A Framework for Designing and Evaluating Mixed-Initiative Optimization Systems

Arthur E. Kirkpatrick, Bistra Dilkina and William S. Havens
School of Computing Science
Simon Fraser University
Burnaby, British Columbia
Canada V5A 1S6
{ted, bnd, havens}@cs.sfu.ca

Abstract

Mixed-initiative approaches are being applied in combinatorial optimization systems such as planning and scheduling systems. Mixed-initiative optimization systems are based upon collaboration between the system and the user. Both agents possess unique and complementary abilities which can be jointly applied to intractable optimization problems. Yet current approaches to designing and evaluating these systems remain *ad hoc*. In this short paper, we give a precise definition of a mixed-initiative optimization system. We identify the salient characteristics of combinatorial problems which make them suitable candidates for mixed-initiative reasoning. We provide a framework which informs both the design and evaluation of these systems. Using this framework, we characterize the functional requirements of any mixed-initiative optimization system. These requirements can help to establish suitable evaluation criteria for these systems. We conclude by situating recent work in this area within our framework.

Introduction

Mixed-initiative optimization (MIO) systems are systems in which the user and system collaborate to solve combinatorial optimization problems, such as planning and scheduling (Howe *et al.* 2000; Kramer & Smith 2002; Scott, Lesh, & Klau 2002). The benefits of MIO systems have been broadly claimed in the literature. The arguments are based upon several subordinate claims. Two experts ought to be better than one for solving complex combinatorial optimization problems. The system and the user possess unique expertise, and each complements the other. The division of labour between these experts should reflect their inherently different capabilities. The automated solving methods deal with the combinatorics of optmization problems, while users have different kinds of expertise. Often the user of a MIO system will be a professional in the field of application and consequently will know aspects of the problem not adequately modelled by the system. For example, some important constraints may not be part of the model, or some preferences on solutions may not be coded in the objective function. Furthermore, the user's experience may suggest directions for finding good solutions in the search space. Mixed-initiative designs allow this expertise to be incorporated into the problem solving process.

We share the general enthusiasm for mixed-initiative systems. Yet we are concerned that beneath this large tent is hidden a broad range of systems with potentially quite different properties. Blanket statements about "mixed-initiative systems" may only apply to some fraction of these systems. The methods for evaluating these systems remain *ad hoc* and there is little advice available on such issues as the functional components of an effective MIO system, or the performance evaluation of these systems. A common ground is needed for discussions of design and evaluation.

Such a common ground is particularly important for evaluation. There have been several recent evaluations of mixed-initiative systems. Most of these studies have aimed to demonstrate that mixed-initiative systems can be advantageous. For example, Klau *et al.* (2002a) write,"our goal is to show that *some* people can guide search, not that most people can" (p. 46, emphasis in original). The specific mechanisms by which mixed-initiative systems enhance performance remain poorly-understood, although Scott, Lesh, & Klau (2002) have made a promising start. We locate seven limitations in current ad hoc approaches to evaluation:

- We lack precise terminology to distinguish mixed-initiative systems and applications. This makes it difficult to generalize results from one study to a broader class of situations.
- We have a multiplicity of goals, which may overlap or contradict one another.
- We lack clear metrics of progress.
- We have limited means of systematically organizing results to date and those of the near future.
- The lack of precise terms and clear metrics makes it difficult to state clearly falsifiable hypotheses for our research.
- We have no conventional protocols that researchers can use to evaluate a new system.
- Researchers have to account for too many variables when constructing a study.

The systems are being built, and the need is acknowledged.

However, we lack a bigger picture in which to locate specific studies.

Our overall goal is to suggest more precise definitions for the mixed-initiative optimization systems community. Specifically, we make the following contributions in this paper:

- We define the scope of mixed-initiative optimization (MIO) systems.
- We present a general framework that can inform both the design and evaluation of effective MIO systems and identify the main parameters of the problem space and use them to categorize the functional requirements of a MIO system.
- We situate other research results in MIO within our framework. The framework suggests the range of applicability of previous work and can be used to identify potential confounds.

We end with a description of how this framework might be used to structure future research in MIO systems.

