Designing CIspace: Pedagogy and Usability in a Learning Environment for AI

S. Amershi, N. Arksey, G. Carenini, C. Conati, A. Mackworth, H. Maclaren, D. Poole
Dept. of Computer Science, University of British Columbia
Vancouver, B.C., Canada (604-822-3061)
{samershi, arksey, carenini, conati, mack, maclaren, poole} @cs.ubc.ca

ABSTRACT

This paper describes the design of the CIspace interactive visualization tools for teaching and learning Artificial Intelligence. Our approach to design is to iterate through three phases: identifying pedagogical and usability goals for supporting both educators and students, designing to achieve these goals, and then evaluating our system. We believe identifying these goals is essential in confronting the usability deficiencies and mixed results about the pedagogical effectiveness of interactive visualizations reported in the Education literature. The CIspace tools have been used and positively received in undergraduate and graduate classrooms at the University of British Columbia and internationally. We hope that our experiences can inform other developers of interactive visualizations and encourage their use in classrooms and other learning environments.

Categories and Subject Descriptors

H.5.1 [**Information Interfaces and Presentation (e.g.,HCI)**]: Multimedia Information Systems–*Animations, Evaluation/methodology*; K.3.1 [**Computers and Education**]: Computer Uses in Education-*Computer-assisted instruction (CAI)*

General Terms

Design, Human Factors

Keywords

Interactive visualization, Pedagogy, Artificial Intelligence

1. INTRODUCTION AND MOTIVATION

CIspace [6] is a set of interactive visualization tools for demonstrating the dynamics of common Artificial Intelligence (AI) algorithms. CIspace currently consists of nine Java applets, encompassing many of the topics covered in undergraduate and graduate AI courses including search, constraint satisfaction, deduction, planning, machine learning, robot control and belief and decision networks.

The project was started in 1999 with the aim of developing a suite of applets that could be used to make learning AI more effective and enjoyable. Before the applets were developed, we used the board to show algorithm dynamics during lectures. Feedback from exams and class interaction convinced us that the students often misunderstood these dynamics.

The CIspace applets use a common canvas that allows for the drawing of graphs that in turn enable visualization. A graph is the common representation for problems in all of the applets. The graph can be used to define the problem (e.g., in the Graph Searching applet) or represent its result (e.g., in the Decision Tree learning applet). Algorithms can then be run on these graphs to show the dynamics.

Other tools and resources for enhancing AI teaching and learning have recently been proposed [8], yet many of these efforts have now either been abandoned [1, 11] or have not developed beyond the prototype stage [9]. Some AI applets have been developed at MIT [13], but each of these only demonstrates AI algorithms on one particular problem and users are not able to construct their own problems. These tools are suitable for in-class demonstrations, but are limited in more active scenarios of learning (e.g. assignments). Other tools, each targeting a single or limited range of AI topics, exist and can be found in educational repositories such as AI Topics [2]. But these leave instructors with the problem of searching for appropriate tools for each topic and then learning to use all of these different tools. In designing CIspace, we aim to address these and other issues faced by instructors and students when teaching and learning AI.

Several faculty and students in our department have been involved in the design and development of CIspace. Following an iterative design process, we first identify pedagogical and usability goals and then we devise and implement techniques to achieve these goals through interactive visualization. Finally, we revise our choices in light of feedback from in-class use, usability inspection and user studies.

The rest of this paper is organized as follow: In Section 2 we discuss the pedagogical and usability goals we have identified as important for CIspace. Section 3 describes key design features we have included in CIspace to help achieve our goals. In Section 4 we describe various evaluations we have conducted on CIspace and in Section 5 we discuss ongoing and future research.

2. GOALS

The underlying goal in developing CIspace is to enhance traditional approaches to teaching and learning AI. This objective can be decomposed into two broad categories of pedagogical and usability goals. These categories are not completely orthogonal in that poor usability can mask pedagogical rewards, and few pedagogical benefits can make efforts towards usability irrelevant. Satisfying goals in both categories, however, greatly improves the effectiveness of any educational aid. In this section we describe

ITiCSE '05, June 27–29, 2005, Monte de Caparica, Portugal.

the key pedagogical and usability goals that we have aimed to achieve in the iterative design of CIspace.

