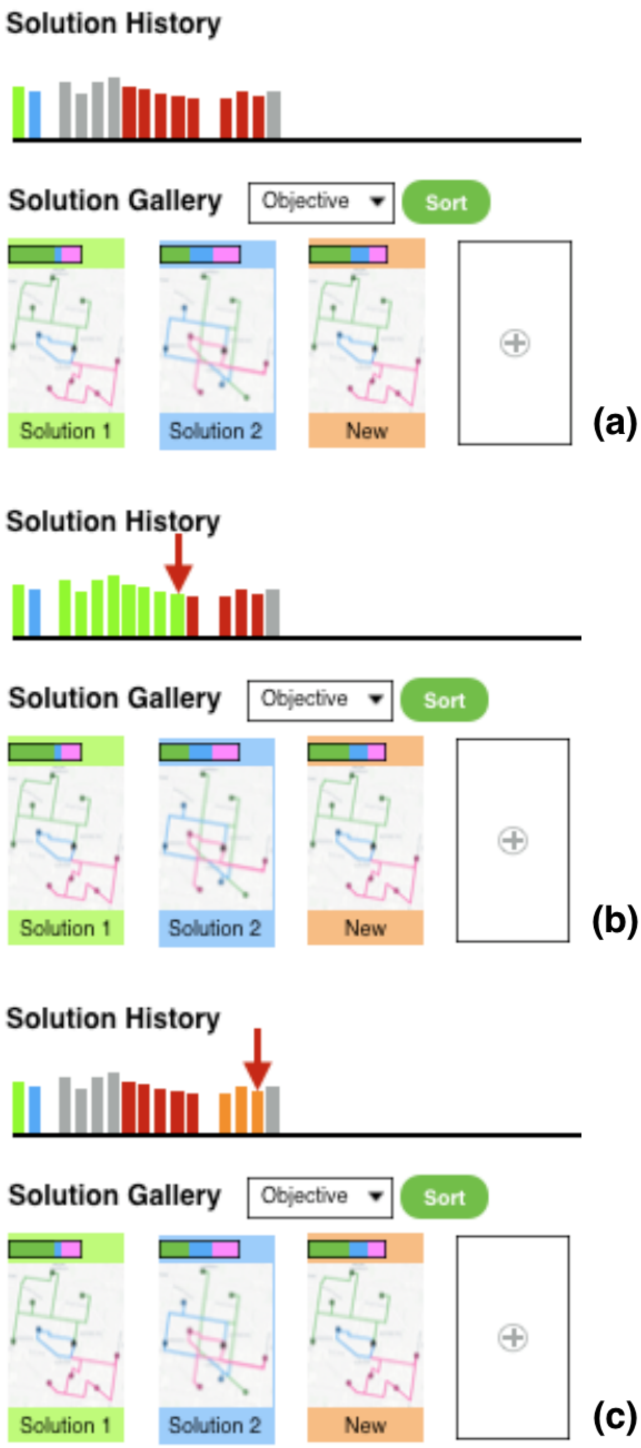

Figure 5.18: The solution history is presented together with the solution gallery to show the provenance of a solution: (a) no provenance is displayed before the mouse hovering on a particular solution; (b) the provenance of the first solution (in green) is shown; (c) the provenance of another solution (the third solution) is indicated in orange in the solution history.

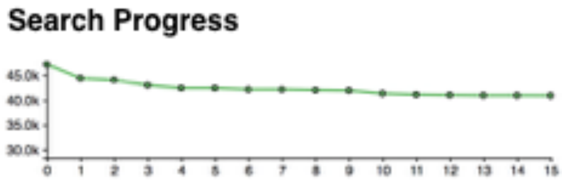

Figure 5.19: The objective line chart to present the optimisation solving progress (or search progress). It shows that the objectives of the interim solutions are decreasing over time indicating the solver finds better solutions.

are assigned to each route with one depot as the head of the route and another depot as the tail. Then, the tail depot of one route is connected to the head depot of another route to make a bigger route. Once every route is connected we have a single grand route. The benefit of doing this is it builds up connections between different routes so that instead of solving multiple small routing problems and merging the solutions, a single bigger routing problem is solved which is much faster and easier.

Each problem instance is solved using Gecode [101], an open source toolkit providing a constraint solver with state-of-the-art performance. Large neighbourhood search (LNS) technique is used to find good or near-optimal solutions. LNS have shown outstanding results in solving various typical optimisation problems including routing and scheduling problems [102–104]. LNS works by repeatedly transforming an existing solution into a different one in the neighbourhood of the solution. The neighbourhood is defined as a number of solutions modified from the original solution after a limited number of changes are made. LNS can iteratively improve an existing solution after searching its neighbourhood. It makes it possible to find a better solution after each iteration so that a very good solution can be found after a few of iterations. However, there is no guarantee that the solution is optimal. Because using the neighbourhood search in LNS can only guarantee the solution is locally optimal in the sense that it is the best solution in the current searched neighbourhood solutions. It is possible that a better solution can be found else where outside of the searched neighbourhood solutions. In a word, global optimally is not guaranteed in LNS which is one reason why human-in-the-loop optimisation may be beneficial.

5.5 A Heuristic evaluation of the interface prototype

We conducted a heuristic evaluation of the prototype human-in-the-loop optimisation tool for VRPTW described above. This was intended to verify that the guidelines had allowed us to come up with a useful and effective interface design.

The evaluation protocol was reviewed and approved by MUHREC. The heuristic evaluation project was titled “Investigating How to Build an Effective Interface in Interactive Optimisation”. The project ID was 17858.

5.5.1 Study Design:

Apparatus & Materials: The interface prototype was used throughout the evaluation. We prepared different sample solutions using the Gecode solver. All sample solutions had one depot, three electricians and less than ten customers to keep them easy to understand for the evaluation.

Participants: Four participants were recruited for this evaluation including a student from our institution and three employees from other organisations. All participants had normal vision and were without any colour vision impairment. All participants were not optimisation experts. The evaluation was run on a MacBook Pro notebook with a 13-inch screen (1280×800).

Procedure: The evaluation took less than one hour on average. The participants were guided to review the interface. The evaluation performed was an informal walkthrough with six predefined activities:

1. *Introduction Activity:* The problem context was briefly explained to participants. Afterwards our interface prototype was presented to participants and they were introduced to the functionality of each viewport including the map, the solution timeline, the solution statistics, the solution history and the solution gallery. The participants were asked to describe their first impression of the interface.
2. *Solving Activity:* Next, we demonstrated how to use the interface to generate a new solution. A special viewport called search processes was presented and explained. Then sorting solutions in the solution gallery based on the distance objective was demonstrated. The participants were asked a few questions evaluating the visualisation and interaction aspects of the interface.
3. *Modifying Activity:* Then we demonstrated how to manually change a solution by reassigning a customer to an electrician (recall from Section 5.3.2). We also explained how to duplicate an existing solution in the solution gallery. Afterwards, we demonstrated and explained how to explore the provenance of a particular solution in the solution history viewport. The participants were asked to evaluate and explain the usefulness of the operations used to manually change a solution.
4. *Re-optimising Activity:* Another approach to modifying a solution using re-optimisation was explained to the participants. Then they were asked about the usefulness of reoptimising a solution.
5. *Refining Activity:* Next, the two ways for refining the problem model were explained. They were adding or removing customers and refining the time window constraints of a customer. The participants were asked to evaluate the usefulness of both operations.
6. *Comparing Activity:* Finally, the participants were asked questions about how to compare two solutions, the criteria for comparison and the usefulness of displaying similarities and differences between solutions.

5.5.2 Results & Discussion

During the introduction activity, when the participants were asked the question: 'In general, how do you like the interface?', all four participants thought the interface was fairly easy to understand except that one participant reported it was a bit complex with many different "things" presented at the same time. All participants stated the interface was properly designed with viewports presenting critical information about a solution such as the routes on the map and time window constraints. When the participants were asked about the most satisfied part of the interface, one participant liked the solution history, another participant liked the solution gallery, the other two participants were satisfied with both the solution history and the solution history. In particular, one participant said:

> "The solution gallery gives me a very good overview of all solutions and the solution history tells me what changes I have made to which particular solution."

This is a good reflection on both ***Guideline 4:*** **Gallery of solutions** and ***Guideline 9:*** **History of solution manipulations**.

In the solving activity, all four participants preferred to solve a complex problem using a semi-automatic approach rather than manual and fully-automatic approach. In particular, one participant stated:

> "If I have to use a machine-based approach, I still have to verify [the machine produced solution]. I do not trust machines."

When the participants were asked the question: 'Are the solution and time window constraints presented in a proper way?' One participant responded:

> "It (the interface) includes everything I need to properly evaluate a solution."

Another participant said:

> "It is a multi-dimensional way to present the solution. It shows the routes. It also displays the timeline and we have statistics about the solution. There is nothing more to present. It has everything that should be presented."

This shows that ***Guideline 1: Appropriate visual representation of solution & constraints*** is well supported in our interface.

When the participants were asked about the usefulness of producing a new solution, all participants agreed that it was necessary to have the option to produce a new solution different from existing ones. In particular, one participant stated:

> "It is useful because I would like to see is there any other possibilities [to solve the problem in a different way]. If I was given this option, I would always use it."

This provides evidence in supporting ***Guideline 7:*** **Generating different solutions**.

All participants agreed that it was useful to provide feedback about the solving process such as displaying the search process line chart and showing the route transitions using animations. One participant responded:

> "It is similar when you reload a web page. Your mouse cursor is spinning. Psychologically, it gives you a feeling that it (the underlying optimisation solver) is working properly."

Another commented:

> "It increases my trust on it (the optimisation solver). Because it shows what it (the solver) has tried [to solve the problem]."

A participant also said:

> "Instead of waiting for the solver to produce a solution, it is an innovative approach. It clearly presents what it (the solver) is doing."

These comments provide strong support for ***Guideline 8:*** **Feedback on solving process**.

Lastly, participants were asked about the usefulness of being able to rearrange and annotate solutions in the solution gallery. Three out of four participants thought it was useful. One participant said it "might be useful" because the participant was not sure about how to use it in practical situations. Some of the participants' comments were:

> "If you like a solution, you can mark it just in case you may forget it later on."

> "It is useful especially when there are a lot of solutions in the gallery. Sometimes the difference between two solutions is subtle. Marking solutions and adding annotations make it easier for me to distinguish one from the other."

Again, this shows that ***Guideline 4*** is partially supported from the functionality aspect.

Next, during the modifying activity, all participants agreed that manually changing a solution is very useful. For example, one participant commented:

> "Manually changing a solution is necessary because I want to verify the solution by making changes when I do not trust the machine (optimisation solver)."

Another participant said:

> "Absolutely. I hope that I can have the control in determining whether or not to use this solution as well as making whatever changes that I want to make."

This shows the importance of ***Guideline 3:*** **Direct manipulation of solutions** and that our interface supports this.

A following up question was asked, 'How useful is it to provide a complete history of solution manipulations?' All the participants commented that it was very useful to have such a history including all changes. In particular, one participant said:

> "You may make a mistake when you are changing a solution, and sometimes you may forget what changes you have made. Having a complete history like this makes it possible to go back to earlier steps. It is also possible that you run into the situation where you always have solutions with violations, then you have to go back to the very beginning [before any violation appears] and start over again."

This clearly demonstrates that our interface supports ***Guideline 9*** and that this is necessary.

In the re-optimising activity, the participants were asked about the usefulness of being able to re-optimise a solution. All participants thought it was necessary to let the optimisation solver re-optimise a solution. In particular, two participants commented re-optimising all routes after adding constraints to prevent the assignment of certain customers to electricians (recall from Section 5.3.2) can be very useful. One participant stated:

> "In practical situations, re-optimising all routes by considering personal preferences and possible special needs is pretty useful."

Another participant said:

> "It is useful because you are given the option to completely resolve the problem with certain constraints applied."

Again these support the importance of ***Guideline 5:*** **User controlled re-optimise** and that our interface implements this guideline.

Afterwards, during the refining activity, all four participants agreed that it was critical to refine the problem when it had been changed, and it needed to be resolved. Our interface allows users to add or remove customers as well as changing the time window constraints of each customer. The participants thought it was common to add and remove customers and that changing an appointment's time or rescheduling frequently happened in everyday life. Thus, ***Guideline 2: Modifiable optimisation models*** is important and well reflected in the interface.

In the final comparing activity, participants were asked to give their opinions on the side-by-side comparison of two solutions. All four participants agreed on its usefulness. Some of the participants' comments were presented below:

> "I can clearly see both solutions, and I do not have to remember one solution or the other. I can just focus on the comparison."

> "The horizontal arrangement of two solutions works for me. It is better than the vertical arrangement. It is a more natural way to do the comparison with the horizontal layout."

One interesting comment was about using the side-by-side comparison to manually change a solution. The participant said:

> "It is useful when I try to change a solution. I can simply load the original solution on the left and change its copy on the right. Because sometimes you do forget what it originally looks like."

This shows that ***Guideline 6:*** **Side-by-side comparisons of two (or perhaps more) solutions** is both important and well supported in our interface.

5.6 Conclusions

In this chapter we have presented nine design guidelines to inform the development of human-in-the-loop optimisation interfaces. The guidelines were extracted and based on our *problem-solving* loop framework. They identify key components that should be included in such interfaces.

We revisited the interactive prototype tool developed for our first use case—prostate brachytherapy—to see how well the tool followed the nine design guidelines. It showed that the majority of the guidelines were reflected in the tool. However some guidelines were only partially reflected, and the last guideline was not supported in the tool. In hindsight, we would have produced a better tool if the guidelines had existed and we had been able to follow them.

