

POSTER PAPERS

POSTER PAPERS

AUTOMATED REASONING

Active Probing Strategies for Problem Diagnosis in Distributed Systems

Mark Brodie, Irina Rish, Sheng Ma, Natalia Odintsova

IBM T.J. Watson Research Center, 19 Skyline Drive, Hawthorne, NY 10532

mbrodie, rish, shengma@us.ibm.com, nodintsova@hotmail.com

Abstract

We address the task of problem determination in a distributed system using *probes*, or test transactions, which gather information about system components. Effective probing requires minimizing the cost of probing while maximizing the diagnostic accuracy of the probe set. We show that pre-planning an optimal probe set is NP-hard and present polynomial-time approximation algorithms that perform well. We then implement an active probing strategy which selects probes dynamically and show that it yields a significant reduction in probe set size in both simulation and a real system environment.

Active Probing System

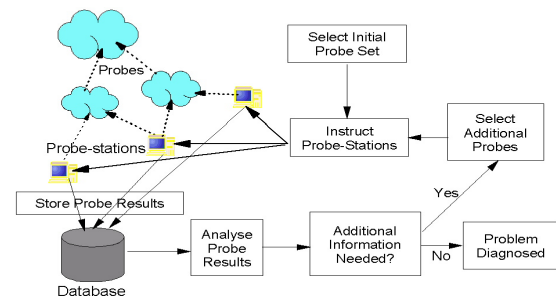


Figure 1: Active Probing System.

1 Introduction

Accurate diagnosis in a complex, multi-component system by making inferences based on the results of various tests and measurements is a common practical problem. Developing cost-effective techniques for diagnosis in such systems requires that high accuracy be achieved with a small number of tests. In this work we present a generic approach to this problem and apply it specifically to the area of distributed systems management.

The key component of our approach is an “active” measurement approach, called *probing*. A probe is a test transaction whose outcome depends on some of the system’s components. Diagnosis is performed by appropriately selecting the probes and analyzing the results. For distributed systems, a probe is a program that executes on a particular machine (called a probe station) by sending a command or transaction to a server or network element and measuring the response. The *ping* program is probably the most popular probing tool that can be used to detect network availability. Other probing tools, such as IBM’s EPP technology, provide more sophisticated, application-level probes. For example, probes can be sent in the form of test e-mail messages, web-access requests, and so on.

Previous work studied the problem of probe-selection [Brodie *et al.*, 2001], [Ozmutlu *et al.*, 2002] and efficient diagnosis from probe outcomes using approximate inference in Bayesian networks [Rish *et al.*, 2002], but the NP-hardness of

probe-selection and the idea of active probing were not considered there.

2 Pre-Planned Probing

In pre-planned probing, given an initial set of probes, we want to compute the smallest subset of probes such that each system state will produce a different set of probe outcomes, allowing the state to be uniquely determined.

Proposition 1 *Probe-set selection is NP-hard.*

Proof: The proof is via a reduction from 3-Dimensional Matching (defined in [Garey and Johnson, 1979]). Details can be found in [Brodie *et al.*, 2003].

We implemented two approximation algorithms for pre-planned probing - greedy (quadratic) search and subtractive (linear) search. Greedy search starts with the empty set and adds at each step the “best” of the remaining probes - the probe which maximizes the information gained about the system state. Computational complexity is quadratic in the size of the initial probe set. Subtractive search starts with the complete set of available probes, considers each one in turn, and discards it if it is not needed. Computational complexity is linear in the size of the initial probe set. Neither algorithm is optimal in general - empirical results are provided below.

Active Probing

Input: A set of available probes $\mathbf{P}$ and a prior distribution over system states $Pr(\mathbf{X})$.

Output: A set $\mathbf{P}_a$ of probes and their outcomes, posterior distribution $Belief(\mathbf{X})$ and its support $\mathbf{S}$.