2.1 Pedagogical Goals

For a learning aid to be a contribution to education it must provide clear and definite pedagogical benefits over traditionally accepted teaching methods. The following are the pedagogical goals that inform the design of CIspace.

(P1) Increase student understanding of the target domain. In the domain of AI this includes understanding of the mappings from abstract knowledge to graphical representations, as well as the various AI algorithms based on these mappings.

(P2) Support different learning abilities, learning styles and levels of knowledge. Individual Differences Theory [10] suggests that visualization tools will have different effects on learners depending on their different aptitudes and learning styles. Different learners may express varying degrees of preference towards a visualization tool, or may show varying levels of improvement from using it. So in order for an educational tool to accommodate the wide range of students that may comprise a classroom it is important that it provides support for these diverse aptitudes and learning styles. Furthermore, each student's understanding of a subject may change over time and at different rates. So a tool should account for individual learning pace and provide support for novices, but should also continue to support learning as a student's knowledge evolves.

(P3) Motivate and generate interest in the subject matter. Much of the research on visualization tools has focused on measuring learning gains to provide evidence of effectiveness [10]. Yet results from these studies continue to be mixed, contrary to the intuition of educators that algorithm visualizations are pedagogically beneficial [15]. Alternatively, results from preliminary investigations of other factors that may indirectly improve learning outcomes such as a tool's potential to stimulate student motivation seem promising [7, 12].

(P4) Promote active engagement with the tools. One way to motivate students to learn is by actively involving them in the learning process. In the context of visualization tools, this may be achieved by supporting interaction between the student and the tool. Active engagement may not only increase motivation but may significantly improve the pedagogical effects of a visualization tool by allowing students to actively construct knowledge and new understandings [10].

(P5) Support various scenarios of learning, including in-class demonstrations, assignments and exploration. Many educators recognize the potential benefits of using visualizations for in-class demonstrations [15], but employing visualizations within the context of course activities such as assignments or individual exploration can lead to higher levels of engagement. It is in these activities that students can become actively engaged by answering questions about visualizations or underlying concepts, changing algorithm input and analyzing corresponding changes in behavior, or constructing new visualizations [15]. These activities can be more engaging than passive activities such as viewing visualizations in class because they require more effort on the part of students [10]. So, taking advantage of visualization tools in all learning activities in a course can increase the educational benefits of the tools.

2.2 Usability Goals

An educational aid may be designed based on effective pedagogical principles, but without satisfying the usability needs of both educators and students, it would rarely turn into an effective teaching system. Usability encompasses a number of criteria including learnability, efficiency and memorability. These are seemingly intuitive objectives, yet usability deficiencies, especially those involving time to learn and use tools, are the most cited reasons preventing educators from adopting visualization tools [14]. So, it is essential for designers of any pedagogical system to tackle these usability goals from the very early stages of the design process. Here we describe usability requirements that we have identified as essential for our tools.

(UI) Easy to learn, and *(U2) Straightforward and efficient to use.* Minimizing learning overhead means that teachers/students can spend more time teaching/learning the target domain knowledge. Ease of use entails that after learning how to use a tool, it should be easy and efficient for educators and students to carry out their tasks including creating or demonstrating visualizations, or learning target concepts.

(U3) Easy to integrate into a course. Educators report in-class demonstrations as the most common use of algorithm visualizations in computer science courses, with fewer educators incorporating them in homework exercises or making them available for individual exploration [15]. Problems adapting visualizations to individual teaching approaches, course content and other course resources discourages tighter integration of visualizations in a course [15]. Ensuring that a tool is easy to learn and use does help reduce instructor effort, still special attention to the design for easy integration is required so as to motivate more educators to take advantage of these potentially powerful resources.

3. CISPACE DESIGN

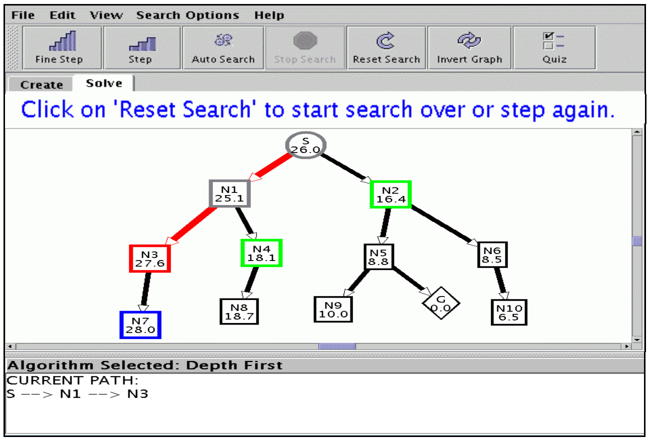
CIspace is an ongoing experiment in pedagogy, and as such we continue to evolve our tools through an iterative approach of design and evaluation. In the following section we describe the design features we have developed to satisfy our pedagogical and usability goals. We justify our choices in the context of recent work on algorithm visualization in computer science education.