In order to build a complete understanding of the human-in-the-loop optimisation and to generalise our research findings from the prostate brachytherapy use case, we decided to investigate a second use case, one which required less expertise and was easy to understand by the general public. We chose VRPTW.

We spent considerable time developing the interface prototype for this second use case. We carefully used the nine design guidelines to inform the development of the prototype, ensuring each guideline was well supported.

Afterwards, we conducted a heuristic evaluation of the interface prototype with four participants. In general, it showed that the design of the prototype had a good reflection of the nine design guidelines and reinforced the usefulness of each guideline. The participants were satisfied with the presentation of solutions as well as the objective and constraints. In particular, the participants thought the combination of the solution history and solution gallery worked well in terms of modifying a solution and comparing multiple solutions.

Chapter 6

Engender Appropriate Trust

Trust plays a critical role in determining users' acceptance of a system. Too much or too little trust are equally dangerous. Often users place little trust in a fully-automatic optimisation system [11]. Because users do not understand the underlying algorithms used by the system. The "black-box" nature of such a fully-automatic optimisation system is the primary reason contributing to the lack of trust.

As previously discussed in Section 2.1.2, overtrust can also happen. David and Kottemann [31] reported the user tended to trust the system more when enough controls were given. They called this phenomenon the "illusion of control".

In this chapter, we investigate how to engender an appropriate level of users' trust in optimisation systems based on the vehicle routing problem with time windows case study introduced in Chapter 5. In particular, we investigate two effects on users' trust: feedback and interaction. We conducted two controlled experiments to do so. Specifically, in the first experiment, we provided users with feedback about intermediate solutions and the objective function to test how trust is affected. In the second experiment, we allowed users to semi-automatically manipulate solutions returned by an optimisation solver to verify its effect on users' trust. The experiment protocols were reviewed and approved by MUHREC. The experiment project was titled "Investigating Interaction on Solution Assessment in Interactive Optimisation". The project ID was 12340.

6.1 Related Work

Over the past four decades there have been hundreds of papers written about trust in automation. The following review is based on Lee and See's influential theoretical framework [11] and the three-layered trust framework of Hoff and Bashir [10]. Fig. 6.1 shows the combined framework.

Lee and See [11] 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

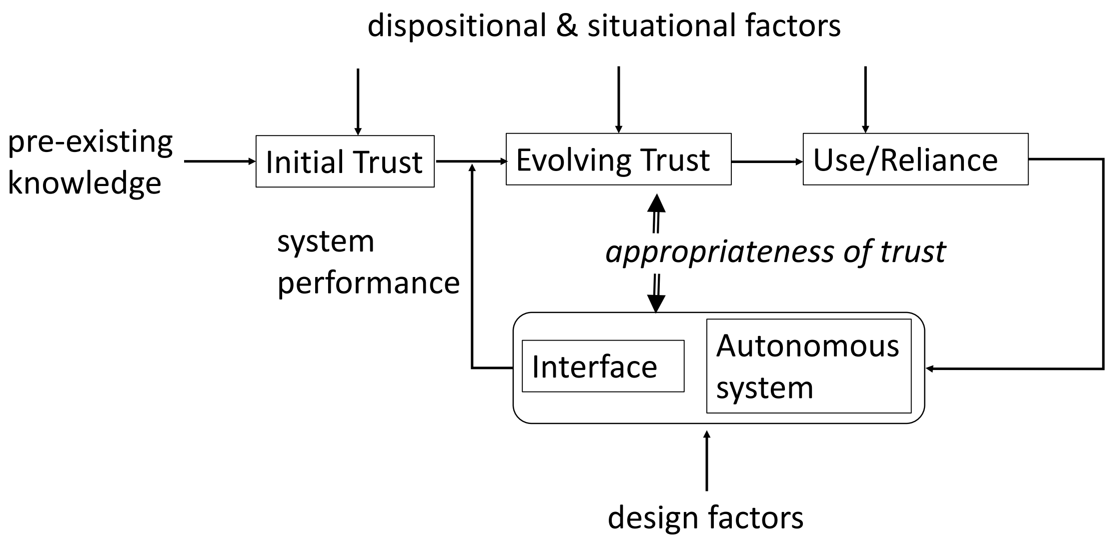

Figure 6.1: High-level framework for understanding the factors that influence trust and reliance on autonomous systems. It incorporates the three-layered trust framework of Hoff and Bashir [10] into the framework of Lee and See [11].

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 [10] 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 [105]. 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. [106, 107]. As also discussed by Schaefer et al. [21], 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. [108–111]. 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 [112], level of control [113], 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 [114–117]. With the arrival of the internet the focus was on trust in on-line systems. Corritore et al. [118] 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 [112, 119], cyberdomains [120], recommendation systems [73, 74], adaptive agent systems [74], and information security classification [76]. More recently, AI applications such as autonomous vehicles [121], robots [122], medical assistance devices [121] and machine learning [77, 123] have received attention.

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 [1], user studies evaluating interactive optimisation systems have focussed on solution quality and time spent to find a solution rather than trust [3, 4, 6, 19, 30, 38, 40, 51, 56, 57]. The only study we are aware of that has investigated trust in optimisation is from our previous work [124]. We 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 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.

Thus, the two studies presented in this chapter 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 measure system performance.

6.2 Study 1: Effect of 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.

6.2.1 Experimental System Design

In this study, we used two different on-line solvers: a *good* solver and a *poor* solver. Both on-line solvers were simulated based on the results computed from off-line solvers. Specifically we solved each problem instance offline before conducting the study, using state-of-the-art constraint solving technology (the problems were modelled in the MiniZinc constraint modelling language [100] and solved with Gecode [101] 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 30% worse, compared to the best solution found. The good on-line solver shows the best solutions found in the offline computations. The poor on-line solver displays the 30% worse solutions. 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. A sample solution from each on-line solver is shown in Fig. 6.2.

We also defined two experimental conditions: *feedback* and *non-feedback* (see Fig. 6.3). The *solver progress* viewport was only shown in the feedback condition. The solver output a solution every time a configuration with a lower objective than the previous solution is reached. Recall from Chapter 5, the solver progress viewport showed interim solutions in

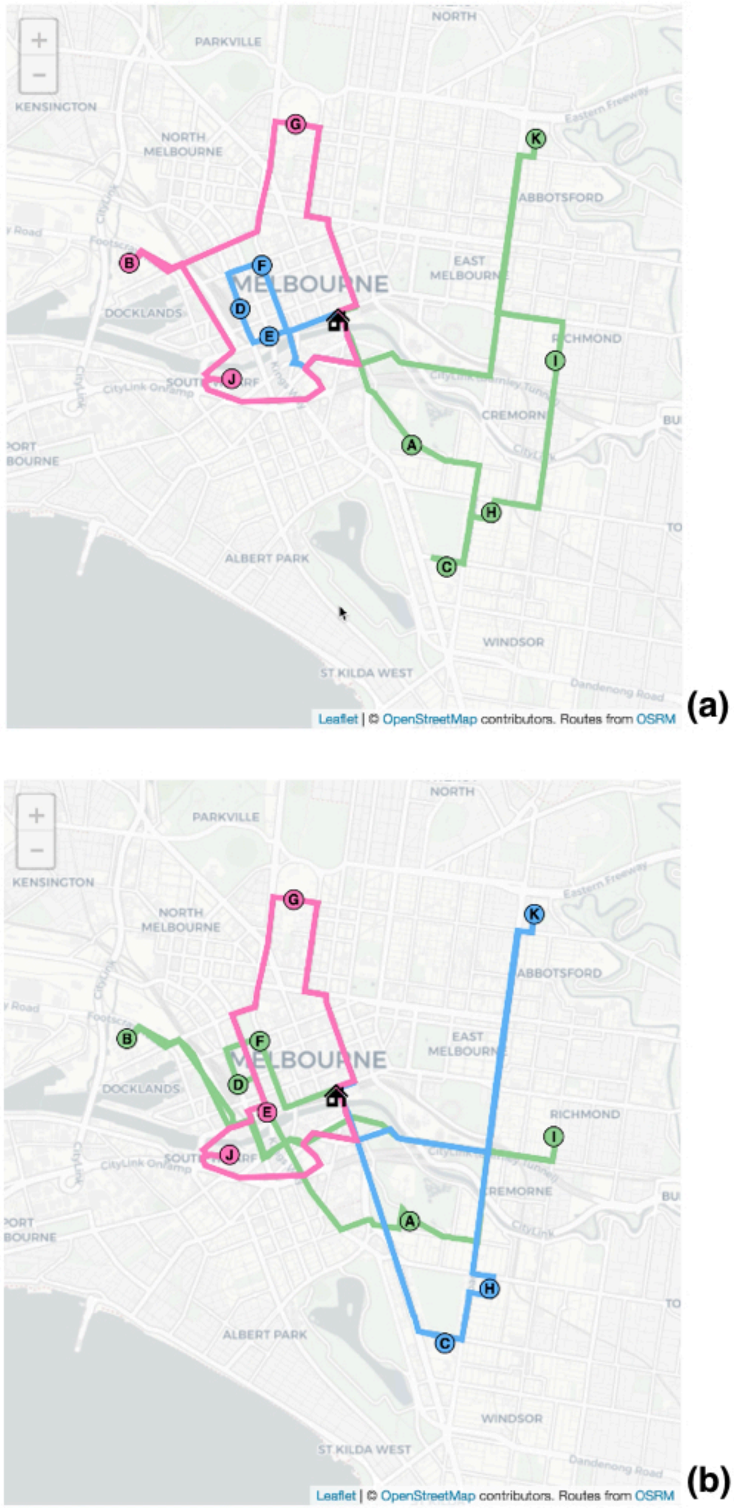

Figure 6.2: Customers are represented as circles with letters in it. Routes are coloured differently to distinguish electricians. The home symbol in the middle is the depot. (a) A sample solution produced by the good on-line solver; (b) A sample solution to the same problem instance produced by the poor on-line solver.

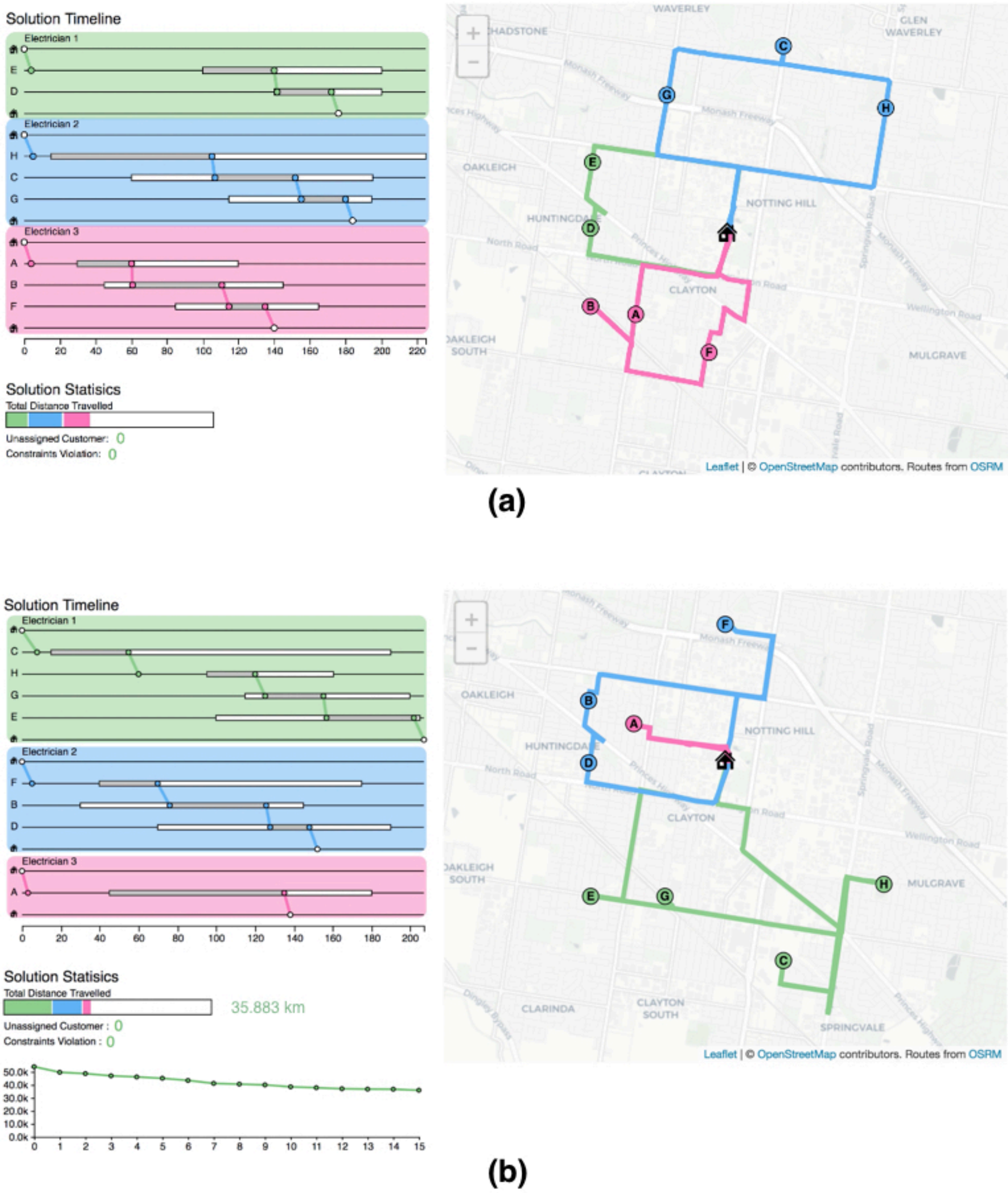

Figure 6.3: The study 1 interface under two conditions: (a) non-feedback condition; (b) feedback condition. Notice: the problem instances in (a) and (b) are different.

