Investigating Human-Computer Optimization

Stacey D. Scott[1], **Neal Lesh, Gunnar W. Klau**[2]

Mitsubishi Electric Research Laboratories
201 Broadway Street
Cambridge, MA 02139, USA
+1 617 621 7583
lesh@merl.com

ABSTRACT

Scheduling, routing, and layout tasks are examples of hard optimization problems with broad application in industry. Past research in this area has focused on algorithmic issues. However, this approach neglects many important human-computer interaction issues that must be addressed to provide people with practical solutions to optimization problems. Automatic methods do not leverage human expertise and can only find solutions that are optimal with regard to an invariably over-simplified problem description. Furthermore, users must understand the generated solutions in order to implement, justify, or modify them. Interactive optimization helps address these issues but has not previously been studied in detail. This paper describes experiments on an interactive optimization system that explore the most appropriate way to combine the respective strengths of people and computers. Our results show that users can successfully identify promising areas of the search space as well as manage the amount of computational effort expended on different subproblems.

Keywords

Semi-automatic optimization, interactive optimization, human-in-the-loop, user study, training systems, tabletop interfaces.

INTRODUCTION

Research on designing systems to solve optimization problems, such as routing, layout, and scheduling problems, focuses on developing automatic algorithms to search the exponentially large space of possible solutions more efficiently. Typically, the user's role in these systems is to specify the problem, including constraints on predefined criteria for evaluating candidate solutions, and then to initiate a computer search to find an optimal solution.

Such research, performed mainly by the operations-research community, neglects aspects of the optimization task that are essential for obtaining usable solutions. System users must understand and trust the generated solutions to make effective use of them. Furthermore, it is often impossible to specify in advance all of the appropriate constraints and selection criteria for all possible scenarios of a problem.

Consider, for example, someone producing a monthly work schedule. She must understand the solution to convey it to the affected employees. Moreover, she must understand how to make modifications as new needs arise. Furthermore, she probably cannot transfer all of her experience in evaluating candidate solutions to the computer. Thus, automatic methods can produce schedules that conflict with the accumulated wisdom of the people who implement them.

One way these issues have been addressed is by building systems that involve people in the optimization process, typically allowing them to guide or steer an optimization algorithm. Users are more likely to understand a solution that they helped create than one that is simply presented to them. Furthermore, in an interactive system, as a user better understands the available choices he can modify the solution selection criteria as well as steer the computer toward solutions that are most appropriate in practice.

By including humans "in-the-loop" during optimization, we can leverage their problem-solving expertise and their skills in areas where they currently outperform computers, such as visual perception and strategic thinking. In essence, combining the human's superior intelligence with the computer's superior computational speed can result in better solutions than either could produce alone [1, 3].

The goal of our research is to investigate human-in-the-loop optimization in more detail. Below, we discuss past approaches to interactive optimization. Then, we present a study that closely examines expert users' performance on three subtasks performed in an interactive optimization system. To our knowledge, this is the first study that evaluates several individual components of human-in-the-loop optimization.

CHI 2002, April 20-25, 2002, Minneapolis, Minnesota, USA.
Copyright 2001 ACM 1-58113-453-3/02/0004…$5.00.

[1] Present address: School of Computing Science, Simon Fraser University, Burnaby, BC, V5A 1S6, Canada, sdscott@sfu.ca.

[2] Present address: Institute of Computer Graphics and Algorithms, Vienna University of Technology, Favoritenstr. 9-11, A-1040 Vienna, Austria, gunnar@ads.tuwien.ac.at.

After our first experiment, we developed an enhanced version of our experimental system intended to train people on one of the subtasks. This was motivated by the observation that users' performance improved during the experiment and also by valuable user comments for improving the system feedback to be more clear and educational. Below, we describe a pilot study of this new system on novice users.

HUMAN-IN-THE-LOOP OPTIMIZATION SYSTEMS

Cooperative systems that leverage the strengths of both humans and computers have been shown to be effective at producing valuable optimization solutions [1, 17]. These interactive systems must somehow distribute the work involved in the optimization task among the human and computer participants. Existing systems have implemented this division of labour in a variety of ways.

In some interactive systems, the users can only indirectly effect the solutions to the current problem. In *interactive evolution*, an approach primarily applied to design problems, the computer generates solutions and the role of the user is to select which solutions will be used to generate novel solutions in the next iteration [9, 13, 16]. Smith *et al.*'s [14] interactive system for solving large scale planning and scheduling problems presents the users with a variety of solutions that optimize different criteria.