3.1 Design Approaches

Coverage and Modularity. CIspace currently includes nine Java applets, each focusing on a different topic traditionally taught in undergraduate and graduate AI courses. The decision to provide coverage of a range of topics is important to reduce the time and effort instructors and students face in searching for visualizations for each new topic [14]. In this way CIspace can be used as a resource throughout a course or sequence of courses (goal *U3*).

The tools were originally created to complement the textbook Computational Intelligence [16], and so were modularized based on topics covered therein. However, since each applet encapsulates a unified and distinct set of fundamental AI concepts, CIspace can also support other popular textbooks [17]. For instructors, this creates flexibility in choosing other resources (goal *U3*). Furthermore, CIspace's modularity gives instructors the option to select only those applets that apply to their intended syllabi (goal *U3*).

Visual Representations. An appropriate graphical representation (see graphs in Figure 1 and 2) for each topic forms the foundation of every applet. The function of these visualizations is to appeal to a wider audience than would text alone by helping to make difficult and often abstract concepts concrete (goals *P1, P2* and *P3*). Though the applets do provide some textual algorithm explanations, they were originally intended to be used along with the textbook [5]. Separating the visualizations from in-depth textual explanations of theory in this way allows instructors flexibility in choosing other supporting resources and in formulating their own explanations tailored to their individual teaching styles (goal *U3*).



Click on 'Reset Search' to start search over or step again.

Algorithm Selected: Depth First
CURRENT PATH:
S ---> N1 ---> N3

Figure 1. Graph Searching Applet

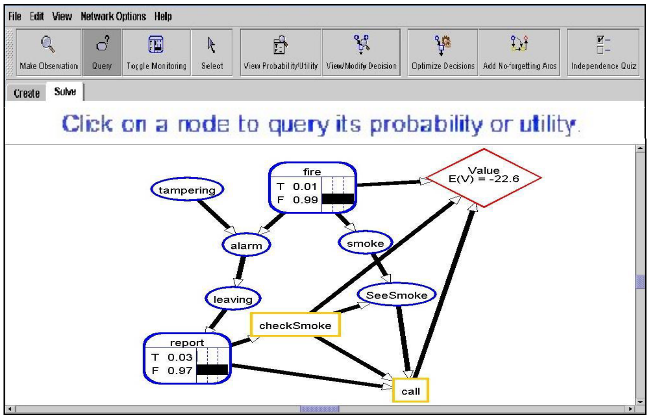
Interactive Simulations. Graphical visualizations alone may not be an improvement over the standard static images used in text or lectures, but animating algorithms using color, movement and textual messages, could be. And this is the intuition of many educators and visualization advocates [15, 18, 9]. Educators also report that the use of simulations have provided for powerful and enjoyable in-class demonstrations of algorithm dynamics (goals *P3* and *P5*) [15]. Yet the interactive nature of the applets, allowing users to manipulate the visual, control the simulation, or directly apply an algorithm to the visual, is what may lead to active engagement and thus improved learning (goals *P4* and *P1*) [10, 18]. We believe this makes the tools appealing for various learning scenarios including lecture demonstrations and individual exploration (goal *P5*).

Control of Algorithm Pace. Each applet provides multi-scaled stepping mechanisms for executing the corresponding algorithms. A user can manually advance through an algorithm at a fine or coarse scale to analyze visualization state changes at every step. In some cases the effects of a step are also reinforced explicitly in text form (see panel below graph in Figure 1). Users can also run the entire algorithm at once at their preferred speed, or, when non-determinism is involved, execute the algorithm many times in a batch run to see performance statistics. In [18], visualizations providing user control over the execution of an algorithm were found to have the most significant pedagogical benefits over other tested design features (goal *P1*). Furthermore, these features enable students to learn at their own pace (goal *P2*).