Initialize: $Belief(\mathbf{X}) = Pr(\mathbf{X})$, $\mathbf{P}_a = \emptyset$, $\mathbf{S} = \text{support of } Pr(\mathbf{X})$.

do

1. select current most-informative probe:

$$Y = \arg \max_{Y \in \mathbf{P} \setminus \mathbf{P}_a} I(\mathbf{X}; Y | \mathbf{P}_a)$$

2. execute Y ; it returns $\hat{Y} = y$ (0 or 1)

3. update $\mathbf{P}_a = \mathbf{P}_a \cup \{Y = y\}$

4. update $Belief(\mathbf{X}) = Pr(\mathbf{X} | \mathbf{P}_a)$

while $\exists Y \in \mathbf{P}$ such that $I(\mathbf{X}; Y | \mathbf{P}_a) > 0$

Return $\mathbf{P}_a$, $Belief(\mathbf{X})$, $\mathbf{S} = \text{support of } Belief(\mathbf{X})$.

Figure 2: Algorithm for Diagnosis using Active Probing.

3 Active Probing

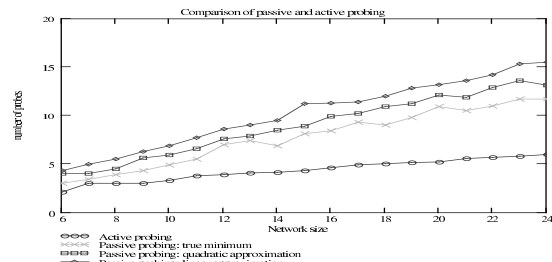
Active probing, allows the selection of later probes to depend on the results of earlier probes. An active probing system is outlined in Figure 1. Probe-stations issue probes which traverse different parts of the network. The results of the probes are analyzed to infer what problems might be occurring. If additional information is needed in order to locate the problem, the “most-informative” probe, which provides the largest information gain about the true state of the system, is computed and sent. When additional probe results are received further inferences are made, and the process repeats until the fault is localized. The algorithm is described in Figure 2.

Active probing allows fewer probes to be used than if the entire probe set has to be pre-planned, though more complex inferential machinery is needed to support it.

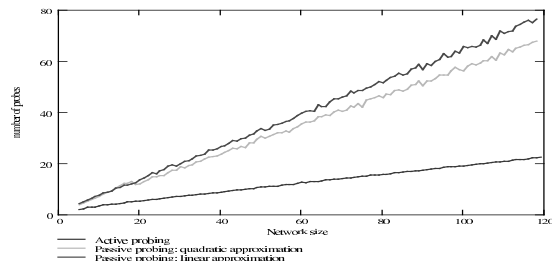
4 Results

Simulation results, using randomly generated networks, are shown in Figure 3a (small networks) and Figure 3b (large networks, where finding the true minimum is impractical). The approximation algorithms for finding the smallest probe set perform well and are close to the true minimum set found by exhaustive search. It is also clear that active probing considerably reduces the size of the probe set when compared with pre-planned, or “passive”, probing.

In Figure 4, we report the results on a real network. This network, supporting e-business applications, includes many servers and routers, and its performance and availability depend on a number of software components. A set of 29 probes was manually selected by an expert. Exhaustive search shows that the minimum number of probes is actually only 24 probes, a saving of 17%. Greedy search returned 24 probes, while subtractive search found 25 probes. For active probing, Figure 3c shows the number of probes needed for localization for each of the different single faults. The number of probes needed never exceeds 16 probes and averaged only 7.5 probes, a large improvement over the 24 probes used by pre-planned probing.



(a) Small Networks (simulations)



(b) Large Networks (simulations)

Figure 3: Active versus pre-planned probing results for randomly generated networks.

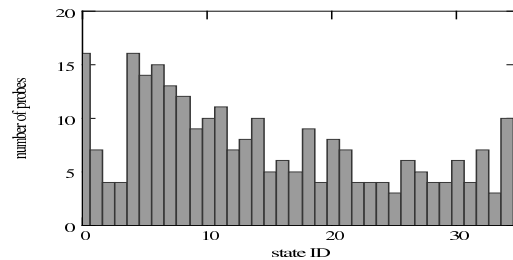


Figure 4: Active probing averages 7.5 probes in a real application where pre-planned probing requires 24 probes.