Colgan *et al*. [4] present a system which allows users to interactively control the parameters which are used to evaluate candidate solutions. Users are provided with a visualization of the optimization process so that they can better understand the trade-offs in the design space. Several constraint-based systems have been developed for drawing applications [8, 10, 11]. Typically, the user imposes geometric or topological constraints on an emerging drawing such that subsequent manipulations are constrained to useful areas of the design space.

Other approaches allow the users to manually modify computer-generated solutions, with little or no restrictions, and then to invoke various computer analyses on the updated solution. An early vehicle-routing system allows users to initiate route analyses and map redrawings, and to request suggestions for improvements after making schedule refinements to the initial solution [17]. Chien *et al.*'s [2] space-shuttle operations scheduling system allows users to invoke a repair algorithm on their manually modified schedules to resolve any conflicts that have been introduced by the user.

The *human-guided simple search* (HuGSS) framework [1] allows users to manually modify solutions and steer the optimization process itself. In this approach, users invoke, monitor, and halt optimizations as well as specify the scope of these optimizations. Thus, users control how much effort the computer expends on particular sub-problems. Users can also backtrack to previous solutions. HuGSS was implemented in an interactive vehicle-routing system [1]. Experiments with this system showed that human-guided optimization outperformed an equivalent amount of unguided optimization. An approach similar to HuGSS was used in an interactive graph drawing system, which also allowed users to add constraints to the problem at runtime [6].

The mixed-initiative approach to human-in-the-loop systems uses agents to mediate the cooperation between the computation system and the user to help the user solve an optimization problem. This approach has been applied to transportation scheduling [3], aircraft design [12], and planning [5, 7].

In the above systems, the role of the user in the optimization process has generally been determined by the intuitions of system designers and the availability of interaction and visualization techniques. Experiments have been performed on some of these systems by having users interactively optimize sample problems using the whole system [1, 3]. In more specific investigations, Do Nascimento and Eades [6] evaluated their system under conditions that varied which system features were available to the user.

What is lacking in the design of interactive optimization systems is input from experiments focused on determining which optimization subtasks are best suited to the strengths of the human and which are most appropriate for the computer. The study presented in this paper examines several user tasks within a human-in-the-loop optimization system and compares users' performance in these tasks to the performance of the computer on the same tasks.

METHOD

Participants

The study's three participants were software professionals and computer-science graduate students. All participants had at least 8 hours of previous experience using the experimental application through their participation in a previous study of the interactive optimization system.

Experimental Apparatus

The experimental setup included a tabletop display, called the Optimization Table (OpTable), see Figure 1. The OpTable is comprised of a top-projected image from an IBM-compatible PC running Linux and a wireless keyboard and mouse. The tabletop surface is a 3-foot by 4-foot whiteboard lying face-up on table of the same size. Participants remained seated at the OpTable for the duration of the experimental sessions.

Human Guided Simple Search

We studied a HuGSS system (described in the previous section), available for research purposes, which allows users to guide optimization of *capacitated-vehicle-routing-with-time-windows* (CVRTW) problems. The CVRTW problem is a variant of the traveling-salesman problem in which multiple trucks must deliver goods to a set of customers, under a variety of time, capacity, and geographic

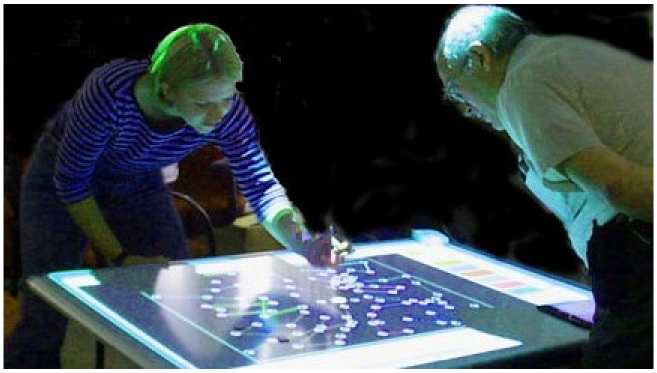

Figure 1. The Optimization Table.