Overview of the Framework

We call our proposal a framework rather than the more demanding terms "model" or "theory". We have in mind an analogy with the framework of a house, which provides a structure to support both the work in progress and the finished product. Parts of a framework may be modified as construction proceeds and parts may be thrown away when they are no longer needed. We suggest the mixed-initiative optimization community can benefit from undertaking development and evaluation in this coordinated way—a common framework for developing tasks and protocols, and for interpreting results.

Within such a framework, researchers can begin detailed analysis, extending questions of evaluation beyond, "Is mixed-initiative optimization possible?", to "When and how does it help, and by how much?", and extending systems design beyond "I think this will be useful", to "Previous research gives this a high likelihood of being useful for these applications".

Evaluation is key to this process, but it is hard to get right and requires great effort. As a practical matter, we must break the evaluation process down into smaller pieces. Given the high dimensionality of the design space, experimental manipulations requires separable research questions. Generalization from evaluations of individual designs requires the ability to locate each design within a larger space. By suggesting the questions to ask about a specific design, the framework provides that generalizability and the interconnection of these results with others. In particular, the community will benefit from standardized evaluation protocols. This allows us to build and extend research systems without doing a full evaluation every time.

Definition of MIO Systems

We begin with a definition of a MIO system. A *mixed-initiative optimization system* is an optimization system featuring both:

- interleaved contributions by the user and the system, together converging on a solution to a single problem.
- asymmetric division of labour such that the contributions made by the computer and the user are distinct.

This definition identifies the characteristics of systems for which a common technology can be developed. It circumscribes our shared field of interest. In particular, it separates the goals of this community from the goals of the user modeling community, which emphasizes different aspects of mixed-initiative systems. We are not arguing that user modeling is incompatible with MIO systems, nor that it is irrelevant to some applications of those systems. We are simply emphasizing that user modeling addresses issues that are independent of the issues common to all mixed-initiative optimization. A given application my need one, the other, or both.

Framework Top Level

At the highest level, our framework describes the context in which the system is used and the system itself. Optimization systems ultimately serve human needs, and thus the context describes the system operators and the social context of their work. It includes such issues as who interacts with the system, what others expect from them, and how they do their job. The context introduces requirements that the resulting system must satisfy. We break the context into two parts, the problem domain and the operator expertise.

For our purposes, the mixed-initiative system can also be considered to have two fundamental parts, the interactive visualization and the solver. We emphasize that this is not intended to be a full description of the architectural options available to designers. Rather, our purpose is to list the high-level system components which will most directly be evaluated. Designers devote most of their attention to the system, and evaluations are likely to compare instantiations of various combinations of them, with little consideration of the context. This is fine—in fact, there is no practical way to cover all possible contexts in a single evaluation. We simply recommend that evaluations explicitly specify the details of their intended context. This will permit readers to determine the range of contexts to which the evaluation results may be generalized.

Properties of Context

The framework's elaboration of social influences on a mixed-initiative system emphasizes that the requirements for that system are strongly shaped by the needs and nature of the organization employing it. Each social component has several properties, and the system requirements are derived from the properties (see Table 1). We will describe each property in turn.

Domains with synchronous collaboration feature teams of individuals working in the same room to solve the problem. The classical (and widespread) example would be several individuals standing at a large whiteboard, discussing, writing, and annotating. Mixed-initiative optimizers for such domains will benefit from having interactive displays that per-

Area	Property	Derived Requirement
Domain	Synchronous collaboration	Simultaneous review and update
	Asynchronous collaboration	Traces left for handoff to others
	Unmodellable aspects	Adding specific constraints and revising solution
	Level of constrainedness	Choice of optimization algorithm
	High dynamism	Rapid solution revision
	Answerability	Explanation
	High stakes	High scrutiny, human approval, explanation
	Task givens and goals	Data model chosen
	Task flow	Activity design
Operator expertise		
Domain-independent	Visual grouping	Display proximity, object similarity
	Conceptual models of other programs	Compatibility with other programs
Domain-specific	Models of objects and relationships	Data model
	Heterogeneity of approaches	Diversity of representations and interaction styles

Table 1: Contextual properties in the mixed-initiative optimization framework.

mit simultaneous review and update of the current solution by multiple users, in the same way that a whiteboard can.