Comparison of Algorithms. Where appropriate the tools promote comparison of different ways of solving the same problem. This happens for instance in the Graph Searching applet which demonstrates several blind and heuristic search strategies for finding paths in a graph. In [15], the authors define and map levels of user engagement to the six progressive levels of understanding defined by Bloom's taxonomy of learning [4]. They consider comparing and analyzing different algorithms for solving a problem to map to the 4th highest level of understanding, i.e. analysis. This suggests that exploiting this feature, through individual exploration or assignments can increase understanding (goals *P1* and *P5*).

Sample Problems. Each tool is equipped with a set of sample problems that attempt to highlight different and important aspects of a given algorithm. Users can find a short description of each example in the applet's documentation. Sample problems are helpful for students new to a subject [4] or who find it difficult to construct their own meaningful problems (goal *P2*). For instructors, this means less time spent searching for examples (goals *U2* and *U3*).

Creation of New Problems. In each applet, students can experiment with inputting their own data, creating new knowledge bases or constructing their own graphs (goals *P2* and *P4*). This form of actively constructing and evaluating new problems is suggested to map to the highest levels of understanding in Bloom's taxonomy, synthesis and evaluation respectively (goals *P4* and *P1*) [15]. Instructors can also use this feature to create their own problems to show in class, to distribute to students for exploration or to use in assignments (goals *U3* and *P5*).



Click on a node to query its probability or utility.

Figure 2. Bayesian and Decision Network Applet

Consistency. A key feature that supports all of the usability criteria we identified for our system in Section 2.2 is consistency across the applets. Aspects common to all applets include applet layout, menu content and layout, graph entities such as nodes an edges, separate modes for creating and solving problems, and analogous methods for executing algorithms[1]. For instance, the Graph Searching and Bayesian and Decision Network applets (Figure 1 and 2 respectively), have a rather similar look and feel. The result of this consistency is that users familiar with one applet can transfer experience to other applets, minimizing learning time and facilitating use (goals *U1* and *U2*). Consistency also reduces the need for instructors to learn possibly highly varied systems

---

[1] For further interface details, see our Look and Feel document at http://www.cs.ubc.ca/labs/lci/CIspace/CIspaceWebDev/CIspace/newlookandfeel/lookandfeel.html.

authored by different developers in dissimilar styles for each new subject in a course (goal *U3*).

Help. Each applet provides guidance for carrying out tasks in the form of carefully placed messages suggesting how to proceed at any given point during the interaction (goal *U2*). During our usability studies (see Section 4) we determined a noticeable location for these messages to be near the top of the applet window (see Figure 2 directly above the graph). Each applet is also accompanied by a set of increasingly detailed help pages, including *QuickStart, General Help,* and *Tutorial* pages (goals *U1* and *U2*). In our pedagogical experiments (see Section 4), we also developed a three minute instructional video that received overwhelmingly positive feedback from the studies participants. On average these participants reported spending less than ten minutes learning to use the applet being evaluated, including watching this video. This led us to develop video tutorials for all the applets to complement the text-based tutorials (goal *U2*).

The following table summarizes the mapping between key design features of CIspace and the goals they attempt to satisfy. Notice that each goal is achieved by at least two design features. We argue that this level of redundancy provides an adequate foundation for a robust and reliable set of tools.

Table 1. Mapping of design features to goals

	P1	P2	P3	P4	P5	U1	U2	U3
Coverage and Modularity								√
Visual Representations	√	√	√					√
Interactive Simulations	√		√	√	√			
Control of Algorithm Pace	√	√						
Comparison of Algorithms	√				√			
Sample Problems		√					√	√
Creation of New Problems	√	√		√	√			√
Consistency						√	√	√
Help						√	√	

4. EVALUATION

The design of CIspace is informed by intuition, research and evaluation. Through iteration we hope to continue developing and improving the effectiveness of our tools. Thus far, CIspace has been well received by both faculty and students at UBC. We also continue to receive encouraging feedback from educators and students internationally, such as "CIspace seems to be a very interesting resource for AI students," and "The program looks like a great teaching aid."