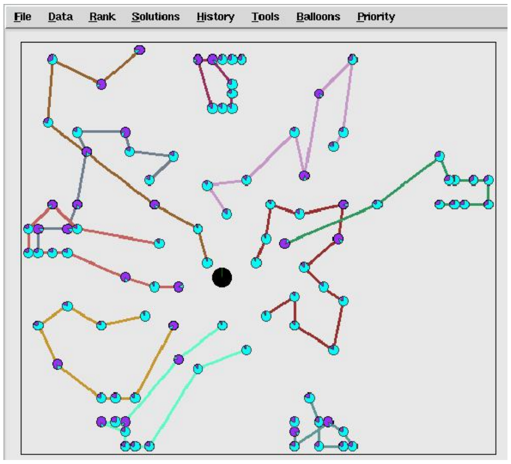

Figure 2. An example of a vehicle routing schedule shown in the HuGSS system. The black circle near the center represents a central depot and the other circles represent the customers. Wedges in the customer circles indicate the time windows during which deliveries can be made. Truck routes are shown by polylines, each in a different color.

constraints. A *solution* to a CVRTW problem is a routing schedule that prescribes which trucks should service which customers and in what order they should service them. The objective is to find the least-cost solution based on the number of trucks and the distance they drive that satisfies all the constraints. Details can be found in [1].

In the HuGSS framework, the users begin with a precomputed solution that an optimization algorithm has produced after many hours of computation. Users then attempt to improve this initial solution by repeatedly performing one of the following three operations: (1) modify the current solution manually, (2) invoke a focused search for improvements to the current solution, or (3) backtrack to a previous solution. Specifically, in the vehicle routing system, users can modify the solution by reassigning a customer to a new route. When a customer is reassigned, the system reoptimizes the two affected routes. Users can also invoke a simple optimization process, which we will refer to as the "search algorithm." When invoked, the search algorithm begins to evaluate ways of reassigning customers to new routes (henceforth called "moves"). When halted by the user, the search algorithm performs the move that most improves the overall schedule. While running, the search algorithm reports the improvement that will result from the best move it has found so far.

The user can also constrain the search algorithm by assigning *mobilities* to each of the customers. These mobilities determine which moves the computer will evaluate. In particular, each customer can be assigned a high, medium, or low mobility and the computer will only consider moves that consist of reassigning high-mobility customers to routes without any low-mobility customers on them. This simple scheme gives the user a great deal of flexibility in determining which moves are evaluated. An advantage of focusing the search is to allow the computer to search more deeply in promising regions of the search space. As a simple example, suppose that a user correctly believes that half of the routes are optimal (for some definition of optimal). By setting all the customers on those routes to low, the user essentially reduces the size of the problem by a factor of two, making it dramatically easier to solve given the exponential nature of the problem.

Thus, a user of the HuGSS vehicle-routing system is always viewing a visualization of one solution, i.e., a routing schedule, to the current problem (see Figure 2). A typical sequence of user actions is to assign mobility values to customers, to invoke the search algorithm (by pressing a button), to watch the progress reports produced by the search algorithm, and then to halt the search at some point (again with a button press), resulting in a schedule update. More details can be found in [1].

Experimental Design

Participants all performed each of the three experimental tasks, in the same order.

Experimental Tasks

Focusing

In the focusing task, users invoked a focused search by setting the mobilities of customers in a series of routing schedules. When a user finished setting the mobilities for a schedule, she would invoke the search algorithm. Participants were told that the goal was to set the customer mobilities to yield the most effective search possible. Once the search algorithm was started, participants could let it continue for at most two minutes, or halt it at any time themselves, in order to see the effect of the mobilities settings they had chosen. For comparison purposes, after each trial the users were shown a description of what happened when an *unfocused* search, i.e., all customers set to high mobility, was precomputed for the same initial schedule. This description showed the improvement that was found, in 10-second intervals, for two minutes of unfocused search. Generally speaking, the user would hope that their focused search would yield greater improvements than the unfocused search.

Finding Targets

In the finding-targets task, users tried to identify certain "target" customers in a series of routing schedules. A target customer was defined as any customer that could be reassigned as part of a move that improved the current score that involved moving two or three customers (as a practical matter, we could not compute all moves that involved reassigning four or more customers). While the focusing task only measures people's ability to guide a particular search algorithm, the finding-targets task more directly measures people's ability to provide information that could be useful for improving search. This information might be used, for example, to probabilistically guide search.