References

- [Brodie *et al.*, 2001] M. Brodie, I. Rish, and S. Ma. Optimizing probe selection for fault localization. In *Distributed Systems Operation and Management*, 2001.
- [Brodie *et al.*, 2003] M. Brodie, I. Rish, Sheng. Ma, and N. Odintsova. Strategies for Problem Determination Using Probing. Technical report, IBM T.J. Watson Research Center, 2003.
- [Garey and Johnson, 1979] M.R. Garey and D.S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-completeness*. W.H. Freeman and Co., San Francisco, 1979.
- [Ozmutlu *et al.*, 2002] H.C. Ozmutlu, N. Gautam, and R. Barton. Zone recovery methodology for probe-subset selection in end-to-end network monitoring. In *Network Operations and Management Symposium*, pages 451–464, 2002.
- [Rish *et al.*, 2002] I. Rish, M. Brodie, and S. Ma. Accuracy vs. Efficiency Trade-offs in Probabilistic Diagnosis. In *Proceedings of the The Eighteenth National Conference on Artificial Intelligence (AAAI-2002)*, Edmonton, Alberta, Canada, 2002.

A Resolution Theorem for Algebraic Domains

Pascal Hitzler

Dresden University of Technology
01062 Dresden, Germany
phitzler@inf.tu-dresden.de

Abstract

W.C. Rounds and G.-Q. Zhang have recently proposed to study a form of resolution on algebraic domains [Rounds and Zhang, 2001]. This framework allows reasoning with knowledge which is hierarchically structured and forms a (suitable) domain, more precisely, a coherent algebraic cpo as studied in domain theory. In this paper, we give conditions under which a resolution theorem — in a form underlying resolution-based logic programming systems — can be obtained. The investigations bear potential for engineering new knowledge representation and reasoning systems on a firm domain-theoretic background.

1 Introduction

Domain Theory [Abramsky and Jung, 1994] is an abstract mathematical theory for programming semantics and has grown into a respected field on the borderline between mathematics and computer science. Studying relationships between domain theory and logic is of acknowledged importance in the field. There has been much work on the use of domain logics as logics of types and of program correctness, with a focus on functional and imperative languages. However, there has been only little work relating domain theory to logic programming or other AI paradigms, two exceptions being the application of methods from quantitative domain theory to the semantic analysis of logic programming paradigms studied by Hitzler and Seda [Hitzler and Seda, 200x], and the work of Rounds and Zhang on the use of domain logics for disjunctive logic programming and default reasoning [Rounds and Zhang, 2001].

The latter authors, in [Rounds and Zhang, 2001], introduced a form of clausal logic generalized to coherent algebraic domains, motivated by investigating the logical content of the Smyth powerdomain construction. In the following, we study this clausal logic, henceforth called *logic RZ* for convenience. The occurrence of a proof theory based on a generalized resolution rule poses the question whether results underlying resolution-based logic programming systems can be carried over to the logic RZ. One of the most fundamental results underlying these systems is the *resolution theorem*

which states that a clause X is a logical consequence of a theory T if and only if it is possible to derive a contradiction, i.e. the empty clause, via resolution from the theory $T \cup \{\neg X\}$ [Robinson, 1965].

What we just called *resolution theorem* is certainly an immediate consequence of the fact that resolution is sound and complete for classical logic. However, it is not obvious how it can be transferred to the logic RZ, mainly because it necessitates negating a clause, and negation is not available in the logic RZ in explicit form. This observation will lead our thoughts, and in the end we will develop conditions on the underlying domain which ensure that a negation is present which allows to prove an analogue of the theorem.