To help guide our design and identify usability problems, we conducted pilot user tests on each applet during their development. Our participants were volunteer graduate students from the UBC Computer Science department who were familiar with the underlying concepts of the applets they were testing. Each participant was given typical tasks to perform on the applet they were testing and then asked to fill out a questionnaire targeting interface issues. These tests helped us refine some of our design features including finding a noticeable location for the instructional messages for using the applet. Some general issues that arose included users having difficulty navigating through the help pages and noticing information appearing at the bottom of some of the applet windows.

To complement the results of our pilot user testing and more thoroughly evaluate the usability of CIspace, we are in the process of performing a Heuristic Evaluation (HE) of CIspace. HE is a popular usability inspection method which aims to identify usability problems in the design of an interface by having a small set of evaluators examine the interface and judge its compliance with recognized usability principles ("heuristics")[2].

HE of CIspace is being performed by members of our group who were not involved in the original design phase. Arguably, they can have a fresh/unbiased look at CIspace's design approaches. At this point, they have performed an evaluation of the design aspects common across all applets, as described in our Look and Feel document, along with a separate evaluation for only a few applets. However, even though they have not covered all of the applets yet, some interesting findings have emerged:

- CIspace has two modes: "Create" in which new problems are loaded or created from scratch, and "Solve" in which the user can apply algorithms to solve a problem. In both modes the user has access to the same initial Help page. This violates the usability principle that Help should be specific to the task. Different Help entry points should be created for each mode (goals *U1* and *U2*).
- Also noticed in [17], HE points out that at times it can be confusing for students to find intended functionality among the many available menu items, especially while learning the first applet. For instance, HE indicates that the View Menu is confusing because it includes both actions to "view" the applet (e.g. show button text) and actions to "view" the graph (e.g. pan/zoom). To address this and similar problems, we plan to redesign what options appear under each menu in order to make their organization more natural and logical (goals *U1* and *U2*).
- One key goal of HE is to identify opportunities to foster consistency (Consistency principle). In this respect, it was noticed that a Quiz button is only present in some applets. The Quiz feature asks the user to predict the correct next step of the current algorithm and as such can stimulate active engagement (goal *P4*) [10]. A Quiz button should be available in all applets.
- The following issue is more at the boundary between usability and pedagogical goals, but it did emerge during HE. Some CIspace tutorials refer to fundamental AI concepts and definitions without explaining them. This is adequate as long as CIspace is used to supplement the textbook. However, as the applets are more and more frequently used without the companion book, we should consider including this material in the tutorials.

To investigate some of the pedagogical effects of using the CIspace tools we have performed two sets of formal experiments comparing the Constraint Satisfaction Problem (CSP) applet to a traditional method of studying sample CSP problems on paper with static images and text [3]. We measured effectiveness in terms of knowledge gain (goal *P1*), and user preference and motivation (goal *P3*). Our results determined that *i)* the CSP applet was at least as effective in increasing understanding of CSP problems as the traditional method of studying, *ii)* students liked studying with the applet more then on paper (t-test, $p<.007$), and *iii)* students chose to use the applet over the paper format (sign-test, $p<.08$, marginally significant).

---

[2] See http://www.useit.com/papers/heuristic for details.

5. ONGOING AND FUTURE WORK

In addition to addressing the usability problems identified by our user testing and HE, we are currently pursuing two promising areas of development to better achieve some of our pedagogical and usability goals. First, we envision developing user customizable applets whose interfaces can be tailored. Each applet would include a menu listing its available functions, and when given the option the user (typically the student) would be able to select which functions to keep. The interface would then change according to the user's selections. And to guide users in selecting features that may be helpful for learning given their level of domain knowledge, we can provide default settings for beginner, intermediate and expert users (goal *P2*). In effect this would create layered interfaces for the CIspace tools so that users are not overwhelmed by the large number of options when they start using the system (goal *U2*).

Secondly, we are developing author customizable applets for authors creating content for a course, book, tutorial or other web based document. These customizable applets can be distributed as stand alone tools or embedded in a web document inline with text and hypertext. To facilitate the creation of these custom applets, we are developing web based forms where authors can simply select functionality to include and features to stylize the interface. The form will then automatically generate the appropriate html code needed to call the customized applet in an authored document. For instructors developing their own resources, this facility is intended to further our goals of creating tools easy to use and integrate in a course (goals *U2* and *U3*). Furthermore, enabling the visualizations to be used together with textual explanations or other forms of media may, according to Dual-coding theory [10], increase the pedagogical value of the tools (goal *P1*). This may also cater to a wider range of learning preferences and styles as some students may feel more comfortable learning with textual explanations then with visualizations alone (goal *P2*).