To identify a customer as a target, the user was required to set that customer to high mobility and set non-targets to low mobility. Participants were told that the goal was to select as many potential targets as possible, while trying to avoid selecting customers that were actually non-targets. When the user was finished identifying potential targets, the current schedule was updated to reveal the actual targets along with the customers he had guessed were targets.

Stopping

In the stopping task, participants were given control only of when to halt the search algorithm, on a sequence of routing schedules. For each schedule, the user could examine the board as long as they wanted. When ready, the user would invoke the (unfocused) search and then watch the progress report, shown at the bottom of the application, to determine when to halt the search algorithm. If not halted by the user, the algorithm would stop automatically after two minutes. To give users some sense of what two minutes was worth, they were told to imagine that each search was part of a one hour long optimization session. Participants were told that the goal was to stop the search algorithm after the most significant score improvement had occurred in the least amount of time. They were told that, as an example, if a 10-point improvement occurred after 30 seconds and then another 1-point improvement occurred after 90 seconds, then it would be considered a more effective use of their scheduling time to stop the search shortly after the 30-second point. After the search algorithm was stopped, the user's schedule improvement was displayed. For comparison purposes, the results of a precomputed two-minute unfocused search on the same schedule, shown in 10-second intervals, were also displayed. Thus, users could see what would have happened had they not halted the search. Generally speaking, the users would hope that no large improvements would occur soon after the point at which they had halted the search.

Selection of Test Routing Schedules

The vehicle-routing schedules used for this experiment came from the Solomon benchmarks [15]. Each problem in this corpus has 100 customers. A variety of initial solutions were precomputed for several problems by running the search algorithm from a random starting schedule repeatedly until the schedule could no longer be improved by reassigning one or two customers. A second, less optimal, class of solutions was also generated by restricting the search algorithm to reassigning at most one customer at a time. We then ran the search algorithm on these schedules to compute the information needed to provide feedback to the users for each task described above.

Selection criteria for the initial routing schedules used for the experimental trials differed slightly based on the experimental task. For the focusing task, the solutions were categorized based on whether a two-minute unfocused search would improve the schedule or not. In the focusing trials, we used an equal number of schedules that would and would not be improved by an unfocused search. For the stopping task, the routing schedules were categorized based on the number of times a different improvement would be reported in the feedback to the user. For example, if the search algorithm found a better move at 30 seconds, and then again at 80 seconds, this would be considered two changes. (If multiple changes occurred within a 10-second window, however, this was counted as a single change.) In the stopping trials, initial solutions were used that yielded no change (0 schedule changes), little change (1-2 schedule changes), and large change (3+ schedule change). For the finding-targets task, routing solutions were selected that had no more than 20 target customers.

Procedure

The entire experimental session lasted 3 to 4 hours. with breaks between the experimental tasks. The experiment started with a 20-minute refamiliarization session with the HuGSS application. Next, participants performed the focusing task, the finding-targets task, and then the stopping task. Participants began each experimental task trial with a new vehicle-routing schedule displayed in the HuGSS system. Once users became familiar with the routing schedule by visually inspecting the schedule, they began the appropriate actions to complete the experimental task as described in the experimental tasks section above.

In both the focusing and finding-targets tasks, participants each performed one practice trial and then ten randomly ordered experimental trials. For the stopping task, participants each performed one practice trial and then nine experimental trials. The nine experimental stopping trials included three sets of routing-schedule sequences. Each sequence contained a starting schedule followed by two successive improvements on that schedule. Each successive schedule in a sequence was the result of a precomputed unfocused two-minute search on the previous routing schedule.

Data Analyses

Data were gathered from two sources during the study, computer logs and field notes recorded by a researcher during the experiment. Mobility settings for the focusing task, user selections for the finding-targets task, and

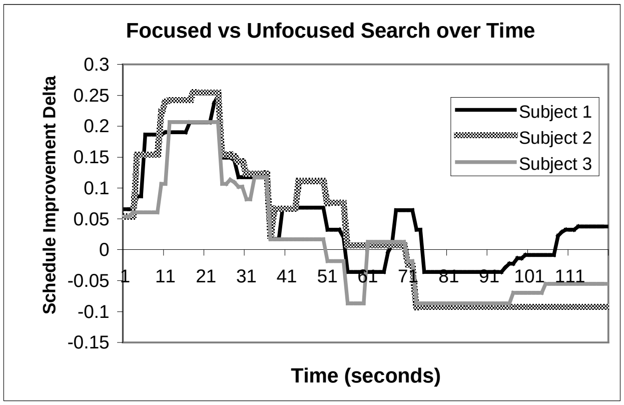

Figure 3. The normalized difference between focused and unfocused search results, over all schedules. Points above zero indicate focused search outperformed unfocused search.

optimization times for the stopping task were extracted from the computer log files.