By contrast, asynchronous collaboration features team effort, but with individual team members working at different times. For example, there may be only a single operator, but the work is turned over from one shift to the next. In these cases, the interactive display will not need simultaneous review and update, but will instead benefit from providing a mechanism for the first shift to leave traces of their choices and pointers to ongoing problems.

Domains with unmodellable aspects have problems that are difficult or impossible to completely represent in the model. There are a wide variety of reasons why a problem feature may be missing from the model. Amongst other reasons, the feature may be so specialized that the modellers forget to include it until they see a "solution" that violates it, the feature may not be representable in the modeling formalism, or the feature may be so specific that it would be impractical to represent its many permutations in the model. Indeed, given the considerable mixed-initiative folklore about incomplete models, it might be argued that every domain has unmodellable aspects. In any event, domains with unmodellable aspects will benefit from systems that allow the operator to add specific constraints and call for a revised solution.

The level of constrainedness of the domain imposes requirements on the system. If the domain is highly-constrained, the system should use a more sophisticated optimization algorithm. If the domain instead is only weakly constrained, the choice of solver might be different or almost irrelevant.

If the domain is dynamic, with requests frequently added and withdrawn, the system will be required to generate revised schedules rapidly.

If the operator is answerable to others for the choice of plan, the system should provide features that facilitate explanation. These could take the form of annotations and other tools that highlight specific aspects of the plan.

High stakes domains have outcomes that are considered critical by the participants. For such domains, the system should support high levels of operator scrutiny and a final human approval before the plan is carried out. These domains will likely also have high answerability as well, and so the system should also facilitate explanation.

The task statement imposes givens (initial conditions) and goals (the form of the desired solution) upon a domain. Different sets of givens and goals will often apply at different times for a single domain. For example, an airport gate scheduling system may be used in two different modes. First, the system could prepare a master schedule by allocating planes to gates under the assumption that every flight arrives exactly on time. In this case, the givens are the complete plane list and the goal is a complete schedule. Then, on a specific day, the plan would be revised as notices of delays arrived. This second task has different givens (just the planes whose schedules have changed, together with the original master schedule) and a more specific goal (accommodate the delays with minimal disruption to the original plan). Different tasks may require different data representations in the interactive display.

The final domain property is the task flow. Operators will often perform a task in a sequence of steps. For example, they may review all assignments of low-priority items before all high-priority ones (or vice versa). The operator task flow imposes requirements on the system's activity design, the steps that the system requires the user to perform. For example, a system that presented items to the operator in random order would be extremely frustrating for an operator who wished to review them in priority order.

Operator Expertise

Arguments for the use of mixed-initiative optimization systems often emphasize the unique expertise offered by the user. We suggest that this expertise has both domain-independent and domain-dependent properties. The domain-independent expertise consists of the human perceptual skills. The primary interactive display of most MIOS is visual, capitalizing on human skills of grouping visually-

related objects. An effective MIOS should display the problem in a way that allows the operator to draw useful conclusions from object proximity and similarity. Note that even within a given domain, different tasks, with their different givens and goals, might be best served by different displays.

A second form of domain-independent expertise is the operator's experience with the conceptual models of other systems. If the MIOS has a conceptual model that matches that of other software commonly used by the operator population, the operators will find the system congenial. Although this form of expertise is unlikely to have a strong positive impact on the effectiveness of a MIOS, it can have a strongly negative impact. System effectiveness can be substantially reduced if the operators are practiced with a conceptual model that is incompatible with the MIOS they are using.

The domain-specific expertise of the operators will also impose strong requirements on the system design. From both training and experience, operators will think in models of objects and relationships. The structure of these models is highly domain-specific, making it difficult to give specific details. We simply note that the data model presented by the MIOS should be well-matched to the models used by the operators.