2 The Logic RZ

We assume that the reader is familiar with basic domain-theoretic notions, and we will follow the terminology from [Abramsky and Jung, 1994] and [Rounds and Zhang, 2001]. In the following, $(D, \sqsubseteq)$ will always be assumed to be a coherent algebraic cpo. We will also call these spaces *domains*. An element $a \in D$ is called an *atom*, or an *atomic element*, if whenever $x \sqsubseteq a$ we have $x = a$ or $x = \perp$. The set of all atoms of a domain is denoted by $A(D)$.

Definition 2.1 Let D be a domain with set $K(D)$ of compact elements. A clause is a finite subset of $K(D)$. If X is a clause and $w \in D$, we write $w \models X$ if there exists $x \in X$ with $x \sqsubseteq w$. A theory is a set of clauses, which may be empty. An element $w \in D$ is a model of a theory T , written $w \models T$, if $w \models X$ for all $X \in T$. A clause X is called a logical consequence of a theory T , written $T \models X$, if $w \models T$ implies $w \models X$. If $T = \{E\}$, then we write $E \models X$ for $\{E\} \models X$. For two theories T and S , we say that $T \models S$ if $T \models X$ for all $X \in S$. We say that T and S are (logically) equivalent, written $T \sim S$, if $T \models S$ and $S \models T$. In order to avoid confusion, we will throughout denote the empty clause by $\{\}$, and the empty theory by $\emptyset$. A theory T is (logically) closed if $T \models X$ implies $X \in T$ for all clauses X . It is called consistent if $T \not\models \{\}$.

Rounds and Zhang originally set out to characterize logically the notion of *Smyth powerdomain* of coherent algebraic cpos. It naturally lead to the clausal logic RZ from Definition 2.1.

We can provide a sound and complete proof theory with simple rules. Consider the following rule, which we call *simplified hyperresolution*.

$$\frac{X_1 \quad X_2; \quad a_i \in X_i}{\text{mub}\{a_1, a_2\} \cup (X_1 \setminus \{a_1\}) \cup (X_2 \setminus \{a_2\})} \quad (\text{shr})$$

Also consider the following rules, which we call the *reduction rule* and the *extension rule*.

$$\frac{X; \quad \{a, y\} \subseteq X; \quad y \sqsubseteq a}{X \setminus \{a\}} \quad (\text{red}),$$

$$\frac{X; \quad y \in K(D)}{\{y\} \cup X} \quad (\text{ext})$$

Theorem 2.2 *The system consisting of (shr), (ext) and (red) is complete.*

For derivations with the rules from above, we use the symbol $\vdash$ as usual.

3 A Resolution Theorem

We simplify proof search via resolution by requiring stronger conditions on the domain.

Definition 3.1 *An atomic domain is a coherent algebraic cpo D with the following property: For all $c \in K(D)$, the set $A(c) = \{p \in A(D) \mid p \sqsubseteq c\}$ is finite and $c = \bigsqcup A(c)$.*

We seek to represent a clause X by a finite set $A(X)$ of atomic clauses which is logically equivalent to X . Given $X = \{a_1, \dots, a_n\}$, we define $A(X) = \{\{b_1, \dots, b_n\} \mid b_i \in A(a_i) \text{ for all } i = 1, \dots, n\}$.

Theorem 3.2 *For any clause X we have $A(X) \sim \{X\}$.*

In view of Theorem 3.2, it suffices to study $T \vdash X$ for theories T and atomic clauses X . We can actually obtain a stronger result, as follows, which provides some kind of normal forms of derivations. For a theory T , define $A(T) = \{A(X) \mid X \in T\}$.