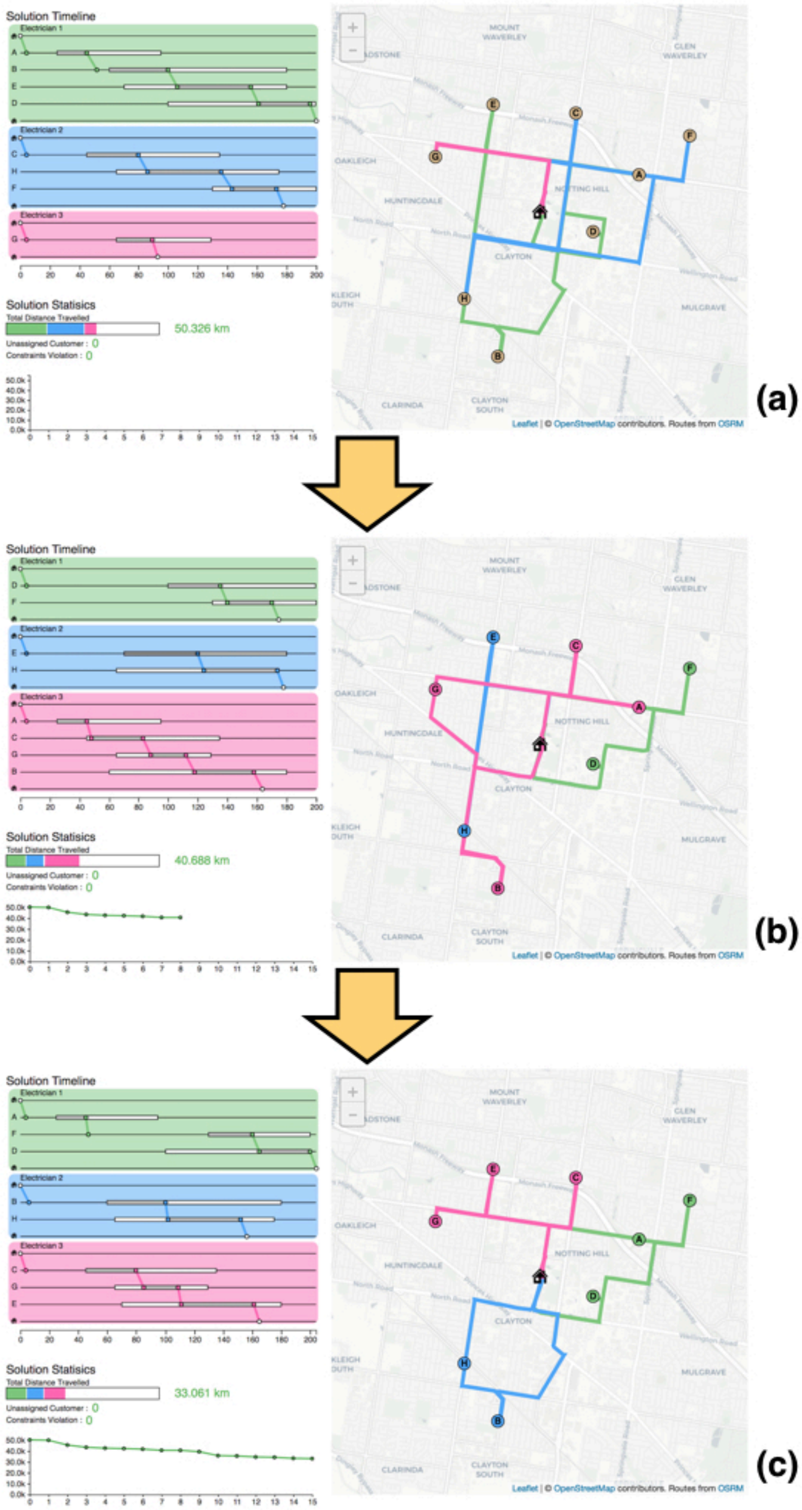

Figure 6.4: Study 1 interface with feedback on solver progress. (a) first solution; (b) an intermediate solution; (c) final solution.

an animated sequence, together with an *objective line chart* that graphed the tour length for each of these interim solutions as they decreased over time as the solver found better solutions. Fig. 6.4 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.

There were three other viewports used in both conditions: *map*, *timeline* and *statistics*. Colour is used to distinguish individual electricians, the same colours being used within each view to identify electricians across all three visuals [125]. Recall from Chapter 5, we represented solutions in the map viewport. It contains geographic information about the depot, customers locations and electricians' travel routes. we used the timeline viewport to show customers' time windows as well as electricians' service periods for each task. The total travel distance objective is numerically presented in the statistic viewport.

6.2.2 Participants and Setting

We recruited 28 participants including students, researchers from our institution and a few employees from other organisations. 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 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. [1] 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.

6.2.3 Tasks and Design

For each problem instance, participants were asked to evaluate solutions returned by each solver (scale 1–7). We then summed these over the four problem instances, to attain an aggregate *solution quality* for each solver in the range (4–28). After evaluating all solutions produced by the same optimisation solver, participants were asked to evaluate the optimisation solver from different aspects (each 1–7). The full text of the questions asked and their sources in past work is as follows:

Functionality [21, 126, 127]–the competence of the optimisation solver to solve the problem: "To what extent does the optimisation solver perform its function properly?"

Understanding [21, 75, 77]–user understanding of the solver's operation: "How well do you understand the strategy used by the solver to find this solution?"

Dependability [21, 114, 126–128]–how reliable the solver is: "To what extent can you count on the optimisation solver to do its job?"

Consistency [114, 126, 128]–how consistently the solver performs on different problems: "To what extent does the optimisation solver perform consistently?"

Satisfaction [21, 75, 127]–acceptability of the solutions: "How satisfied are you with the performance of the optimisation solver?"

[1] In particular, experts here refer to either optimisation experts or experts specialised in the VRP problem (not just VRPTW, but VRP in general) or both.

Faith [114, 128]–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 [127]–the overall degree of trust in the solver: "Overall, how much do you trust the optimisation solver?"

The trust measures are based on those used in the trust literature but modified for an optimisation context. 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 [114]. Single-item measures of trust have been found to be reliable [129].

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. We introduced two levels of difficulty, varying in customer numbers and electrician numbers. After piloting, each *easy* problem instance had 8 customers and 3 electricians and each *hard* problem instance had 15 customers and 4 electricians.

The experiment was within-subject: 28 participants $\times$ 4 optimisation solvers $\times$ (2 difficulty levels $\times$ 2 problem instances $\times$ 1 solution evaluation question + 7 solver evaluation questions) = 1,232 responses (44 responses per participant). 4 optimisation solvers = 2 feedback options $\times$ 2 solution quality levels. *Easy* problem instances were always presented before *hard* instances.

6.2.4 Procedure

After answering a short questionnaire capturing demographic information and expertise, the eye tracker was recalibrated for each participant. The eye tracker was used for investigating the time that users spend on different areas of interests. In this case, the areas of interests were the map, the timeline and the objective line chart. We wanted to know how users make use of the different components in the interface when they were asked to evaluate a solution.

Then participants were *trained* in the use of the system. Training was designed to ensure that all participants thoroughly understood the goal of the optimisation process and how to evaluate the quality of a routing solution returned by a solver. A short video was presented as a brief introduction to the experiment. 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. This was followed by a sample solution with 5 questions to further test participant's understanding of the experiment 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.

During the *experiment* participants were asked to evaluate the four optimisation solvers. For each solver they were shown each sample solution on a separate page and asked to rate its quality. Participants were then asked to answer 7 questions evaluating the solver. At the end of the study a post-questionnaire was used to record participants' feedback.

A 7-point Likert scale was used to capture answers to both evaluation tasks. Latin square design was used to balance the order of solvers. [2]

6.2.5 Results and Discussion

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. 6.5.

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 analysis of variance (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 honestly significant difference (HSD) post hoc tests on significant main effects [130]. The statistical analysis is shown in Fig. 6.6.

The analysis showed that feedback significantly increased rating of participant's understanding of the solver. There were few statistically significant differences between experts and non-experts. However non-experts' rating of functionality of the poor solver was significantly higher than that of the experts.

The most important result of the analysis is that adding feedback significantly increased participant's 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. This provides strong support for our hypothesis that providing feedback about intermediate solutions increases user trust in optimisation.

This increased level of trust, however, was largely unwarranted. 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 feedback–see Fig. 6.7. This is extraordinary, and shows that providing feedback on intermediate solutions can lead to over-trust even by experts.

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 thought the objective line chart is useful, 5 out of 28 (17.86%) participants thought it is not that useful and they did not use it during the study, All the remaining 2 (7.14%) participants 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 is 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."

[2]Latin square design was used to shuffle different solvers presented to different participants to control the possible bias caused by the solver presentation order. We also tried to keep the presentation orders equally distributed and assigned to participants.

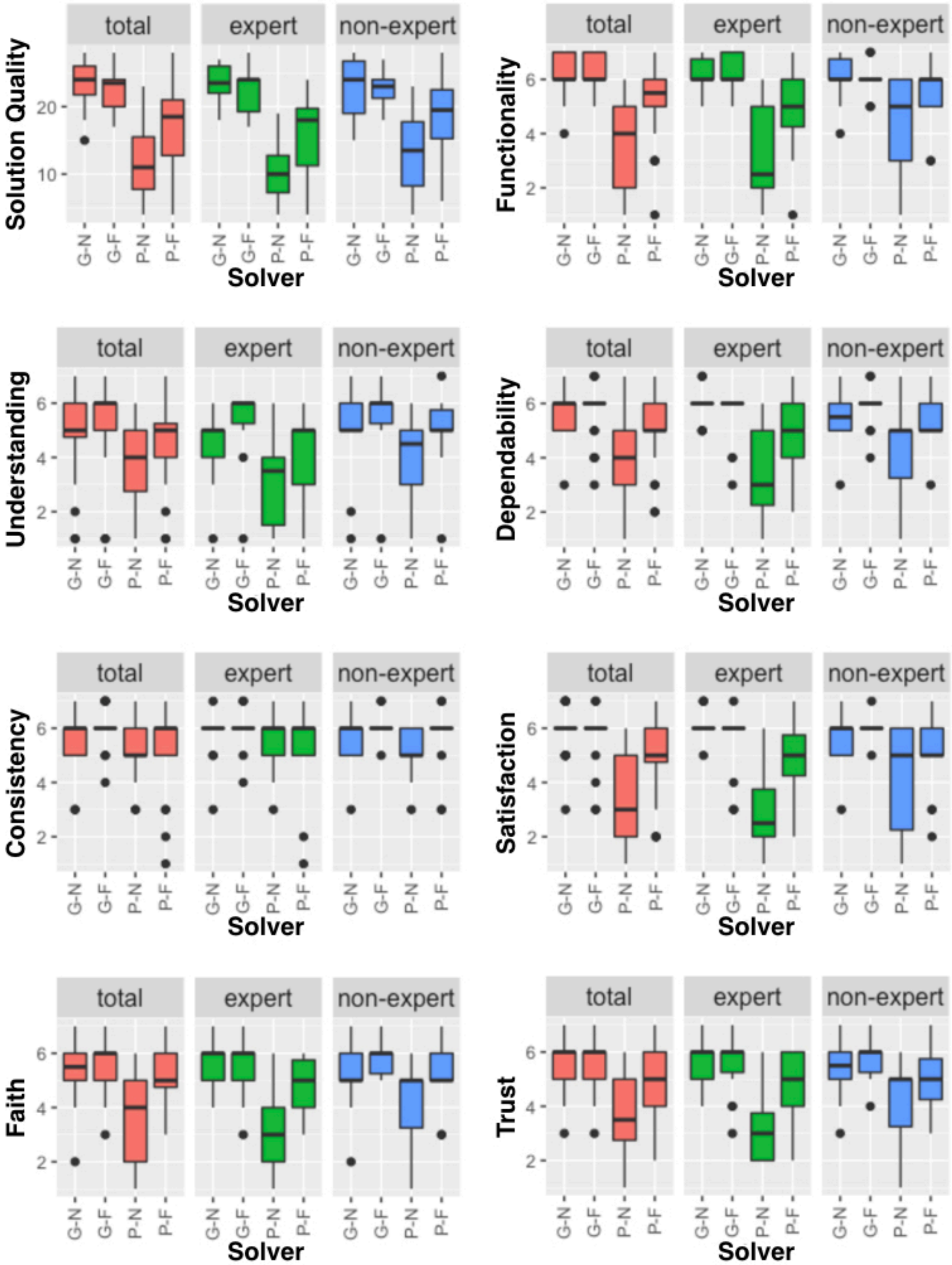

Figure 6.5: Study 1 aggregate *solution quality*, solver *functionality*, *understanding*, *dependability*, *consistency*, *satisfaction*, *faith* and overall *trust* measures. For the x-axis labels the first letter represents the solver quality: *G – Good*; *P – Poor* and the second letter represents the feedback option: *N – Non-feedback*; *F – Feedback*. So G–N represents the good solver without feedback.