Post-processing was performed on the focusing task data to enable comparison of participants' results to the results of the computer's unfocused search for the corresponding initial solutions. Each user's mobility settings for the focusing task trials were applied to a full two-minute focused optimization and its progress was recorded every minute. Therefore, regardless of when the user stopped the optimization during the trial, both the optimization progress and end schedule improvement was known for the total time period for all trials.

DISCUSSION OF RESULTS

The goal of this study was to elucidate how best to leverage the strengths of the human collaborator in an interactive optimization system. We had people repeatedly perform various subtasks of the overall optimization process within an interactive system. We hoped to gain insight into which of these subtasks people perform well, how much system use varies from user to user, and which aspects of the system the users' found especially frustrating or helpful. We were also interested in understanding how best to train people to effectively use interactive optimization systems.

Focusing

To explore the users' ability to focus the search algorithm process we examined several aspects of the focusing task.

One question we wanted to answer was whether the subjects' mobility settings improved the performance of the search algorithm compared to an unfocused search. (Note that this is not the same as trying to prove that human-guided search is better than automatic search. To show this would require exploring all possible ways of automatically setting mobilities. Furthermore, it is possible that computers could one day automate whatever strategy the human users are using to guide the search.) Focused searches improved 63.3% of the routing schedules (subject 1 and 3 both improved 70% of their schedules, subject 2 improved 50%), while unfocused searches only improved 50%, which followed from our selection of the schedules as described above. It is promising that users were able to improve 26.6% of the schedules that the computer could not improve.

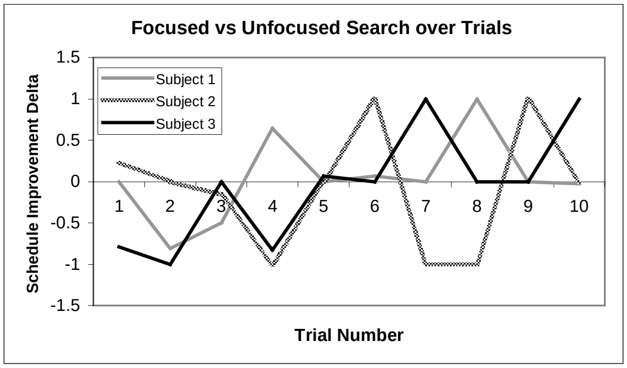

Figure 4. The normalized difference between focused and unfocused search results per trial.

To compare the extent of the improvements made by focusing the search, we normalized the scores for a schedule based on the maximum improvement found by any focused or unfocused search on that schedule. Thus, the maximum score possible for a schedule was 1.0 and the score for any search that did not improve the schedule was 0. Figure 3 shows the average normalized improvement, compared to unfocused search, for each user for each second of the two minutes of search. Thus, in this figure, any point above zero represents better performance than unfocused search, while any point below zero represents worse performance than unfocused search. As shown in the figure, after two minutes of search, only one user's focused search outperforms the unfocused search. Interestingly, the focused search of all three users significantly outperforms unfocused search for the first 30-40 seconds of search. This shows that the users were able to identify regions of the solutions that were especially promising for shorter searches. If future testing confirms this pattern, then interactive optimization systems could be designed to take advantage of this, e.g., by relaxing the user's focus as time elapses or encouraging shorter searches. The latter would allow the user to evaluate more candidate solutions during an optimization session.

Furthermore, it seems likely that the users improved during the course of the experiment. Figure 4 shows the normalized improvement of each subject's focused search compared to unfocused search for each trial. (The users were given the same set of solutions in random order.) Although there is not enough data to confirm a learning trend, it appears that learning took place. This suggests that the system used in this experiment could be developed into a training system to help people learn to set mobilities more effectively.

Table I. Mobilities distribution across users and board.