Theorem 3.3 *Let D be an atomic domain, T be a theory, X be a clause and*

$$T \vdash T_1 \vdash \dots \vdash T_N \vdash X$$

be a derivation. Then there exists a derivation

$$A(T) \vdash A(T_1) \vdash \dots \vdash A(T_N) \vdash A(X)$$

using only the atomic extension rule

$$\frac{X; \quad y \in A(D)}{\{y\} \cup X} \quad (\text{axt})$$

and the multiple atomic shift rule (mas), as follows.

$$\frac{a_i \in X_i; \quad \text{mub}\{a_i \mid i \leq n\} = \{x_j \mid j \leq m\}; \quad b_i \in A(x_i)}{\{b_1, \dots, b_m\} \cup \bigcup_{i \leq n} (X_i \setminus \{a_i\})}$$

It also turns out that the rules from Theorem 3.3 are sound. We investigate next a notion of negation on domains.

Definition 3.4 *An atomic domain is called an atomic domain with negation if there exists an involutive and Scott-continuous negation function $\neg : D \rightarrow D$ with the properties:*

- (i) $\neg$ maps $A(D)$ onto $A(D)$.
- (ii) For all $p, q \in A(D)$ we have $p \nabla q$ if and only if $q = \neg p$.
- (iii) For every finite subset $A \subseteq A(D)$ such that $p \uparrow q$ for all $p, q \in A$, the supremum $\bigsqcup A$ exists.

Proposition 3.5 *Let D be an atomic domain with negation. Then for all $c \in K(D)$ we have $\neg c = \bigsqcup \{\neg a \mid a \in A(c)\}$.*

The following result, an analogue to the resolution theorem mentioned in the introduction, allows one to replace the search for derivations by search for contradiction.

Theorem 3.6 *Let D be an atomic domain with negation. Let T be a theory and X be an atomic clause. Then $T \models X$ if and only if $T \cup \{\{\neg a\} \mid a \in X\} \vdash \{\}$.*

On atomic domains with negation, we can therefore establish the following sound and complete proof principle.

Theorem 3.7 *Let T be a theory and X be a clause. Consider $T' = A(T)$. For every atomic clause $A \in A(X)$ attempt to show $T' \cup \{\{\neg a\} \mid a \in A\} \vdash \{\}$ using (axt) and (mas). If this succeeds, then $T \models X$. Conversely, if $T \models X$ then there exists a derivation $T' \cup \{\{\neg a\} \mid a \in A\} \vdash \{\}$ for each $A \in A(X)$ using only the above mentioned rules.*

4 Conclusions

We have seen that for certain domains logical consequence in the logic RZ can be reduced to search for contradiction, a result which yields a proof mechanism similar to that underlying the resolution principle used in resolution-based logic programming systems. The result should be understood as foundational for establishing logic programming systems on hierarchical knowledge, built on a firm domain-theoretic background. Further research is being undertaken to substantiate this.

References

- [Abramsky and Jung, 1994] Samson Abramsky and Achim Jung. Domain theory. In Samson Abramsky, Dov Gabbay, and Thomas S.E. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 3. Clarendon, Oxford, 1994.
- [Hitzler and Seda, 200x] Pascal Hitzler and Anthony K. Seda. Generalized metrics and uniquely determined logic programs. *Theoretical Computer Science*. To appear.
- [Robinson, 1965] J. Allan Robinson. A machine-oriented logic based on the resolution principle. *Journal of the ACM*, 12(1):23–41, 1965.
- [Rounds and Zhang, 2001] William C. Rounds and Guo-Qiang Zhang. Clausal logic and logic programming in algebraic domains. *Information and Computation*, 171(2):156–182, 2001.

A Novel Framework for Integrating Discrete Event System Control and Diagnosis

Gregory Provan*

Rockwell Scientific Company
1049 Camino Dos Rios, Thousand Oaks, CA 91360

1 Towards an Integrated Control/Diagnosis Framework

Diagnosis of discrete event systems (DES) is an important task, with several desiderata that include the ability to (1) encode both control and diagnosis properties within a single representation, and (2) to compose system models from component models in a simple and efficient manner, e.g., without the kind of state explosion that can occur during parallel composition of component Finite State Machines (FSMs).