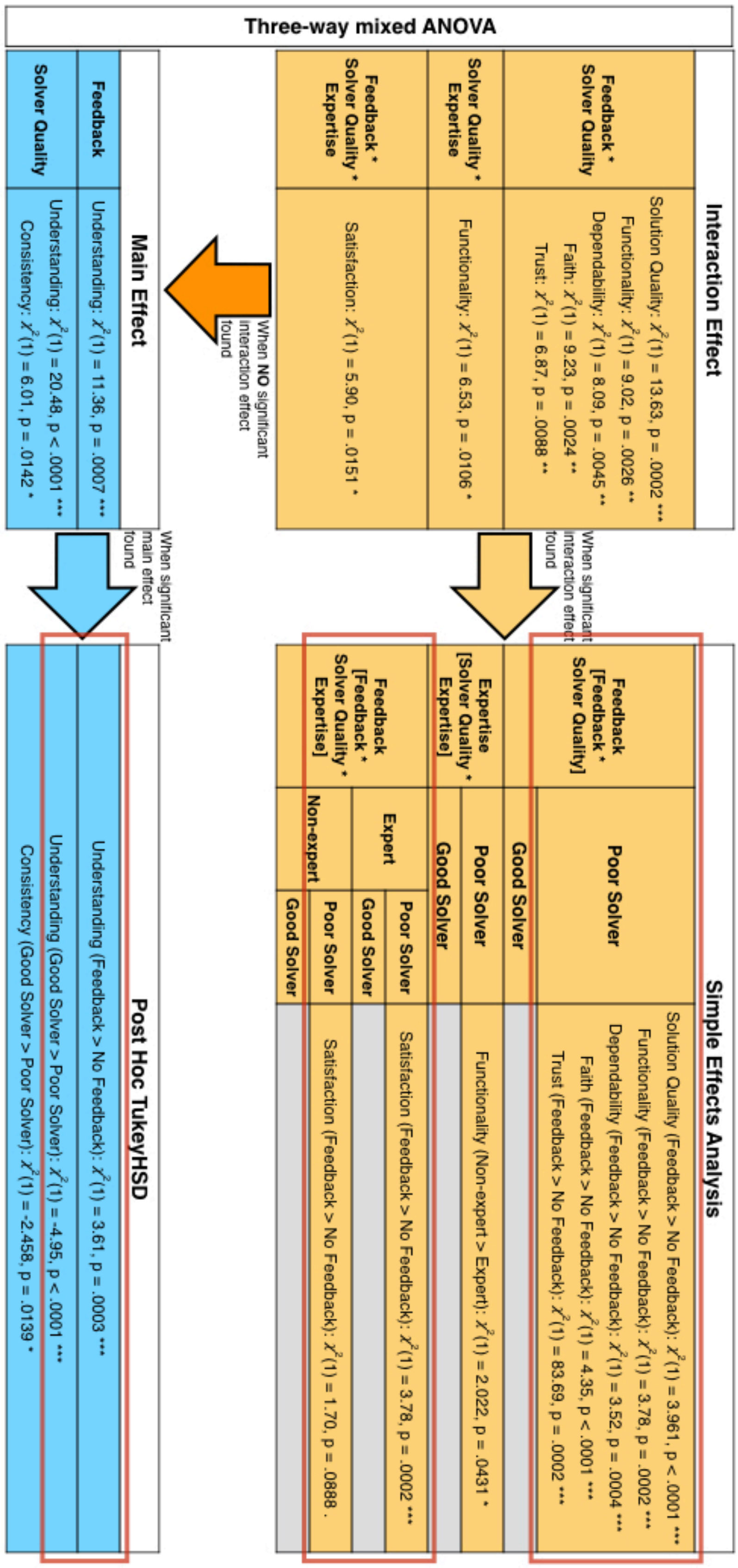

Figure 6.6: Study 1 statistical analysis results. Noteworthy findings are indicated by a red rectangular outline. Significance codes are as follow: p $\leqslant$ 0.001 (***); p $\leqslant$ 0.01 (**); p $\leqslant$ 0.05 (*); p $\leqslant$ 0.1 (.).

Solution Quality

Solvers	Good—F	Good—NF
Poor—F ≥	6 (3 E + 3 NE)	5 (3 E + 2 NE)
Poor—NF ≥	1 (0 E + 1 NE)	2 (0 E + 2 NE)

Overall Solver Trust

Solvers	Good—F	Good—NF
Poor—F ≥	13 (6 E + 7 NE)	13 (5 E + 8 NE)
Poor—NF ≥	8 (3 E + 5 NE)	5 (1 E + 4 NE)

Figure 6.7: Study 1: number of participants giving equal or better rating to the poor solvers as compared to the good solvers: *E – Expert*, *NE – Non-expert*.

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

> "Proving 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."

After analysing the eye-tracker data from 24 of the 28 participants (4 participants did not use the eye tracker because of location issues.), it revealed that 18 out of the 24 (75%) participants have spent more than 4 seconds looking at the objective line chart on average for each problem instance. Overall participants have spent nearly 100 seconds evaluating the quality of a solution for each problem instance. It looks like participants did not spend much time on looking at the objective line chart. However, it does not mean that participants did not make good use of it. A possible explanation is that the visualisation itself is simple to see and easy to understand. It is mainly about displaying the trend of the solving process to indicate that the solver is finding better and better solutions. It is rather trivial. However, participants have comments asking whether they can see each interim solution in more detail. The chart is not interactive for this experiment. Thus, they cannot check interim solutions. However, we did change it to an interactive histogram in our second experiment to support the checking of interim solutions.

6.3 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. Because of the number of conditions a between-participant design was used. The tasks and protocol were the same as the first study.

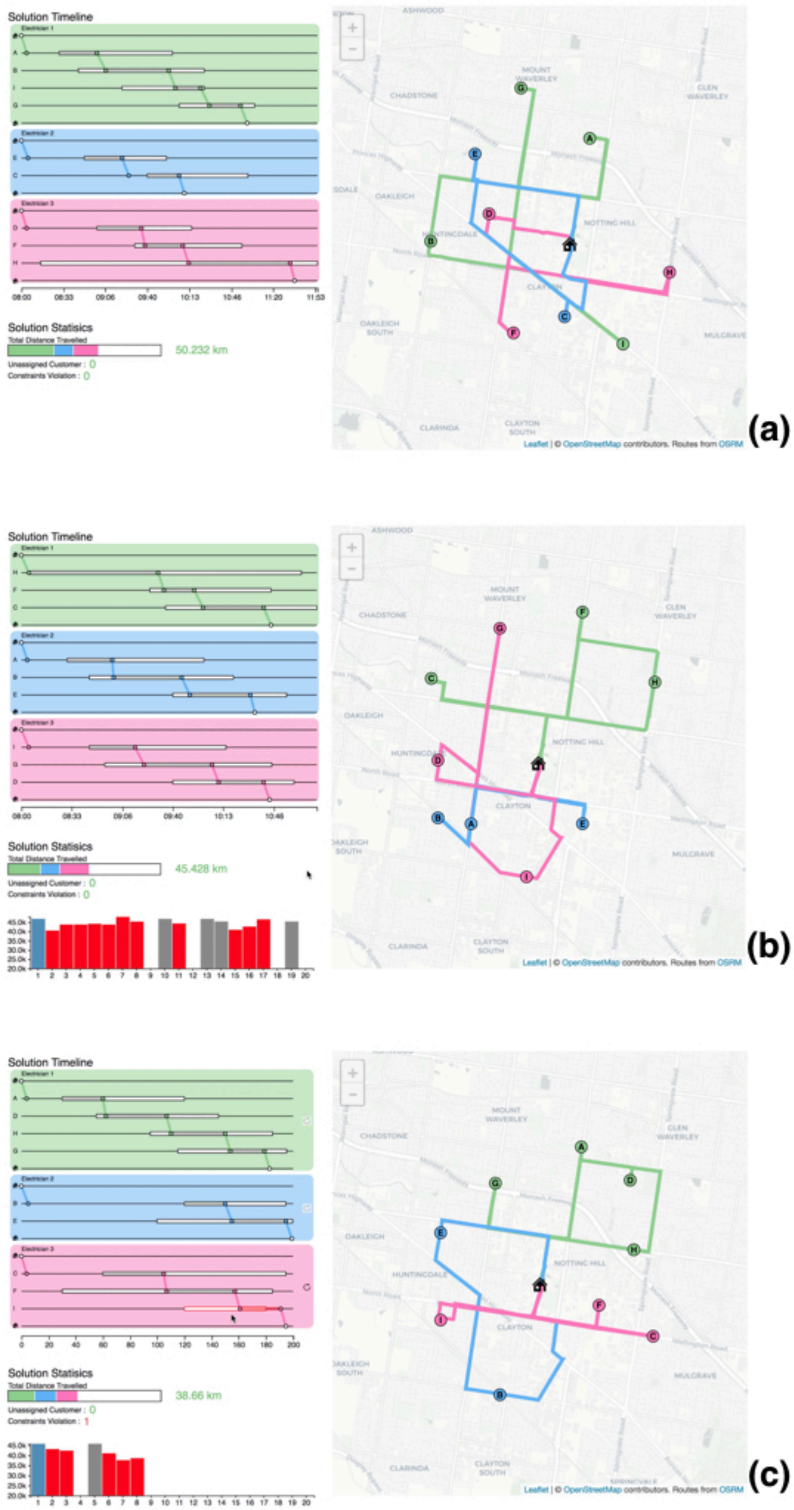

Figure 6.8: The study 2 interface under three conditions: (a) no interaction (NI) condition; (b) manual interaction (MI) condition; (c) semi-automatic interaction (SI) condition. The NI condition in study 2 is the same as the non-feedback condition in study 1. In both MI and SI conditions, the solution history is represented under the solution statistics. Moreover, in the SI condition, a re-optimise button is added to each electrician. The button turns off (greys out) when either the routes are optimal or there is at least one violation in the routes. The button turns on when the optimisation solver finds a better solution with a shorter total distance of the routes. Notice: the problem instances in (a), (b) and (c) are different.

6.3.1 Experimental System Design

In Study 2, we had three conditions: *No Interaction (NI)*, *Manual Interaction (MI)* and *Semi-automatic Interaction (SI)* (see Fig. 6.8).

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.

An example of a participant exploring different solutions in the SI condition is shown in Fig. 6.9. Our map, timeline and solver statistics views were the same as in Study 1, supporting the same basic brushing and hovering interactions, and were provided in all three conditions. We used the objective histogram to replace the line chart view of objectives used in the *feedback* condition of Study 1 to provide a more suitable interface for interaction.

From study 1, we noticed that most of the time, participants could easily distinguish the 30% worse poor solver from the good solver. To make our second study more challenging, we added a new medium on-line solver, which displays the 15% worse solution compared to the best solution.

6.3.2 Participants, Setting and Data

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. We divided the 30 participants into 3 groups and assigned them into 3 different conditions: NI, MI and SI. The study was run on a MacBook Pro notebook with a 2.6 GHz Intel i5 processor and a 13-inch screen (1280 $\times$ 800).

We used the same approach from Study 1 to generate different problem instances for this study. However, we added one more customer for *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.

The *training* was extended to explain solution manipulation to participants in the *SI* and *MI* conditions at the start of the hands-on exercises. They were encouraged to use interaction freely during the *experiment*. Participants responses were timed and a maximum of 5 minutes was allowed for each problem instance, to control the overall experiment time.

The post-questionnaire was adapted and restructured to ask participants about the effect of interaction on trust.

6.3.3 Design

The experiment was between-subject: 10 participants $\times$ 3 experimental conditions $\times$ 3 optimisation solvers $\times$ (2 difficulty levels $\times$ 3 problem instances $\times$ 1 solution evaluation question + 7 solver evaluation questions) = 1,170 responses (39 responses per participant). Again *easy* problem instances were presented before *hard* instances.

Figure 6.9: 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.