Another avenue that could be explored in the future is in developing intelligent tutoring systems to personalize help with using the applets. Pilot participants of our formal experiments tended to misuse some of the CSP applet features[3], suggesting that students may need more guidance in the best use of the applets for learning (goal *U2*). This could be provided through an adaptive help feature that attempts to model a user's level of knowledge and recognizes when a user is having difficulty and then offers tips for resolving the problem (goal *P2*).

6. ACKNOWLEDGEMENTS

The CIspace project has been supported by the Natural Sciences and Engineering Research Council of Canada (NSERC). CIspace has been designed and developed by a team of students and faculty in the Department of Computer Science at UBC. We thank former members of this team including M. Cline, W. Coelho, P. Gorniak, H. Hoos, K. O'Neill, M. Pavlin, J. R. Santos, S. Sueda, L. Tung, and A. Yap.

7. REFERENCES

[1] AI Education Repository. http://www.cs.cofc.edu/~manaris/ai-education-repository/.

[2] American Association for Artificial Intelligence. AI Topics. http://www.aaai.org/AITopics/html/welcome.html.

[3] Amershi, S., Arksey, N., Carenini, G., Conati, C., Mackworth, A., Maclaren, H. and Poole, D. Fostering Student Learning and Motivation: an interactive educational tool for AI. *Tech. Rep. TR-2005-06,* University of British Columbia, 2005.

[4] Atkinson, R.K., Derry, S.J., Renkl, A., and Wortham, D. Learning from Examples: Instructional Principles from the Worked Examples Research. *Review of Educational Research 70,* 2 (2000), 181-214.

[5] Bloom, B. S. (ed.) Taxonomy of Educational Objectives; the Classification of Educational Goals, Handbook I: Cognitive Domain. Addison-Wesley, New York, 1956.

[6] CIspace: Tools for Learning Computational Intelligence. http://www.cs.ubc.ca/labs/lci/CIspace/.

[7] Demetriadis, S., Triatafillou, E., and Pombortsis, A. A Phenomenographic Study of Students' Attitudes Toward the Use of Multiple Media for Learning. *ITiCSE 2003,* ACM Press (2003), 183-187.

[8] Effective Interactive AI Resources Workshop. *IJCAI 2001*.

[9] Greiner, R., and Schaeffer, J. The Aixploratorium: A Vision for AI and the Web. *IJCAI 2001 Workshop*, 3 pages.

[10] Hundhausen, C.D., Douglas, S.A., and Stasko J.T. A Meta-Study of Algorithm Visualization Effectiveness. *Journal of Visual Languages and Computing 13,* 3 (2002), 259-290.

[11] Ingargiola, G. et al. A Repository that Supports Teaching and Cooperation in the Introductory AI Course. *SIGCSE Bulletin 24,* 3 (1994), 36-40.

[12] Kehoe, C., Stasko J., and Taylor, A. Rethinking the evaluation of algorithm animations as learning aids: an observational study. *Int. J. Human-Computer Studies 54,* 2 (2001), 265-284.

[13] MIT OpenCourseWare, Aritificial Intelligence Tools. http://ocw.mit.edu/OcwWeb/Electrical-Engineering-and-Computer-Science/6-034Artificial-IntelligenceFall2002/Tools/.

[14] Naps, T.L., Röβling, G. and Working Group. Evaluating the Educational Impact of Visualization. *ITiCSE 2003,* ACM Press (2003), 124-136.

[15] Naps, T.L., Röβling, G. and Working Group. Exploring the Role of Visualization and Engagement in Computer Science Education. *ITiCSE 2002,* ACM Press (2003), 131-152.

[16] Poole, D., Mackworth, A, and Goebel, R. Computational Intelligence: A Logical Approach. Oxford University Press, New York, 1998.

[17] Russell, N. and Norvig, P. Online Demos (Applets) of AI. http://aima.cs.berkeley.edu/demos.html.

[18] Saraiya, P. et al., Shaffer, C.A., McCrickard D.S. and North, C. Effective Features of Algorithm Visualizations. *SIGCSE 2004,* ACM Press (2004), 382-3.

---

[3] Which were subsequently removed for the actual experiments.