The final property in our categorization of expertise is the heterogeneity of solution approaches. An operator may have several ways of solving a problem and different operators within the same community may use different approaches. This is often domain-specific. Some operations communities have strong conventions that all operators are trained to observe, while other communities may be more diverse. We caution designers not to rely too strongly on perceived homogeneity in a community, as even within highly-trained groups there is often subtle variation. In general, an effective MIOS should provide a diversity of views, representations, and interaction styles, to support a diversity of solution approaches.

We have described these contextual properties in detail to emphasize the diversity of contexts in which MIOS might be applied. The effectiveness of a mixed initiative system depends upon how well it is matched to the specifics of the domain and the expertise of the operators. Interpretation of evaluation results must take these factors into account. An otherwise perfectly effective system may have poor performance if it is evaluated on a domain whose requirements are ill-matched to the system's. Results from an evaluation will best generalize to applications whose domains and operator expertise are similar to those of the evaluation.

Properties of Systems

Our model of system properties is deliberately simpler than our model of context. For purposes of summarizing the properties of a system that have the largest effect on evaluation, we break the system down into two parts, the interactive display and the solver (see Table 2).

It is not our intent to list all possible features and properties of mixed-initiative optimization systems. Defining these features and properties is a significant part of the overall research program in these systems. We offer this list as a starting point.

The interactive display has three main properties. The visualization is the visual representation used to display the problem statement and the current solution. An essential outcome of this design is the data model presented to the user.

The second property is the chosen interaction techniques. These will have specific speeds and place certain attentional loads on the operators. An ideal interaction technique will be fast and require so little explicit attention that the operator's reasoning about the problem will not be disrupted. Actual interaction techniques require some compromise of these ideals.

The third property of the interactive display is the detailed visual display design. These choices will determine where and how much the operators can apply their domain-independent expertise to the problem. This includes choices of which aspects of the problem will be represented in close proximity and how data values will be encoded.

The second component of the system highlighted in our framework is the solver. There are many ways of categorizing solvers, and development of new variations is an active area of research. Given that MIOS researchers typically have a strong understanding of the properties of various solvers, we only present here two example properties, showing how they may be connected to the contextual issues described earlier.

Solvers are often categorized as implementing either systematic or local search. Seen in terms of their relationship to the contextual properties described above, the main outcome of this distinction is their suitability for dynamic problem domains. Systematic search, while offering the potential for higher optimization, is less likely to be responsive to shifting requirements. Local search is more likely to apply in these domains.

A second property of a solver is how close its solutions lie to the optimal. Optimality is likely to be of higher value in domains that are capital-intensive , but may not be a possibility for highly dynamic domains, whose volatile constraints make it difficult to even define optimality.

Previous Work

The framework provides a structure for organizing discussion of the research results to date on mixed-initiative optimization. In this section, we review many of these results and locate common threads and unexplored areas in the field.

We start by considering the user modeling community's work on mixed-initiative systems. This literature is concerned with rather different issues than mixed-initiative optimization.

Horvitz (1999) proposed 12 principles for the effective integration of automated reasoning and direct user control. The goal of this integration was an agent that could act as a "benevolent assistant" (p. 160) to the user. The parameters of such an agent are rather different from those of the mixed-initiative optimization systems described in this pa-

Component	Property	Outcomes
Interactive display	Visualization	Data model
	Interaction techniques	Speed, attentional load
	Visual display	Proximity of views, coding of values
Solver	Systematic vs. incremental	Suitability for volatile domains
	Optimality	Quality of solution in highly subscribed domains

Table 2: System properties in the mixed-initiative optimization framework.

per. Assistive agents are expected to have transparent algorithms that perform actions the user also has the resources and representations to perform. The goal is to relieve the user of tedious, repetitive actions. From this perspective, the system has the initiative most of the time and needs to decide when to engage the user based on a user model—a set of beliefs about the abilities, goals and intentions of the user.

Fleming & Cohen (2001) develop guidelines for the design and evaluation of mixed-initiative systems. However, they start with the assumption that the central problem is how the system will take the initiative to request assistance from the user. They propose an approach similar to Horvitz, based on having an explicit model of the user's intentions and abilities.