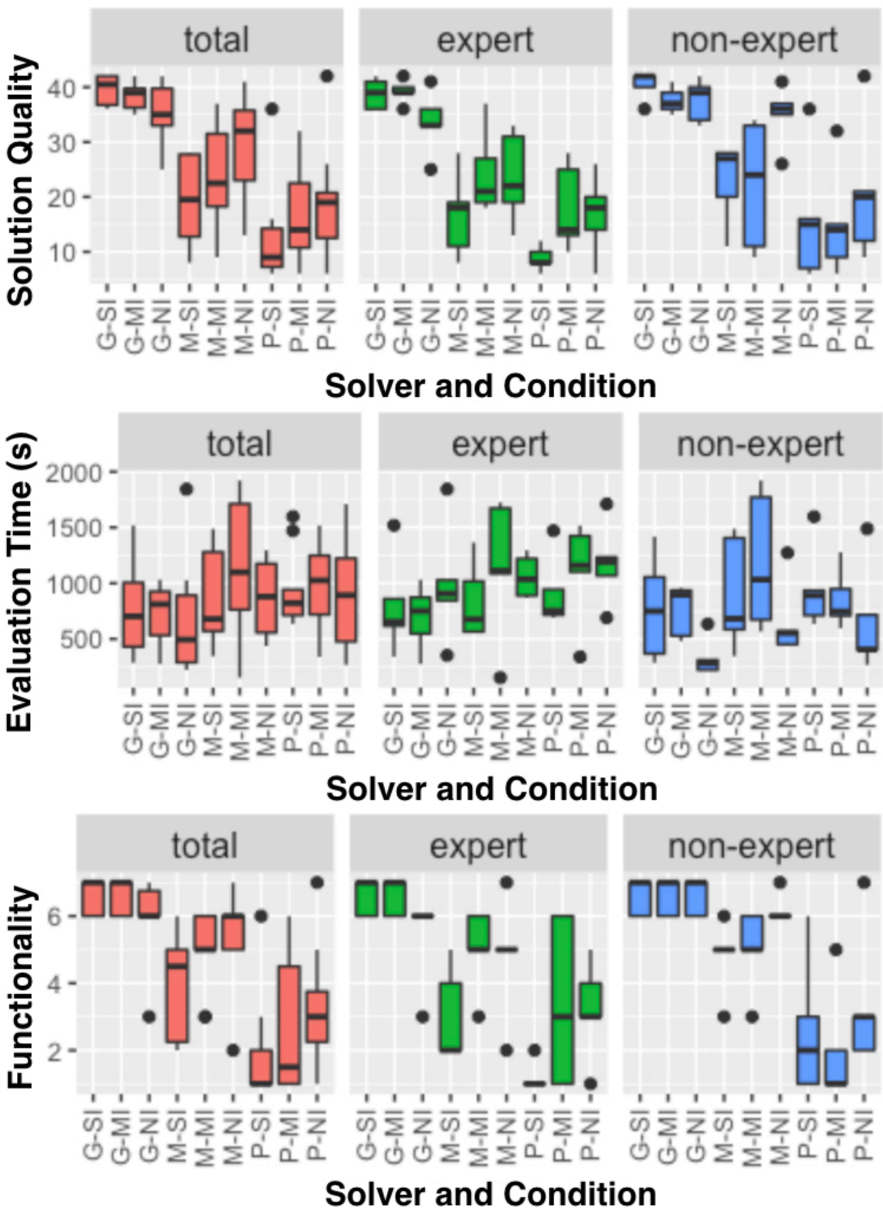

Figure 6.10: Study 2 aggregate *solution quality*, total solution evaluation time (in seconds) and the solver functionality measure. For the x-axis labels the first letter represents solver quality: *G – Good*; *M – Medium*; *P – Poor* and the second double letters represent experimental condition: *SI – Semi-automatic Interaction*; *MI – Manual Interaction*; *NI – No Interaction*. So G–SI represents the good solver in the semi-automatic interaction condition.

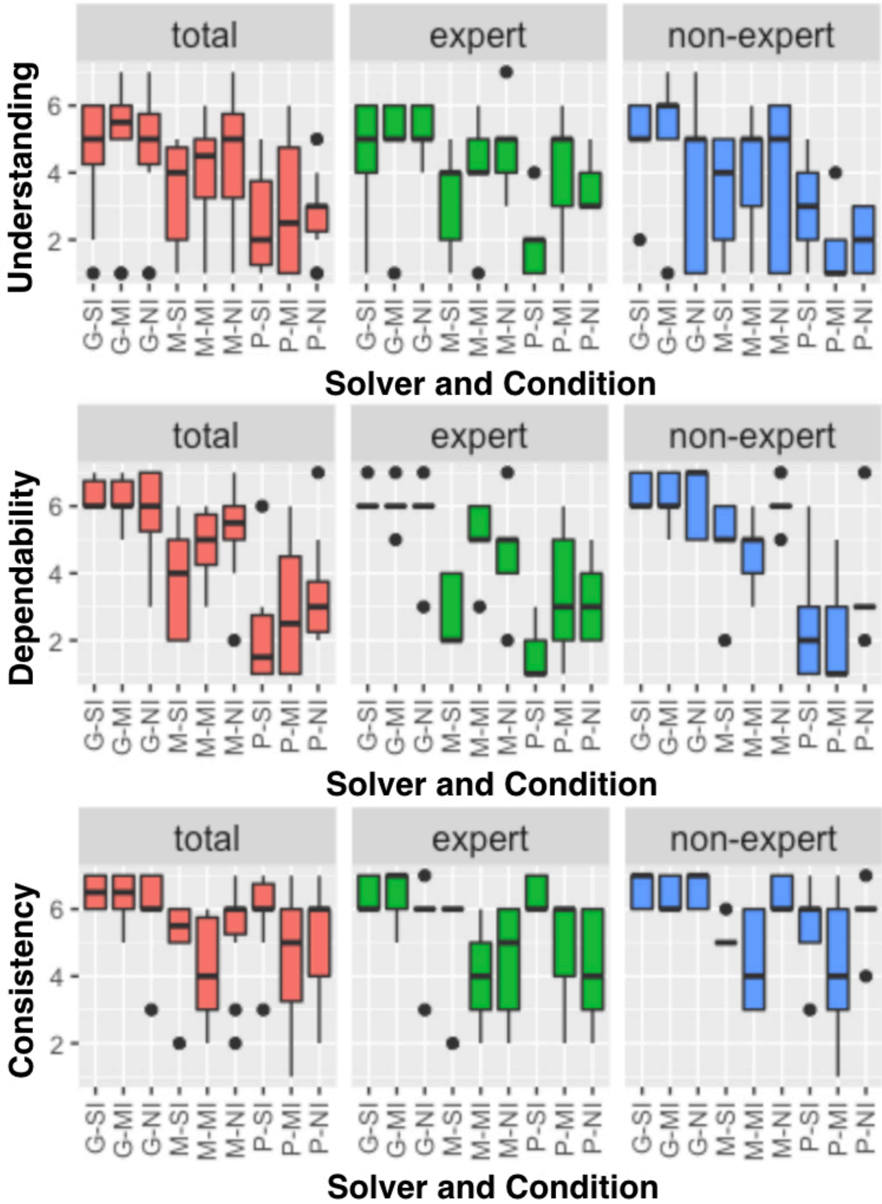

Figure 6.11: Study 2 the solver understanding, dependability and consistency measures.

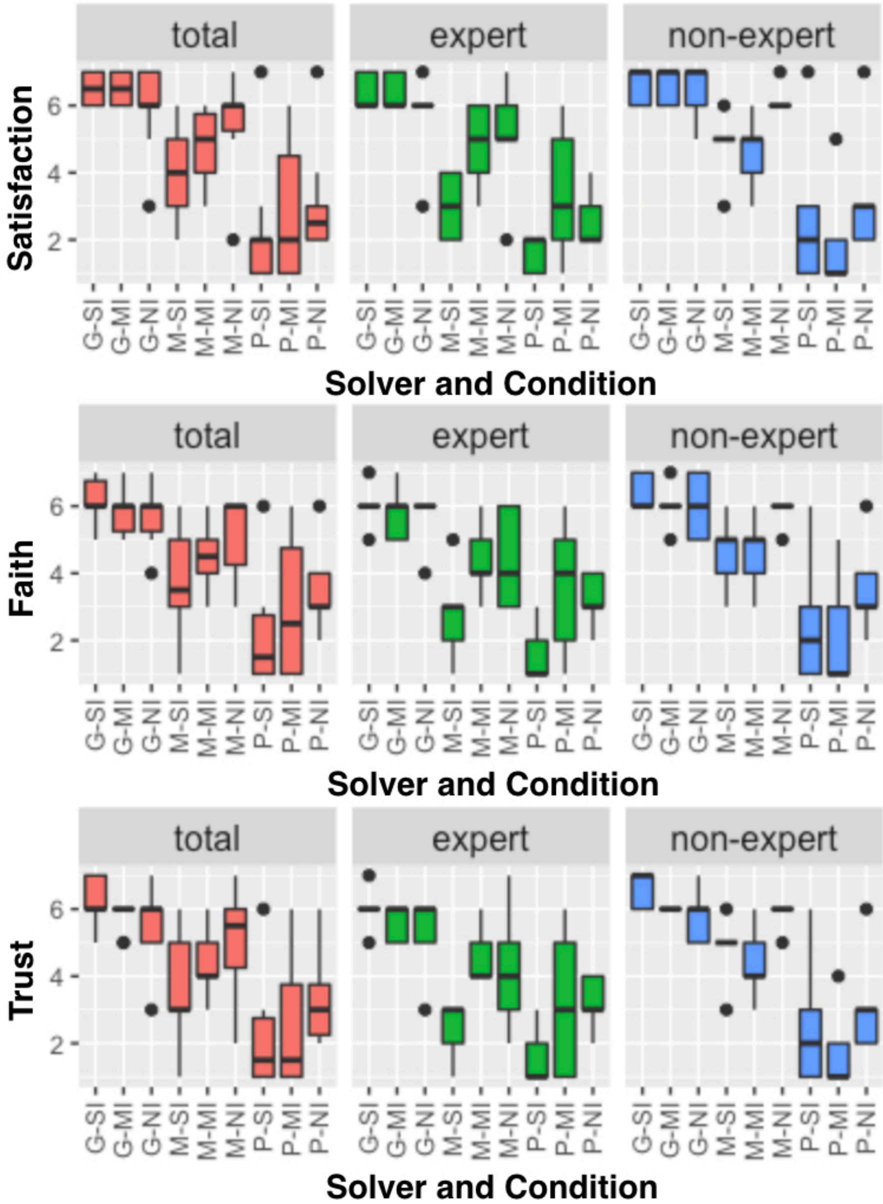

Figure 6.12: Study 2 the solver satisfaction, faith and overall trust measures.

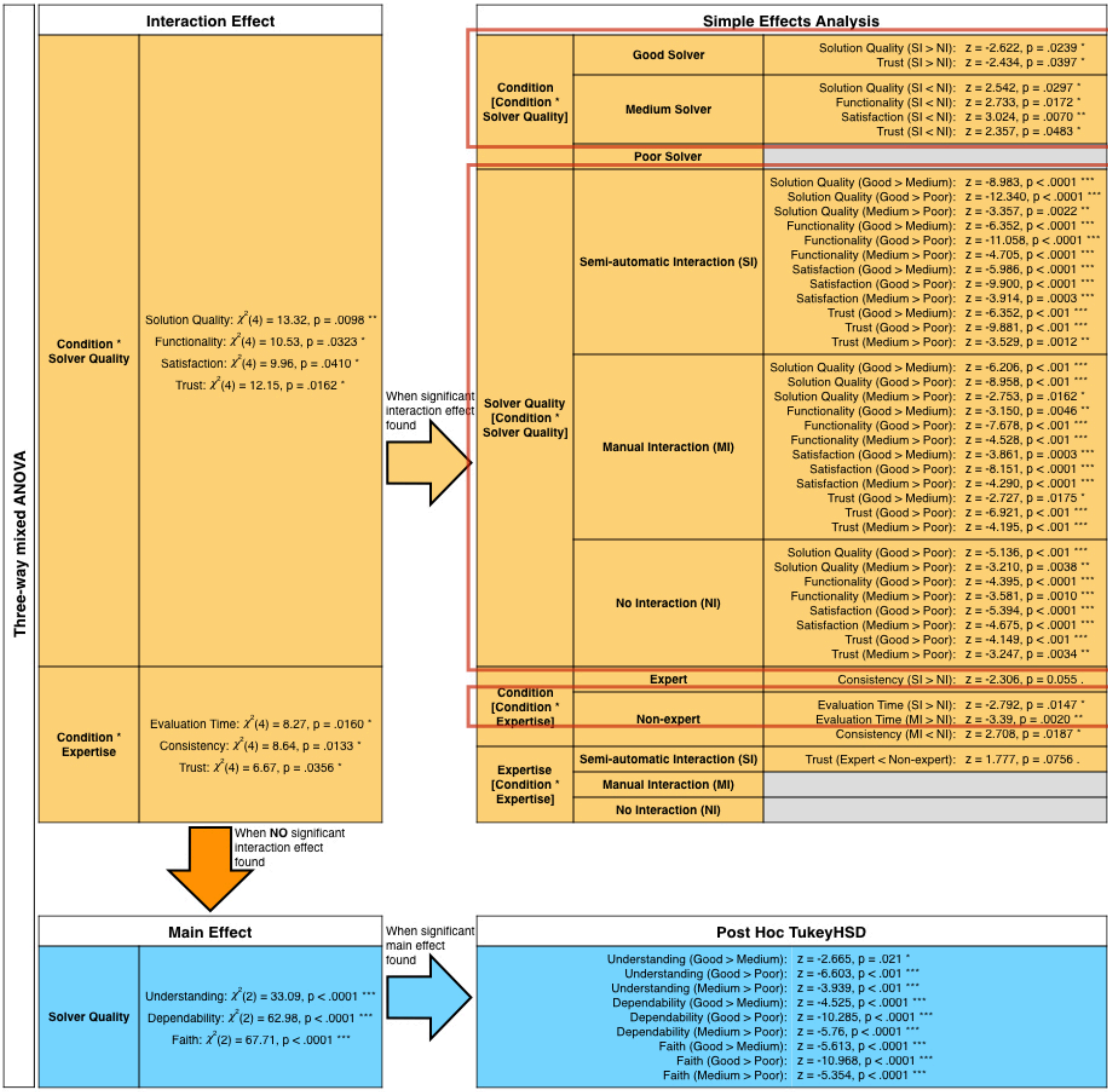

Figure 6.13: Study 2 statistical analysis results. Noteworthy findings are indicated by a red rectangular outline.

Solution Quality

Comparison	Semi-automatic Interaction	Manual Interaction	No Interaction
Medium Solver ≥ Good Solver	0	1 (1 E + 0 NE)	2 (1 E + 1 NE)
Poor Solver ≥ Good Solver	0	0	1 (0 E + 1 NE)
Poor Solver ≥ Medium Solver	1 (0 E + 1 NE)	1 (0 E + 1 NE)	3 (2 E + 1 NE)

Overall Solver Trust

Comparison	Semi-automatic Interaction	Manual Interaction	No Interaction
Medium Solver ≥ Good Solver	0	3 (2 E + 1 NE)	5 (1 E + 4 NE)
Poor Solver ≥ Good Solver	0	2 (2 E + 0 NE)	2 (1 E + 1 NE)
Poor Solver ≥ Medium Solver	3 (2 E + 1 NE)	2 (2 E + 0 NE)	3 (2 E + 1 NE)

Figure 6.14: Study 2: number of participants giving wrong relative ratings to solver pairs among the 3 experimental conditions.