Source	High	Medium	Low	Same (%)
All Subjects	41.27	10.60	49.13	
Subject 1	34.10	0.00	66.90	
Subject 2	29.70	17.30	54.00	
Subject 3	60.00	14.50	26.50	
Board 1	42.67	29.67	28.67	7
Board 2	44.00	15.00	42.00	1
Board 3	42.00	10.67	48.33	5
Board 4	44.33	5.67	51.00	62
Board 5	47.33	4.67	49.00	86
Board 6	41.67	12.33	47.00	0
Board 7	37.67	1.67	61.67	28
Board 8	39.33	10.00	51.67	8
Board 9	35.00	13.00	53.00	23
Board 10	38.67	3.33	59.00	6

We were also interested in comparing the focusing styles of the different users, as well as the choices made by different users on the same schedules. Table I shows the average number of high, medium, and low mobilities used by each subject, as well as for each routing schedule. Additionally, for each schedule, it shows the percentage of customers who were assigned the same mobility by all three users. As shown in the table, Subject 1 produced much more focused searches than Subject 3, and never used medium mobility. Additionally, the average number of high-mobility customers was surprisingly similar from schedule to schedule. This suggests that people applied a similar amount of focus on each schedule. Two subjects indicated that they were trying to make the most of the two-minute period by focusing the search narrowly enough so that the computer would consider complex moves during the two-minute search time, yet not so narrowly that the search algorithm would complete before the two minutes ended.

The table also shows that people used similar mobility settings on some schedules but very different settings on others. This suggests that people often see different promising areas when they look at the same schedule. Subjects seemed to use a variety of strategies to determine how to set customer mobilities. One subject used the truck capacities as a guide. For example, he would set customers on an underutilized truck to medium mobility and customers on nearby, more heavily utilized trucks to high mobility. These mobilities would encourage the search algorithm to redistribute the customers on these routes. Other users focused the search algorithm on overlapping routes containing customers with flexible time-windows. These, and other strategies used, all have merit, and often produced different improvements in the schedules. Consequently, it seems valuable to design collaborative optimization systems that support groups of people working together to solve an optimization problem.

Table II. User performance for the finding-targets task.

User	True Positive	True Negative	False Positive	False Negative	Precision†	Recall‡
1	9	843	48	100	0.158	0.083
2	18	815	76	91	0.191	0.165
3	26	633	258	83	0.092	0.239
Mean	17.67	763.67	127.33	91.33	0.147	0.162

†Precision is the ratio of correct selections to selections.
‡Recall is the ratio of correct selections to actual targets.

Finding Targets

One claim made of the HuGSS system is that people can use their visual perception to identify promising regions in which to focus the search algorithm. The goal of this task was to determine if the users could identify customers that would participate in moves that would improve a given solution. Table II shows the results of this task in terms of how many of the targets the users correctly selected (true positives), how many of the non-targets they correctly did not select (true negatives), how many non-targets they incorrectly selected (false positives) and how many targets they failed to select (false negatives).

In all ten boards, there were 1000 customers and 109 actual targets (true positives plus false negatives), and thus random selection of targets would yield, on average, a precision of .109. That is, 10.9% of randomly selected customers would actually be targets. Two of the three users had substantially higher precision. This suggests that they can provide a search algorithm with valuable information, namely which customers are more likely to be involved in a move that improves the schedule.

These results are surprising, considering that users found this task to be quite frustrating. Some users' comments made during this session were: "this is so frustrating"; "it's very hard for the eye to measure where things will move because of the time windows"; "this [task] is impossible to predict"; "this is brutal"; and "how can you determine the best customer to move, that's what the computer is good at". Part of the users' frustration resulted from the scarceness of information given in the trial feedback. Only the target customers were indicated on the feedback screen, without any indication of the receiving routes for each target nor any indication of the groupings of customers that constituted a two- or three-customer move. Users found this frustrating, because they could not generalize or create rules to help them improve in the task. Two of the users tried to learn how to improve their accuracy by examining the feedback information but they found it very confusing. This

Table III. Average stopping time and schedule improvement across all trials.

Stopping Source	Average Time	Average Schedule Improvement
Subject 1	58.67	0.810
Subject 2	57.00	0.738
Subject 3	56.67	0.801
Fixed at 60	60.00	0.677
Fixed at 90	90.00	0.800
Fixed at 120	120.00	1.000
Random	65.00*	0.637

* The average random stopping time is skewed above the 60- second halfway point because it is not possible to stop at zero seconds.

suggests that with more informative feedback, users might be able to improve their task accuracy.