Many formalisms have been proposed for the Model-Based Diagnosis (MBD) of a DES, but each has drawbacks. The representations based on FSMs, e.g., [Sampath *et al.*, 1995], have a clear control semantics but suffer from incompletely-specified diagnostic semantics and state explosion during model composition. MBD representations, e.g., [Darwiche, 1998], have been used to model DESs using space-efficient compositional methods [Darwiche and Provan, 1996], but lack well-developed control-theoretic specifications.

This article proposes a framework based on FSMs that has a clear control and diagnostics semantics, yet enables model composition without state-space explosions. We propose a two-level modeling framework, based on a system hypergraph $\mathcal{H}$ that can serve as a generator for MBD and FSM models. $\mathcal{H}$ specifies the component causal relations using the graphical framework described within causal networks (CN) [Darwiche, 1998], whereby we can clearly specify independence relations for system variables, and as a consequence compose system models with system state spaces limited through independence relations. Our representation is not significantly more complex than that of either FSM or CN alone, yet is equivalent to both these representations. We introduce to MBD modeling additional requirements for specifying control transitions, and to FSM modeling requirements on state descriptions and causal independence of components.

2 A Discrete Event Modeling Framework

We adopt a representation that is a hybrid of two key modeling approaches, an MBD approach, Causal Networks (CN) [Darwiche, 1998], and an FSM control/diagnosis approach [Sampath *et al.*, 1995].

We propose a two-level, component-based framework to model complex discrete event systems.¹ Our higher level system representation, termed System Causal Graph (SCG), provides a well-defined approach (based on bond graphs) for describing the high level component (or sub-system) configurations and the physical/causal inter-relations of the components (or sub-systems). We model the behaviors of individual components using a lower level representation, called a Control Causal Network (CCN). The CCN models can be converted into regular Finite State Machines or Propositional Logics for further analysis using FSM-based discrete event system techniques (e.g., supervisory control theory) or model-based reasoning techniques (e.g., CN diagnostics), respectively [Provan and Chen, 2003]. The behaviors of the complete system are described by (the composition of) the individual component models.

2.1 Component-Based Modeling Approach

This section describes the CCN framework we adopt for modeling the components of a system. This representation adopts the CN graphical structure and diagnostics specifications, and encodes propositional equations for each node in the graph based on an extension of FSMs. To explain our framework, we first briefly introduce CNs and FSMs.

We adopt the CN model specification [Darwiche, 1998], called a *system description* Φ , which defines a tuple $(P, \mathcal{G}, \Psi)$, where (a) P is a finite set of discrete-valued variables comprising two disjoint variable types: P_m represents the failure modes of the components, and $P_{\bar{m}}$ represents system properties other than failure modes; (b) $\mathcal{G}$ is a directed acyclic graph that defines the causal relations over the variables in P ; (c) Ψ is a set of propositional sentences, the *domain axioms*, constructed from members in P . This approach has a number of important properties, such as compositional modeling based on the independence properties specified in $\mathcal{G}$: if component i has behavior equations Ψ_i , then the system behavior is specified simply by the union of the component equations, i.e., $\Psi = \cup_i \Psi_i$ [Darwiche, 1998]. Specifications of, and algorithms for computing a diagnosis, minimal diagnosis, etc., are all well-defined [Darwiche, 1998].

An FSM is defined as $G = (Q, \Sigma, \delta, q_0)$, where Q is the state space, Σ is the set of events, δ is the partial transi-

*email: gprovan@rwsco.com

¹Our approach can be easily expanded into a multi-level nested hierarchical structure to accommodate more complex systems.

tion function $\delta : Q \times \Sigma \rightarrow Q$ (and defines the transitions between states in Q), and q_0 is the initial state of the system. This framework has been extended for failure diagnosis through the use of unobservable failure transitions and auto-construction of an observer G_{diag} [Sampath *et al.*, 1995].