6.3.4 Data 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. 6.10, Fig. 6.11 and Fig. 6.12.

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 [130]. The statistical analysis is shown in Fig. 6.13.

6.3.5 Results and Discussion

We found that non-experts spent more time to evaluate solution quality under both *semi-automatic* and *manual interaction* conditions than the *no interaction* condition.

The most important result of the analysis is that for the good solver *semi-automatic interaction* as compared to *no interaction* led to significantly increased rating of *solution quality* and overall *trust* in the solver while for the medium solver it lead to decreased rating of *solution quality*, *solver functionality*, *satisfaction* and overall *trust*. In other words, *semi-automatic interaction* led to a larger difference in trust between the good and medium solvers. We also saw that *manual interaction* led to a larger difference in solution and solver rating between good and medium solvers as compared to *no interaction*, but less than for *semi-automatic interaction*, though this was not statistically significant. This is also supported by Fig. 6.14 which shows the number of participants who incorrectly rated the *poor* solver as producing better solutions or being more trustworthy than the *medium* or *good* solver or the *medium* solver as better than the *good*.

These results provide strong support for the hypothesis that interaction leads to better calibrated trust in solvers. [3]

We also collected qualitative feedback from the post-questionnaire asking participants about the effect of interaction on trust. The 10 participants from the *no interaction* group are excluded from the analysis because they are not provided with any interactions during the study. For the rest 20 participants from the *semi-automatic* and *manual* conditions, 17 out of 20 (85 %) participants thought manually changing a solution via interaction can increase trust in good optimisation solvers. The other 3 participants were neutral about the change on trust.

Most participants thought their trust increased because they did not find any better solution after they tried. One participant said:

> "If the solver gives a good first impression and I dont 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."

One participant also told us:

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

6.4 Conclusions

In our first experiment, we found that providing feedback about intermediate solution and objective function leads to increase trust in an optimisation solver. In fact, we found that it leads to over-trust. Simply adding visual effects to show intermediate solutions and a decreasing objective function (for minimisation problems) means that many people will think that a poor solver is giving better results than a good solver, leading to greater trust in the poor solver.

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 empirical support for the belief by some optimisation researchers [1] that interaction leads to greater trust.

Simple animation and feedback through the line chart as used in the first experiment led to misplaced trust (their trust in the weaker solver was disproportionately increased relative to the stronger solver). We do not recommend to use this approach to gain users' trust. However, feedback coupled with interaction as used in the second experiment, led to more appropriately increased trust, and is therefore our recommended approach for autonomous applications to engender the right amount of trust.

[3] The better calibrated trust refers to good solvers are not under trusted. Both medium and poor solvers are not over trusted.

Chapter 7

Conclusions

In this thesis, we have presented how to build an effective interface to support human-in-the-loop optimisation. We conclude the thesis in this chapter by first revisit the research questions, followed by a discussion of contributions, and then we present the future direction of this research.

As an important tool for decision support, optimisation has had tremendous success in helping people to solve complex decision problems in a wide range of application areas. However, traditional fully-automatic optimisation has its limitations. There is a growing recognition by the optimisation community that there is a need to directly engage users in the automatic optimisation process. Such a human-in-the-loop approach for optimisation has advantages in improving or even eliminating the limitations from fully-automatic optimisation. However, as a key aspect in human-in-the-loop optimisation, HCI is usually overlooked by optimisation researchers. In this thesis, we focus on the HCI aspect which includes appropriate visual representations and interaction techniques to support human-in-the-loop optimisation. In order to do so, we addressed two main research questions and two supplementary research questions as follows:
RQ 1: *How do we develop effective interfaces to support human-in-the-loop optimisation?*

- RQ 1-1: *What are the high-level goals and processes in human-in-the-loop optimisation?*
- RQ 1-2: *What are effective visualisation and interaction techniques to support human-in-the-loop optimisation?*

RQ 2: *How to engender the right amount of users' trust in optimisation systems?*

7.1 Contributions

7.1.1 Prostate Brachytherapy Case Study

In Chapter 3, we introduced our first case study, prostate brachytherapy, a treatment approach for prostate cancer. In particular, we developed a prototype human-in-the-loop optimisation tool for prostate brachytherapy that goes beyond the current practice in supporting focal therapy, a new treatment approach targeting tumour cells directly rather than simply seeking coverage of the entire prostate gland, also known as the traditional whole gland therapy approach.

This tool was developed with help from our external collaborators: a medical physicist and a bioengineer. It took a number of iterations to produce a satisfactory looking of the interface in cooperating a visualisation to present a single solution. This iterative process took more than a year before the final interface was settled upon.

Unlike typical user studies in human-in-the-loop optimisation, our tool was evaluated by seven radiation oncology professionals using a two-stage semi-structured interview, as discussed in Chapter 4. Our first-stage interview focussed on the traditional whole gland therapy treatment approach with two aims. First, it helped us to gain a better understanding of the clinical treatment planning procedure, which was beneficial in building our theoretical framework described in Section 4.2. Second, it clarified the reasons why clinical professionals do not use and do not like a fully-automatic optimisation system. Based on the valuable feedback from the first-stage interview, we made a number of changes and adjustments to our tool to prepare for the second-stage interview which focussed on the new focal therapy approach. The aim of the second-stage interview was to explore how radiation oncology professionals use such a human-in-the-loop optimisation in doing treatment planning using the unfamiliar focal therapy. The results showed that professionals did build trust in focal therapy after using the tool and importantly they overwhelmingly prefer to use human-in-the-loop optimisation to create focal therapy treatment plans.

7.1.2 Vehicle Routing Problem with Time Windows Case Study

In Chapter 5, we presented the second case study, vehicle routing problem with time windows. We developed another human-in-the-loop optimisation tool for this case study. One aim of the second case study is to use it as a testbed to investigate how to build an effective interface to support human-in-the-loop optimisation. We did a small heuristic evaluation of the tool with four participants described in Section 5.5. We found that the design of the interface had a good reflection of the design guidelines presented in Section 5.1.

7.1.3 The Problem-solving Loop

As mentioned above in Section 7.1.1, we suggested a novel theoretical framework called the problem-solving loop. (recall Section 4.2). The design of the framework was enlightened by the identification of the clinical treatment planning workflow as a finding of the first-stage semi-structured interview described in Section 4.1. The articulation and formulation of the framework drew on ideas from the sense-making loop in visual analytics mentioned in Chapter 1 and in Section 2.3. The problem-solving loop identified important high-level user goals and tasks in human-in-the-loop optimisation. With this, we were able to answer the first supplementary question RQ 1-1: *What are the high-level goals and processes in human-in-the-loop optimisation?*

7.1.4 Nine Design Guidelines

At the beginning of Chapter 5, we presented nine design guidelines for building an effective interface to support human-in-the-loop optimisation. The design guidelines were extracted and generalised based on the problem-solving loop discussed above. However, the guidelines drew a focus on the HCI aspect. It included important visual components and essential interaction styles and operations to be implemented in a human-in-the-loop optimisation interface. We revisited the design of our first prototype human-in-the-loop optimisation tool for the first case study, prostate brachytherapy (recall Section 5.2), which was developed before the design guidelines. We found that overall it has a good reflection of the guidelines. Also in Chapter 5, we presented our design of the interface for the second case study, vehicle routing problem with time windows in Section 5.4. We found the design guidelines are beneficial for the development of the interface design. In Section 5.5, we presented our small case study, a heuristic evaluation of the usability of the interface. It revealed that the design of the interface was effective and it conformed to the guidelines.

With the nine design guidelines, we were able to positively answer the second supplementary research question RQ 1-2: *What are effective visualisation and interaction techniques to support human-in-the-loop optimisation?* With both supplementary research questions RQ 1-1 and RQ 1-2 answered, we solved the first main research question RQ 1: *How do we develop effective interface to support human-in-the-loop optimisation?*

7.1.5 Engenderment of Appropriate Level of Trust

We conducted two controlled experiments to investigate users' trust described in Chapter 6. Our first controlled experiment (recall Section 6.2) focussed on the effect of feedback on trust when users were provided with information about interim solutions and the objective function. We found that simply adding visual effects to show intermediate solutions and a decreasing objective function (for minimisation problems) makes many people think a poor optimisation solver is giving better results than a good solver, leading to a greater trust in the poor solver. However, we do not recommend this approach as it leads to misplaced over-trust. In our second controlled experiment (recall Section 6.3), we aimed to verify the effect of interactions on trust. We found that allowing users to semi-automatically manipulate solution returned by an optimisation system leads to better-calibrated trust. It provided the first empirical support for the belief by some optimisation researchers that interaction can lead to greater trust. With this, we were able to answer the second main research question RQ 2: *How to engender the right amount of users' trust in optimisation systems?*

7.2 Future Directions

Guidelines for Solutions Presentations and Communications.
In this thesis, we presented nine design guidelines to inform the development of human-in-the-loop optimisation interfaces. However, the guidelines were primarily focusing on the optimisation solving process of the optimisation loop, the main part of the problem-solving loop. The optimisation loop also included the communication and solution presentation aspects in human-in-the-loop optimisation, which is another aspect other than HCI also receives little attention by optimisation researchers. It is worthwhile to develop guidelines to support better solution presentations and to develop an effective way to communicate the solution to managers, policy makers and other stakeholders.
Guidelines for the Model-defining Loop.
It is also possible to identify guidelines for the model-defining loop, which will be primarily used by optimisation experts to make the mathematical model more flexible and more interactive so that even problem domain experts can refine the model. For example, they can do this by adding new constraints and exploring the search space without the help from optimisation experts.
Enabling Collaborative Solution Annotation and Problem-solving.
It is necessary to support users annotating solutions in human-in-the-loop optimisation. What is even better and powerful is to enable collaborative annotation of a solution with different users. This is an approach to encourage and to enable a better way to communicate solutions.

Similar to the collaborative annotating interaction idea, it might be useful to even support collaborative problem-solving in human-in-the-loop optimisation. For example, multiple users can simultaneously manipulate different parts of a solution. With the support of versatile interactions discussed above, this can become a new way to solve a complex optimisation problem with many users involved at the same time. However, the effectiveness of such a collaborative problem-solving approach in human-in-the-loop optimisation needs to be tested and verified in future research.

More Investigation on Trust.

In this thesis, we investigated the effect on users' trust by providing feedback about interim solutions and the objective function and allowing users to semi-automatically manipulate solutions. However, trust is very abstract and complex. There exist many different factors affecting users' trust. Extending from our investigation on feedback, evaluating the effect of system transparency on trust by providing interactive visualisation to support the exploration of the search space to further improve the transparency of an optimisation system is a possible future research direction to better understand users' trust on optimisation systems.

Implementation of generating different solutions.

In Chapter 5, ***Guideline 7*** recommends generating different solutions. However, the implementation can be extremely difficult. As discussed in Section 5.1, one possible approach is to define a solution diversity matrix to measure the differences between solutions. This is an interesting direction for future experimentation.

7.3 Final Words

This thesis presented how to build an effective interactive interface to support human-in-the-loop optimisation with two case studies: prostate brachytherapy and vehicle routing problem with time windows. We presented a theoretical framework, the problem-solving loop, which identified high-level user goals and tasks in human-in-the-loop optimisation. We also presented nine design guidelines for the design of a human-in-the-loop optimisation interface. We investigated the effect of feedback and interaction on users' trust. In doing so, we answered two search questions.

We hope the problem-solving loop framework and the design guidelines will be the basis for the development of human-in-the-loop optimisation interfaces to be used in many other application areas to achieve better problem-solving and more sense-making presentation and communication of solutions.

Acronyms

ACO Ant Colony Optimisation.

ANOVA Analysis of Variance.

CVRTW Capacitated Vehicle Routing with Time Windows.

DHS Department of Homeland Security.

EBRT External Beam Radiation Therapy.

EDEN Exploratory Data Analysis Environment.

GA Genetic Algorithm.

HCI Human Computer Interaction.

HDR High Dose Rate.

HSD Honestly Significant Difference.

HuGSS Human-guided Simple Search.

ID Identification.

IEC Interactive Evolutionary Computation.

IT Information Technology.

LDR Low Dose Rate.

LNS Large Neighbourhood Search.

LP Linear Programming.

MI Manual Interaction.

MP Medical Physicist.

MUHREC Monash University Human Research Ethics Committee.

NI No Interaction.

NLP Nonlinear Programming.

NVAC National Visualisation and Analytics Centre.

OAR Organs at Risk.

OSM Open Street Map.

OSRM Open Source Routing Machine.

PC Personal Computer.

PICO Physical Intervention in Computational Optimisation.

PTV Planning Target Volume.

RO Radiation Oncologist.

RT Radiation Therapist.

RV Rectum Volume.

SI Semi-automatic Interaction.

TCD Tumour Cell Density.

TCP Tumour Control Probability.

TP Treatment Planner.

TSP Travelling Salesman Problem.

UV Urethra Volume.

VRP Vehicle Routing Problem.

VRPTW Vehicle Routing Problem with Time Windows.

References

[1] D. Meignan, S. Knust, J.-M. Frayret, G. Pesant, and N. Gaud, “A review and taxonomy of interactive optimization methods in operations research,” *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 5, no. 3, p. 17, 2015.