Stopping

Although not part of the original HuGSS design, the ability to halt the search algorithm at any time has become an integral part of all HuGSS systems. Users seem to enjoy having this control over the optimization process. Before this experiment, however, it was not at all obvious that people had an accurate sense of when to stop the search.

However, our data shows that users performed very well on the stopping task. We normalized the scores for each trial, dividing the improvement obtained at the point the user halted the search by the maximum improvement that would have been obtained by a full two-minute search. As shown in Table III, all three users stopped the search, on average, after approximately 60 seconds. Although they only used half the total time, two of the users obtained over 80% of the possible schedule improvements and the other user obtained 73% of the possible improvement. This is impressive in that it suggests the users were able to correctly guess that they were getting a large percentage of the possible gains. Table III also shows the results that would be obtained by various fixed policies that always halt the search after a fixed amount of time has elapsed. The users performed much better than a fixed 60-second policy, and about as well as a fixed 90-second policy. This shows people could allocate the computational effort much more effectively than a fixed strategy could.

Figure 5 shows the amount of time each user spent on each schedule. As shown in these charts, there was relatively little variation between users on this task.

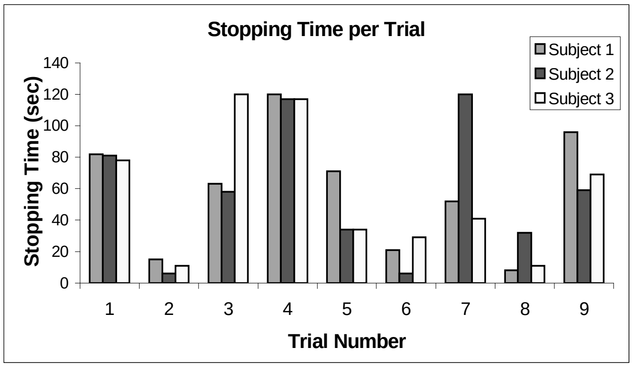

Figure 5. Stopping times for each user shown for all trials.

INITIAL EXPLORATIONS IN TRAINING USERS

As a follow up to the above experiment with experienced users, we also ran a longer series of the focused trials on two novice system users. These participants received two hours of training on the HuGSS system prior to the experimental session. Based on users' comments in the above experiment, slight modifications were made to the experimental application. More trials were added and the per-trial feedback was enhanced to first display the updated schedule that resulted from the user's focused search, and then to display the feedback screen with the results of the unfocused search. During the experimental session for the novice users, one practice trial was performed followed by 40 experimental trials[3], split over two days (20 each day).

The novice users performed quite well. Their focused searches, on average, improved 68.4% of the routing schedules, while the unfocused searches improved only 55.7% of the schedules. The average, normalized improvement (again, we normalized by the best score found by any search per schedule) yielded by a two-minute focused search was 0.52, while the unfocused search yielded only 0.40 improvement. One user's focused searches improved the schedules, on average, by 36.6% and the other's by 25.9%.

Although both users performed well, they often did so on different schedules. In 58% of the schedules one user's normalized score was at least 0.5 higher than the other's. This suggests that people might be using a variety of strategies. It also suggests that invoking multiple searches per schedule might be beneficial and that groups might outperform individuals on this task,

The users' performance over the 40 trials is shown in Figure 6. This figure shows the average normalized improvement, compared to unfocused search, for each experimental trial.

Subject 2's performance decreased in the last few trials of each day. Interestingly, this user commented that he was feeling tired during the end of session and believed it was making the task more difficult. This suggests that the focusing task requires a great deal of user concentration.

Subject 1's average performance was better in the first session, while Subject 2 performed better in the second. Subject 1 reported that he took more risks during the second session. Both performance improvement and the

[3] Due to technical difficulties, the first subject only performed 39 test trials and only 38 of these used a starting schedule also used by the second subject.

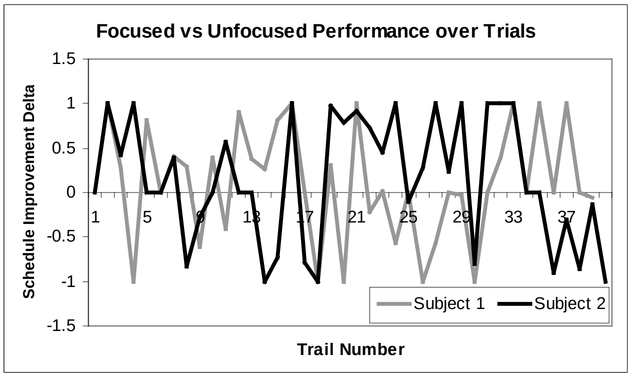

Figure 6. The normalized difference between focused and unfocused search for novices, per trial. Points above zero indicate focused search outperformed unfocused search.

tendency to take risks after some initial training suggest that this system is a good environment for learning how to focus an interactive search.