The work on assistive agents is explicitly excluded by our definition, because such systems do not have an asymmetric division of labour between user and system. In contrast to assistive agents, mixed-initiative optimization systems are designed to produce degrees of optimization that the operator simply could not achieve unaided. Their algorithms are unlikely to be transparent, and their choices may require considerable effort for the operator to understand. The scheduling task is a primary focus of the operator's job and is likely to be her highest priority task. Indeed, the description of the human partner as an "operator" rather than a "user" emphasizes this primacy of the task.

Rich, Sidner, & Lesh (2001) cast human-computer interaction in terms of a collaborative dialogue process between the user and an intelligent interface agent. The authors base their approach to mixed initiative on human collaboration. They argue for an interface agent that engages in a discourse with the user in a similar way that the user would engage with another human. In particular, they argue that an intelligent user interface has to support the following questions:

- Who should/can/will do ____?
- What should I/we do next ____?
- Where am/was I ____?
- When did I/you/we do ____?
- Why did you/we (not) do ____?
- How do/did I/we/you do ____?

Rich et al. propose an intermediate level of software explicitly concerned with managing these questions.

We consider Rich et al.'s questions complementary to our framework, as they are more abstract and at a much higher level. We believe that many of the crucial elements for effective MIO system design lie in its rich, specific context. The questions will best be framed in that context, which may be difficult or impossible if the algorithm that generates the question is insulated from the context.

Howe *et al.* (2000) present a study on mixed initiative scheduling for the Air Force satellite control network. This scheduling problem is oversubscribed—no feasible schedule can satisfy all the requests. They propose a MIO system where the system finds a good but infeasible solution and lets the user negotiate the infeasibilities. A mixed-initiative approach is appropriate for this application because it is hard to express the true objective with a weighted linear sum of criteria, and because the dynamic arrival of emergency requests changes the problem specification as the solver runs. The authors point out the limited number of designs in the research literature mixed-initiative systems. They incorporate in their prototype some of these designs: providing an interactive Gantt chart where the user can interact with a schedule at an abstract, graphical level that hides schedule implementation and optimization details to an appropriate degree; and allowing the user to change the schedule, then call the scheduler to propagate the effects and to optimize, if possible. In terms of our framework, this paper emphasizes the dynamism and highly constrained nature of their domain. Their comments about the limited number of available design ideas and their use of Gantt charts demonstrate the importance of domain-specific models of objects and relationships.

Kramer & Smith (2002) describe the AMC Barrel Allocator, a mixed-initiative resource allocation tool for airlift and air-refueling management. They argue that a dynamic environment is not the only reason for using the mixed-initiative approach. It is also necessary to achieve the transition from manual to fully automated system. Kramer & Smith emphasize that a mixed-initiative optimization system allows a continuum of automation. In deploying their research to production, they found that operators must first gain trust and understanding of the system by inspecting solutions and performing what-if scenarios trust a system before they will accept it in a mission-critical workflow. Kramer and Smith point out that one of the main functional requirements for a mixed-initiative system is to provide explanations for system decisions. In terms of our framework, this paper emphasizes the dynamism, high stakes, and answerability of their problem domain, and argues that the MIO system must be well-matched to the task flow.

Klau *et al.* (2002b) present the HuGS Platform, a toolkit that supports development of human-guided search systems. They discuss four different applications built in the HuGS platform: a graph layout problem that minimizes edge cross-

ings between nodes, a modified version of the travelling salesperson problem, a simplified version of the protein folding problem, and a jobshop application.

They present several motivations for mixed-initiative optimization. First, users need to understand and trust the generated solutions in order to effectively implement, justify and modify them, what we would call the answerability of the system. Second, the problem model usually includes only partially specified constraints and criteria, what we call the unmodellable aspects of the domain. They argue that a mixed-initiative system also leverages on human abilities that outperform the systems: visual perception, learning from experience, and strategic assessment. These strengths range over properties of both domain-dependent and domain-independent expertise.