[2] Z. Huang, L. Zhang, S. Cheng, J. Zhang, and X. Xia, “Back-analysis and parameter identification for deep excavation based on pareto multiobjective optimization,” *Journal of Aerospace Engineering*, vol. 28, no. 6, p. A4014007, 2014.

[3] D. Anderson, E. Anderson, N. Lesh, J. Marks, B. Mirtich, D. Ratajczak, and K. Ryall, “Human-guided simple search,” in *AAAI/IAAI*, 2000, pp. 209–216.

[4] J. Patten and H. Ishii, “Mechanical constraints as computational constraints in tabletop tangible interfaces,” in *Proceedings of the SIGCHI conference on Human factors in computing systems.* ACM, 2007, pp. 809–818.

[5] A. Uğur and D. Aydin, “An interactive simulation and analysis software for solving TSP using Ant Colony Optimization algorithms,” *Advances in Engineering software*, vol. 40, no. 5, pp. 341–349, 2009.

[6] E. Brochu, T. Brochu, and N. de Freitas, “A Bayesian interactive optimization approach to procedural animation design,” in *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation.* Eurographics Association, 2010, pp. 103–112.

[7] B. Belin, M. Christie, and C. Truchet, “Interactive design of sustainable cities with a distributed local search solver,” in *Integration of AI and OR Techniques in Constraint Programming.* Springer, 2014, pp. 104–119.

[8] C. A. Steed, D. M. Ricciuto, G. Shipman, B. Smith, P. E. Thornton, D. Wang, X. Shi, and D. N. Williams, “Big data visual analytics for exploratory earth system simulation analysis,” *Computers & Geosciences*, vol. 61, pp. 71–82, 2013.

[9] D. A. Keim, J. Kohlhammer, G. Ellis, and F. Mansmann, *Mastering the Information Age: Solving Problems with Visual Analytics.* Eurographics Association, 2010.

[10] K. A. Hoff and M. Bashir, “Trust in automation: Integrating empirical evidence on factors that influence trust,” *Human Factors*, vol. 57, no. 3, pp. 407–434, 2015.

[11] J. D. Lee and K. A. See, “Trust in automation: Designing for appropriate reliance,” *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 46, no. 1, pp. 50–80, 2004.

[12] A. Richards and J. P. How, “Aircraft trajectory planning with collision avoidance using mixed integer linear programming,” in *American Control Conference, 2002. Proceedings of the 2002*, vol. 3. IEEE, 2002, pp. 1936–1941.

[13] L. Wang, P. K. Basu, and J. P. Leiva, "Automobile body reinforcement by finite element optimization," *Finite Elements in Analysis and Design*, vol. 40, no. 8, pp. 879–893, 2004.

[14] J. Madar, J. Abonyi, and F. Szeifert, "Interactive particle swarm optimization," in *Intelligent Systems Design and Applications, 2005. ISDA'05. Proceedings. 5th International Conference on.* IEEE, 2005, pp. 314–319.

[15] H. Takagi, "Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation," *Proceedings of the IEEE*, vol. 89, no. 9, pp. 1275–1296, 2001.

[16] H. Madsen, "Automatic calibration of a conceptual rainfall–runoff model using multiple objectives," *Journal of hydrology*, vol. 235, no. 3, pp. 276–288, 2000.

[17] G. W. Klau, N. Lesh, J. Marks, and M. Mitzenmacher, "Human-guided search," *Journal of Heuristics*, vol. 16, no. 3, pp. 289–310, 2010.

[18] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," in *ACM transactions on graphics (TOG)*, vol. 23, no. 3. ACM, 2004, pp. 309–314.

[19] A. Kapoor, B. Lee, D. Tan, and E. Horvitz, "Interactive optimization for steering machine classification," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* ACM, 2010, pp. 1343–1352.

[20] J. J. Thomas, *Illuminating the path:[the research and development agenda for visual analytics].* IEEE Computer Society, 2005.

[21] K. E. Schaefer, J. Y. Chen, J. L. Szalma, and P. A. Hancock, "A meta-analysis of factors influencing the development of trust in automation: Implications for understanding autonomy in future systems," *Human factors*, vol. 58, no. 3, pp. 377–400, 2016.

[22] R. H. von Alan, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS quarterly*, vol. 28, no. 1, pp. 75–105, 2004.

[23] J. Nocedal and S. J. Wright, *Numerical Optimization.* Springer, New York, 1999.

[24] R. Horst and P. M. Pardalos, *Handbook of global optimization.* Springer Science & Business Media, 2013, vol. 2.

[25] S. Boyd and L. Vandenberghe, *Convex optimization.* Cambridge university press, 2004.

[26] T. Weise, "Global optimization algorithms-theory and application," *Self-published*, vol. 2, 2009.

[27] K. Miettinen, *Nonlinear multiobjective optimization.* Springer Science & Business Media, 2012, vol. 12.

[28] D. Fudenberg and J. Tirole, "Game theory, 1991," *Cambridge, Massachusetts*, vol. 393, no. 12, p. 80, 1991.

[29] B. Roy, "Paradigms and challenges," in *Multiple criteria decision analysis: state of the art surveys.* Springer, 2005, pp. 3–24.

[30] M. L. Cummings, J. J. Marquez, and N. Roy, "Human-automated path planning optimization and decision support," *International Journal of Human-Computer Studies*, vol. 70, no. 2, pp. 116–128, 2012.

[31] F. D. Davis and J. E. Kottemann, "User perceptions of decision support effectiveness: Two production planning experiments," *Decision Sciences*, vol. 25, no. 1, pp. 57–76, 1994.

[32] J. Wallenius, "Comparative evaluation of some interactive approaches to multicriterion optimization," *Management science*, vol. 21, no. 12, pp. 1387–1396, 1975.

[33] S. Zionts and J. Wallenius, "An interactive programming method for solving the multiple criteria problem," *Management science*, vol. 22, no. 6, pp. 652–663, 1976.

[34] F. Glover, C. McMillan, and B. Novick, "Interactive decision software and computer graphics for architectural and space planning," *Annals of operations research*, vol. 5, no. 3, pp. 557–573, 1985.

[35] C. V. Jones and W. L. Maxwell, "A system for manufacturing scheduling with interactive computer graphics," *IIE transactions*, vol. 18, no. 3, pp. 298–303, 1986.

[36] M. Fisher, "Interactive optimization," *Annals of Operations Research*, vol. 5, no. 1-4, pp. 541–556, 1986.

[37] H. W. Hamacher and K.-H. Kuefer, "Inverse radiation therapy planning: A multiple objective optimisation approach," in *Monitoring, Evaluating, Planning Health Services*. World Scientific, 1999, pp. 177–189.

[38] C. Thieke, K.-H. Küfer, M. Monz, A. Scherrer, F. Alonso, U. Oelfke, P. E. Huber, J. Debus, and T. Bortfeld, "A new concept for interactive radiotherapy planning with multicriteria optimization: First clinical evaluation," *Radiotherapy and Oncology*, vol. 85, no. 2, pp. 292–298, 2007.

[39] D. L. Craft, T. S. Hong, H. A. Shih, and T. R. Bortfeld, "Improved planning time and plan quality through multicriteria optimization for intensity-modulated radiotherapy," *International Journal of Radiation Oncology* Biology* Physics*, vol. 82, no. 1, pp. e83–e90, 2012.

[40] G. Bailly, A. Oulasvirta, T. Kötzing, and S. Hoppe, "Menuoptimizer: Interactive optimization of menu systems," in *Proceedings of the 26th annual ACM symposium on User interface software and technology*. ACM, 2013, pp. 331–342.

[41] T. Dwyer, K. Marriott, and M. Wybrow, "Dunnart: A constraint-based network diagram authoring tool," in *Graph Drawing*. Springer, 2009, pp. 420–431.

[42] S. Ben Ismail, F. Legras, and G. Coppin, "A new interactive evolutionary algorithm for the vehicle routing problem," in *Proceedings of the 14th annual conference companion on Genetic and evolutionary computation*. ACM, 2012, pp. 661–668.

[43] M. Ai-Chang, J. Bresina, L. Charest, A. Chase, J. C.-J. Hsu, A. Jonsson, B. Kanefsky, P. Morris, K. Rajan, J. Yglesias *et al.*, "Mapgen: Mixed-initiative planning and scheduling for the Mars Exploration Rover mission," *Intelligent Systems, IEEE*, vol. 19, no. 1, pp. 8–12, 2004.

[44] K. Miettinen, F. Ruiz, and A. P. Wierzbicki, "Introduction to multiobjective optimization: interactive approaches," in *Multiobjective Optimization*. Springer, 2008, pp. 27–57.

[45] K. Miettinen and M. M. Mäkelä, "Synchronous approach in interactive multiobjective optimization," *European Journal of Operational Research*, vol. 170, no. 3, pp. 909–922, 2006.

[46] J. Allen and G. Ferguson, "Human-machine collaborative planning," in *Proceedings of the Third International NASA Workshop on Planning and Scheduling for Space*, 2002, pp. 27–29.

[47] A. Teacher and D. Griffiths, "Hapstar: automated haplotype network layout and visualization," *Molecular Ecology Resources*, vol. 11, no. 1, pp. 151–153, 2011.

[48] S.-B. Cho, "Towards creative evolutionary systems with interactive genetic algorithm," *Applied Intelligence*, vol. 16, no. 2, pp. 129–138, 2002.

[49] H. A. do Nascimento and P. Eades, "User hints: a framework for interactive optimization," *Future Generation Computer Systems*, vol. 21, no. 7, pp. 1177–1191, 2005.

[50] C. Zeidler, C. Lutteroth, W. Sturzlinger, and G. Weber, "The Auckland Layout Editor: An improved GUI layout specification process," in *Proceedings of the 26th annual ACM symposium on User interface software and technology.* ACM, 2013, pp. 343–352.

[51] K. A. Butler, J. Zhang, C. Esposito, A. Bahrami, R. Hebron, and D. Kieras, "Work-centered design: A case study of a mixed-initiative scheduler," in *Proceedings of the SIGCHI conference on Human factors in computing systems.* ACM, 2007, pp. 747–756.

[52] K. Fagerholt and H. Lindstad, "Turborouter: An interactive optimisation-based decision support system for ship routing and scheduling," *Maritime Economics & Logistics*, vol. 9, no. 3, pp. 214–233, 2007.

[53] A. M. Brintrup, J. Ramsden, and A. Tiwari, "An interactive genetic algorithm-based framework for handling qualitative criteria in design optimization," *Computers in Industry*, vol. 58, no. 3, pp. 279–291, 2007.

[54] J. Bénabes, F. Bennis, E. Poirson, and Y. Ravaut, "Interactive optimization strategies for layout problems," *International Journal on Interactive Design and Manufacturing (IJIDeM)*, vol. 4, no. 3, pp. 181–190, 2010.

[55] K. Savla, T. Temple, and E. Frazzoli, "Human-in-the-loop vehicle routing policies for dynamic environments," in *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on.* IEEE, 2008, pp. 1145–1150.

[56] S. D. Scott, N. Lesh, and G. W. Klau, "Investigating human-computer optimization," in *Proceedings of the SIGCHI conference on Human factors in computing systems.* ACM, 2002, pp. 155–162.

[57] M. Shibuya, H. Kita, and S. Kobayashi, "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, 1999, pp. 646–651.

[58] J. L. Bresina, A. K. Jónsson, P. H. Morris, and K. Rajan, "Activity planning for the Mars Exploration Rovers." in *ICAPS*, 2005, pp. 40–49.

[59] W. van Wezel and R. Jorna, "Cognition, tasks and planning: Supporting the planning of shunting operations at the Netherlands Railways," *Cognition, Technology & Work*, vol. 11, no. 2, pp. 165–176, 2009.

[60] Y. Boykov and M.-P. Jolly, "Interactive organ segmentation using graph cuts," in *Medical Image Computing and Computer-Assisted Intervention*. Springer, 2000, pp. 276–286.

[61] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen, "Interactive digital photomontage," in *ACM Transactions on Graphics (TOG)*, vol. 23, no. 3. ACM, 2004, pp. 294–302.

[62] M. Andersen, S. Kleindienst, L. Yi, J. Lee, M. Bodart, and B. Cutler, "An intuitive daylighting performance analysis and optimization approach," *Building Research & Information*, vol. 36, no. 6, pp. 593–607, 2008.

[63] J. B. Kollat and P. Reed, "A framework for Visually Interactive Decision-making and Design using Evolutionary Multi-objective Optimization (VIDEO)," *Environmental Modelling & Software*, vol. 22, no. 12, pp. 1691–1704, 2007.

[64] C. V. Jones, "Visualization and optimization," *ORSA Journal on Computing*, vol. 6, no. 3, pp. 221–257, 1994.

[65] K. Miettinen, "Survey of methods to visualize alternatives in multiple criteria decision making problems," *OR spectrum*, vol. 36, no. 1, pp. 3–37, 2014.

[66] S. Goodwin, C. Mears, T. Dwyer, M. G. de la Banda, G. Tack, and M. Wallace, "What do constraint programming users want to see? Exploring the role of visualisation in profiling of models and search," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 281–290, 2017.

[67] T. Torsney-Weir, A. Saad, T. Moller, H.-C. Hege, B. Weber, J.-M. Verbavatz, and S. Bergner, "Tuner: Principled parameter finding for image segmentation algorithms using visual response surface exploration," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 1892–1901, 2011.

[68] D. Coffey, C.-L. Lin, A. G. Erdman, and D. F. Keefe, "Design by dragging: An interface for creative forward and inverse design with simulation ensembles," *IEEE transactions on visualization and computer graphics*, vol. 19, no. 12, pp. 2783–2791, 2013.