CONCLUSIONS AND FUTURE WORK

Previous interactive optimization systems have demonstrated that having humans in-the-loop can enhance optimization. However, we believe that further progress in this area requires more rigorous investigation of the individual components of these systems and the assumptions that underlie them.

In particular, a key issue in designing interactive optimization systems is determining the most appropriate division of labour between the human and computer participants. The preliminary studies reported provide some insights on how to build better interactive optimization systems. The studies suggest that people are especially effective at managing how computational effort is expended in the optimization process and focusing short searches, while somewhat less effective at visually identifying promising areas of the search space.

The main contribution of this research, however, is demonstrating that applying HCI evaluation techniques to the study of interactive optimization systems can elucidate the most appropriate division of labour between the human and computer participants. This knowledge, in turn, can help design more effective systems.

The success of our preliminary investigations warrant further, more rigorous experiments involving more subjects and exploring other aspects of the optimization process.

ACKNOWLEDGEMENTS

We would like to thank Joe Marks for his comments and discussion, as well as our subjects for their participation and patience.

REFERENCES

1. Anderson, D., Anderson, E., Lesh, N., Marks, J., Mirtich, B., Ratajczak, D., and Ryall, K. (2000). Human-guided simple search. *Proc. of AAAI 2000*, pp. 209-216.
2. Chien, S., Rabideau, G., Willis, J., and Mann, T. (1999). Automating planning and scheduling of shuttle payload operations. *J. of Artificial Intelligence*, 114, pp. 239-255.
3. Cohen, P.R., Oates, T., and St. Amant, R. (1996). Plan Steering and Mixed-Initiative Planning. In A. Tate, ed. *ARPI Supplement to Proc. 3rd Intl. Conf. on AI Planning Systems*, pp. 105-112.
4. Colgan, L., Spence, R., and Rankin, P. (1995). The Cockpit Metaphor. *Behaviour & Information Technology*, 14(4), pp. 251-263.
5. Cox, M.T., and Veloso, M.M. (1997). Supporting Combined Human and Machine Planning: An Interface for Planning by Analogical Reasoning. *Proc. 2nd Intl. Conf. on Case-Based Reasoning*, pp. 531-540.
6. Do Nascimento, H.A.D, and Eades, P. (2002). To appear in *Proc. of Graph Drawing '02*.
7. Ferguson, G. and Allen, J. (1998). Trips: an integrated intelligent problem-solving assistant. *Proc. 15th National Conf. of Artificial Intelligence*, pp. 567-572.
8. Gleicher, M. and Witkin, A. (1994). Drawing with constraints. *Visual Computer*, 11, 39-51.
9. Kochhar, S. and Friedell, M. (1990). User control in cooperative computer-aided design. *Proc. of UIST '90*, pp. 143-151.
10. Nelson, G. (1985). Juno, a constraint based graphics system. *Computer Graphics*, 19(3) (*Proc. of SIGGRAPH '85*), pp. 235-243.
11. Ryall, K., Marks, J., and Shieber, S. (1997). Glide: an interactive system for graph drawing. *Proc. of UIST '97*, pp. 97-104.
12. Shahroudi, K.E. (1997). Design by continuous collaboration between manual and automatic optimization. Centrum voor Wiskunde en Informatica Technical Report SEN-R9701.
13. Sims, K. (1991). Artificial evolution for computer graphics. *Computer Graphics*, 25(3) (*Proc. of SIGGRAPH '91*), pp. 319-328.
14. Smith, S.F., Lassila, O., and Becker, M. (1996). Configurable, Mixed-Initiative Systems for Planning and Scheduling. In A. Tate, ed. *Advanced Planning Technology,* AAAI Press.
15. Solomon, M.M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35(2), pp. 254-265.
16. Todd, S. and Latham, W. (1992). *Evolutionary Art and Computers*. Academic Press.
17. Waters, C.D.J. (1984). Interactive vehicle routeing. *J. of the Operational Research Society*, 35(9), pp. 821-826.