Each application in the HuGS platform provides visualizations to display the current solution to the user for inspection and modification. Klau et al. argue that the usefulness of the system depends highly on the quality of visualization and recommend visualizations that highlight differences from the previous solution. They suggest an evaluation of the quality of the visualization by running a series of experiments on the same problems for the same time, and using two different visualizations. They propose a visualization quality metric of the number of optimal solutions that users are able to produce. In terms of our framework, these evaluation methods are focused on the interactive display component. Klau et al. end by highlighting some ongoing challenges for the mixed-initiative systems: large-scale problems where the whole solution cannot be viewed at once (again, located within the interactive display component of our framework), and mixed-initiative systems where there is more than one human user (synchronous collaboration).

Scott, Lesh, & Klau (2002) give a lucid outline of the benefits of using a mixed-initiative optimization system. Their research focuses on evaluating a specific aspect of mixed-initiative optimization systems. The authors argue that the design of interactive optimization systems needs 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. Their study examines several user tasks within a mixed-initiative optimization system for vehicle routing and compares users' performance in these tasks to the performance of the computer on the same tasks. They evaluate the users' contribution on three different subtasks: focusing search through mobilities, finding targets that guide the search towards better solutions, and controlling computational effort by halting the search. Their studies suggest that people are especially effective at managing how computational effort is expended in the optimization process and at focusing short searches. However, the experiments showed that humans were somewhat less effective at visually identifying promising areas of the search space.

In terms of our framework, the work of Scott et al. is motivated by the contextual concerns of answerability and the unmodellable aspects of the domain. Because their experimental participants were not vehicle routing specialists, the evaluation focused on HuGS' support for domain-independent expertise. Their project is a carefully-done, substantial study with strong controls and high validity. However, their paper itself does not specify the context. By providing a context, our framework allows more precise generalization from these results.

Conclusion

The potential benefits of mixed-initiative optimization systems are suggested by informal reasoning from basic principles and has been demonstrated by initial research. Having established its basic feasibility, we can now turn to questions of how much, and under what circumstances, and through which mechanism we can benefit from a MIO system. We have argued that context is rich and diverse, and that the effectiveness of a MIO system is determined by the degree to which it is matched to the requirements of its context. Key MIO system design decisions should be evaluated in terms of the context in which the system will be used, or in terms of requirements that are shared across multiple contexts. Our framework highlights this role of context and provides a more detailed language for describing the relationship between context and system. We hope that these more precise descriptions can support the construction of a more consistent and solid structure of mixed-initiative optimization research.

Acknowledgments

The work in this paper was supported by grants from Precarn, Inc. and the National Sciences and Engineering Council of Canada (NSERC).

References

Fleming, M., and Cohen, R. 2001. A user modeling approach to determining system initiative in mixed-initiative AI systems. In *UM '01: Proceedings of the 8th International Conference on User Modeling 2001*, 54–63. Springer-Verlag.

Horvitz, E. 1999. Principles of mixed-initiative user interfaces. In *CHI'99: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 159–166. ACM.

Howe, A. E.; Whitley, L. D.; Barbulescu, L.; and Watson, J.-P. 2000. Mixed initiative scheduling for the air force satellite control network. In *Proceedings of Second NASA International Workshop on Planning and Scheduling for Space*.

Klau, G. W.; Lesh, N.; Marks, J.; and Mitzenmacher, M. 2002a. Human-guided tabu search. In *AI'02: Proceedings of Eighteenth National conference on Artificial intelligence*, 41–47. American Association for Artificial Intelligence.

Klau, G.; Lesh, N.; Marks, J.; Mitzenmacher, M.; and Schafer, G. 2002b. The HuGS platform: A toolkit for interactive optimization. In *AVI'02: Proceedings of Advanced Visual Interfaces (AVI)*.

Kramer, L., and Smith, S. 2002. Optimizing for change: Mixed-initiative resource allocation with the AMC Barrel Allocator. In *Proceedings of the 3rd International NASA*

Workshop on Planning and Scheduling for Space. Houston: The Institute for Advanced Interdisciplinary Research.

Rich, C.; Sidner, C. L.; and Lesh, N. 2001. Collagen: Applying collaborative discourse theory to human-computer interaction. *AI Magazine* 22(4):15–25.

Scott, S. D.; Lesh, N.; and Klau, G. W. 2002. Investigating human-computer optimization. In *CHI '02: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 155–162. ACM.