[69] S. Bruckner and T. Moller, "Result-driven exploration of simulation parameter spaces for visual effects design," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 1468–1476, 2010.

[70] A. Diehl, L. Pelorosso, C. Delrieux, C. Saulo, J. Ruiz, M. E. Gröller, and S. Bruckner, "Visual analysis of spatio-temporal data: Applications in weather forecasting," in *Computer Graphics Forum*, vol. 34, no. 3. Wiley Online Library, 2015, pp. 381–390.

[71] W. Berger, H. Piringer, P. Filzmoser, and E. Gröller, "Uncertainty-aware exploration of continuous parameter spaces using multivariate prediction," in *Computer Graphics Forum*, vol. 30, no. 3. Wiley Online Library, 2011, pp. 911–920.

[72] M. Sedlmair, C. Heinzl, S. Bruckner, H. Piringer, and T. Möller, "Visual parameter space analysis: A conceptual framework," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 2161–2170, 2014.

[73] D. Cosley, S. K. Lam, I. Albert, J. A. Konstan, and J. Riedl, “Is seeing believing? How recommender system interfaces affect users’ opinions,” in *Proceedings of the SIGCHI conference on Human factors in computing systems.* ACM, 2003, pp. 585–592.

[74] J. L. Harman, J. O’Donovan, T. Abdelzaher, and C. Gonzalez, “Dynamics of human trust in recommender systems,” in *Proceedings of the 8th ACM Conference on Recommender systems.* ACM, 2014, pp. 305–308.

[75] A. Glass, D. L. McGuinness, and M. Wolverton, “Toward establishing trust in adaptive agents,” in *Proceedings of the 13th international conference on Intelligent user interfaces.* ACM, 2008, pp. 227–236.

[76] W. Pieters, “Explanation and trust: What to tell the user in security and AI?” *Ethics and information technology*, vol. 13, no. 1, pp. 53–64, 2011.

[77] M. T. Ribeiro, S. Singh, and C. Guestrin, ““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, 2016, pp. 1135–1144.

[78] J. W. Tukey, *Exploratory data analysis.* Reading, Mass., 1977, vol. 2.

[79] B. Stewart, C. P. Wild *et al.*, “World cancer report 2014,” *Self*, 2018.

[80] B. R. Pieters, D. Z. de Back, C. C. Koning, and A. H. Zwinderman, “Comparison of three radiotherapy modalities on biochemical control and overall survival for the treatment of prostate cancer: a systematic review,” *Radiotherapy and Oncology*, vol. 93, no. 2, pp. 168–173, 2009.

[81] A. U. Kishan, R. R. Cook, J. P. Ciezki, A. E. Ross, M. M. Pomerantz, P. L. Nguyen, T. Shaikh, P. T. Tran, K. A. Sandler, R. G. Stock *et al.*, “Radical prostatectomy, external beam radiotherapy, or external beam radiotherapy with brachytherapy boost and disease progression and mortality in patients with gleason score 9-10 prostate cancer,” *Jama*, vol. 319, no. 9, pp. 896–905, 2018.

[82] S. Langley, H. U. Ahmed, B. Al-Qaisieh, D. Bostwick, L. Dickinson, F. G. Veiga, P. Grimm, S. Machtens, F. Guedea, and M. Emberton, “Report of a consensus meeting on focal low dose rate brachytherapy for prostate cancer,” *BJU international*, vol. 109, no. s1, pp. 7–16, 2012.

[83] A. Haworth, S. Williams, H. Reynolds, D. Waterhouse, G. M. Duchesne, J. Bucci, D. Joseph, S. Bydder, and M. Ebert, “Validation of a radiobiological model for low-dose-rate prostate boost focal therapy treatment planning,” *Brachytherapy*, vol. 12, no. 6, pp. 628–636, 2013.

[84] A. Haworth, M. Ebert, D. Waterhouse, D. Joseph, and G. Duchesne, “Prostate implant evaluation using tumour control probability — The effect of input parameters,” *Physics in medicine and biology*, vol. 49, no. 16, p. 3649, 2004.

[85] A. Haworth, C. Mears, J. M. Betts, H. M. Reynolds, G. Tack, K. Leo, S. Williams, and M. A. Ebert, “A radiobiology-based inverse treatment planning method for optimisation of permanent I-125 prostate implants in focal brachytherapy,” *Physics in medicine and biology*, vol. 61, no. 1, p. 430, 2015.

[86] J. Nielsen and R. Molich, "Heuristic evaluation of user interfaces," in *Proceedings of the SIGCHI conference on Human factors in computing systems.* ACM, 1990, pp. 249–256.

[87] P. Deransart and M. V. Hermenegildo, *Analysis and visualization tools for constraint programming: constraint debugging.* Springer Science & Business Media, 2000, no. 1870.

[88] L. De Raedt, A. Passerini, and S. Teso, "Learning constraints from examples," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[89] R. Arcangioli, C. Bessière, and N. Lazaar, "Multiple Constraint Aquisition," in *IJCAI: International Joint Conference on Artificial Intelligence*, New York City, United States, Jul. 2016, pp. 698–704. [Online]. Available: https://hal-lirmm.ccsd.cnrs.fr/lirmm-01374712

[90] N. Beldiceanu and H. Simonis, "A constraint seeker: Finding and ranking global constraints from examples," in *International Conference on Principles and Practice of Constraint Programming.* Springer, 2011, pp. 12–26.

[91] H. C. Purchase, E. Hoggan, and C. Görg, "How important is the mental map?– an empirical investigation of a dynamic graph layout algorithm," in *International Symposium on Graph Drawing.* Springer, 2006, pp. 184–195.

[92] H. C. Purchase and A. Samra, "Extremes are better: Investigating mental map preservation in dynamic graphs," in *International Conference on Theory and Application of Diagrams.* Springer, 2008, pp. 60–73.

[93] K.-K. Muniswamy-Reddy, D. A. Holland, U. Braun, and M. I. Seltzer, "Provenance-aware storage systems." in *USENIX Annual Technical Conference, General Track*, 2006, pp. 43–56.

[94] J. Nielsen, "10 usability heuristics for user interface design," *Nielsen Norman Group*, vol. 1, no. 1, 1995.

[95] T. Petit and A. C. Trapp, "Finding diverse solutions of high quality to constraint optimization problems," in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[96] E. Hebrard, B. O'Sullivan, and T. Walsh, "Distance constraints in constraint satisfaction." in *IJCAI*, vol. 2007, 2007, pp. 106–111.

[97] E. Hebrard, B. Hnich, B. O'Sullivan, and T. Walsh, "Finding diverse and similar solutions in constraint programming," in *AAAI*, vol. 5, 2005, pp. 372–377.

[98] G. B. Dantzig and J. H. Ramser, "The truck dispatching problem," *Management Science*, vol. 6, no. 1, pp. 80–91, 1959.

[99] E. Tufte and P. Graves-Morris, "The visual display of quantitative information.; 1983," 2014.

[100] N. Nethercote, P. J. Stuckey, R. Becket, S. Brand, G. J. Duck, and G. Tack, "Minizinc: Towards a standard cp modelling language," in *Principles and Practice of Constraint Programming – CP 2007*, C. Bessière, Ed. Springer Berlin Heidelberg, 2007, pp. 529–543.

[101] Gecode Team, "Gecode: A generic constraint development environment," 2016. [Online]. Available: http://www.gecode.org

[102] M. Palpant, C. Artigues, and P. Michelon, "Lssper: Solving the resource-constrained project scheduling problem with large neighbourhood search," *Annals of Operations Research*, vol. 131, no. 1-4, pp. 237–257, 2004.

[103] G. Laporte, R. Musmanno, and F. Vocaturo, "An adaptive large neighbourhood search heuristic for the capacitated arc-routing problem with stochastic demands," *Transportation Science*, vol. 44, no. 1, pp. 125–135, 2010.

[104] J. E. Korsvik, K. Fagerholt, and G. Laporte, "A large neighbourhood search heuristic for ship routing and scheduling with split loads," *Computers & Operations Research*, vol. 38, no. 2, pp. 474–483, 2011.

[105] M. S. Cohen, R. Parasuraman, and J. T. Freeman, "Trust in decision aids: A model and its training implications," in *in Proc. Command and Control Research and Technology Symposium*. Citeseer, 1998.

[106] N. R. Bailey and M. W. Scerbo, "Automation-induced complacency for monitoring highly reliable systems: the role of task complexity, system experience, and operator trust," *Theoretical Issues in Ergonomics Science*, vol. 8, no. 4, pp. 321–348, 2007.

[107] E. de Visser and R. Parasuraman, "Adaptive aiding of human-robot teaming: Effects of imperfect automation on performance, trust, and workload," *Journal of Cognitive Engineering and Decision Making*, vol. 5, no. 2, pp. 209–231, 2011.

[108] P. Briggs, B. Burford, and C. Dracup, "Modelling self-confidence in users of a computer-based system showing unrepresentative design," *International Journal of Human-Computer Studies*, vol. 49, no. 5, pp. 717–742, 1998.

[109] B. Fogg and H. Tseng, "The elements of computer credibility," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1999, pp. 80–87.

[110] J. Kim and J. Y. Moon, "Designing towards emotional usability in customer interfaces?trustworthiness of cyber-banking system interfaces," *Interacting with Computers*, vol. 10, no. 1, pp. 1–29, 1998.

[111] Y.-M. Li and Y.-S. Yeh, "Increasing trust in mobile commerce through design aesthetics," *Computers in Human Behavior*, vol. 26, no. 4, pp. 673–684, 2010.

[112] D. Gefen, E. Karahanna, and D. W. Straub, "Trust and TAM in online shopping: An integrated model," *MIS quarterly*, vol. 27, no. 1, pp. 51–90, 2003.

[113] F. M. Verberne, J. Ham, and C. J. Midden, "Trust in smart systems: Sharing driving goals and giving information to increase trustworthiness and acceptability of smart systems in cars," *Human Factors*, vol. 54, no. 5, pp. 799–810, 2012.

[114] J. Lee and N. Moray, "Trust, control strategies and allocation of function in human-machine systems," *Ergonomics*, vol. 35, no. 10, pp. 1243–1270, 1992.

[115] B. M. Muir and N. Moray, "Trust in automation. part ii. experimental studies of trust and human intervention in a process control simulation," *Ergonomics*, vol. 39, no. 3, pp. 429–460, 1996.

[116] R. Parasuraman and V. Riley, "Humans and automation: Use, misuse, disuse, abuse," *Human factors*, vol. 39, no. 2, pp. 230–253, 1997.

[117] N. Moray, T. Inagaki, and M. Itoh, "Adaptive automation, trust, and self-confidence in fault management of time-critical tasks." *Journal of Experimental Psychology: Applied*, vol. 6, no. 1, p. 44, 2000.

[118] C. L. Corritore, B. Kracher, and S. Wiedenbeck, "On-line trust: concepts, evolving themes, a model," *International journal of human-computer studies*, vol. 58, no. 6, pp. 737–758, 2003.

[119] M. K. Lee and E. Turban, "A trust model for consumer internet shopping," *International Journal of Electronic Commerce*, vol. 6, no. 1, pp. 75–91, 2001.

[120] R. R. Hoffman, J. D. Lee, D. D. Woods, N. Shadbolt, J. Miller, and J. M. Bradshaw, "The dynamics of trust in cyberdomains," *IEEE Intelligent Systems*, no. 6, pp. 5–11, 2009.

[121] M. Hengstler, E. Enkel, and S. Duelli, "Applied artificial intelligence and trust?the case of autonomous vehicles and medical assistance devices," *Technological Forecasting and Social Change*, vol. 105, pp. 105–120, 2016.

[122] K. E. Schaefer, D. R. Billings, J. L. Szalma, J. K. Adams, T. L. Sanders, J. Y. Chen, and P. A. Hancock, "A meta-analysis of factors influencing the development of trust in automation: Implications for human-robot interaction," ARMY RESEARCH LAB ABERDEEN PROVING GROUND MD HUMAN RESEARCH AND ENGINEERING DIRECTORATE, Tech. Rep., 2014.

[123] B. Kim, "Interactive and interpretable machine learning models for human machine collaboration," Ph.D. dissertation, Massachusetts Institute of Technology, 2015.

[124] J. Liu, T. Dwyer, K. Marriott, J. Millar, and A. Haworth, "Understanding the relationship between interactive optimisation and visual analytics in the context of prostate brachytherapy," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 319–329, 2018.

[125] Z. Qu and J. Hullman, "Keeping multiple views consistent: Constraints, validations, and exceptions in visualization authoring," *IEEE transactions on visualization and computer graphics*, vol. 24, no. 1, pp. 468–477, 2018.

[126] D. Artz and Y. Gil, "A survey of trust in computer science and the semantic web," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 5, no. 2, pp. 58–71, 2007.

[127] J.-Y. Jian, A. M. Bisantz, and C. G. Drury, "Foundations for an empirically determined scale of trust in automated systems," *International Journal of Cognitive Ergonomics*, vol. 4, no. 1, pp. 53–71, 2000.

[128] B. M. Muir, "Trust between humans and machines, and the design of decision aids," *International Journal of Man-Machine Studies*, vol. 27, no. 5-6, pp. 527–539, 1987.

[129] T. Christophersen and U. Konradt, "Reliability, validity, and sensitivity of a single-item measure of online store usability," *International Journal of Human-Computer Studies*, vol. 69, no. 4, pp. 269 – 280, 2011.

[130] A. Field, J. Miles, and Z. Field, *Discovering Statistics using R*. Sage publications, 2012.