In our hybrid representation, we model each component using a control causal network (CCN) consisting of a tuple $(P, \{\Sigma, Q, \Gamma, \delta, (q_0, p_0), Q_m\}, \mathcal{G}, \Psi)$, where P is the set of parameters, Σ is the event set, Q the state set, Γ the set of guards (or pre-conditions), $\delta : \Sigma \times Q \times \Gamma \times P \rightarrow Q \times P$ the transition function, q_0 the initial state, p_0 the initial value of parameters, Q_m the marked states, $\mathcal{G}$ a graph, and Ψ a set of propositional equations.² Parameters in the parameter set P are discrete-valued, and guards $\gamma \in \Gamma$ are predicates on the parameters in P . We can view δ as a set of transitions. An equation describing transition $\tau \in \delta$, denoted $\gamma_\tau \wedge \sigma_\tau \Rightarrow f_\tau(p)$, can be interpreted as follows: If at state q the guard γ_τ is true and the event σ_τ occurs, then the next state is q' and the parameters at q' will be updated to $f_\tau(p)$. We call the functions $f_\tau(p)$ the *actions* of the transition τ . The set of all sequences of event labels (traces) of a CCN given initial parameter setting p_0 is the language L it generates. We have shown in [Provan and Chen, 2003] that this representation possesses the control-theoretic properties of the finite state machine with parameters representation, and is based on standard propositional logic with well-known semantics.

2.2 System Causal Graph

An SCG $\mathcal{H} : (\mathcal{N}, \mathcal{E})$ is a directed graph whose nodes in $\mathcal{N}$ represent the components of the system and whose edges in $\mathcal{E} : \mathcal{N} \times \mathcal{N}$ denote the causal relations between the components. A node $N_i \in \mathcal{N}$ is associated with a CCN component model A_i , which represents the component's behaviors.

The SCG's graph $\mathcal{H}$ has properties similar to those of the graph $\mathcal{G}$ of a CCN. $\mathcal{H}$ must preserve a number of control and simulation/diagnostics properties.³ As an example, in order to prevent possible ambiguity and direct circularity in the causal semantics, we assume that the system represented in an SCG should be structured such that any component in the SCG may not be *both* the (direct) predecessor *and* the (direct) successor of the same component in the SCG.

3 Properties of Integrated Representation

In the full paper [Provan and Chen, 2003], we prove a number of important properties of the SCG. One key property involves the sound FSM and CN models that can be generated, where by sound we mean that the model obeys the syntactic and semantic requirements of the (FSM or CN) representation in question:

²The definition is a discrete-parameter restriction of Finite State Machines with Parameters [Chen and Lin, 2000], such that we use a logical representation for state transitions. Our representation extends the representation and control semantics of the causal network models defined in [Darwiche and Provan, 1996] by adding an explicit transition framework for each CN sentence in Ψ .

³The full paper, [Provan and Chen, 2003], discusses $\mathcal{H}$ and its properties in detail.

Lemma 1 *Mapping an SCG into an FSM (CN) produces a sound system-level FSM (CN), respectively.*

To show the CN model created by the above procedure is equivalent to the FSM model, we show that they generate the same language.

Theorem 1 *Given an FSM G and a CN model Φ generated from an SCG with initial state-setting (q_0, p_0) , the FSM and CN languages are equal, i.e., $L(G, q_0) = L(\Phi, p_0)$.*

From a control perspective, this means that the FSM and CN possess the same control properties, e.g., liveness, correctness, etc. From a plant modeling perspective, we can use the logical representation of the SCG to validate a plant model using a theorem prover.

In addition, [Provan and Chen, 2003] shows the equivalence of diagnostic capabilities of the generated models. We assume that, to perform fault isolation, for the FSM (CN) approach we use a diagnoser G_{diag} (CN model Φ), respectively.

Theorem 2 *If we map an SCG into a FSM G and a CN model Φ , then: (a) Given observation set Θ , G is diagnosable (i.e., we can isolate failure events) iff Φ is diagnosable; (b) Given observation set Θ , the set of diagnoses for G is equivalent to the set of diagnoses for Φ .*

This result means that we can now establish equivalent diagnostic capabilities between an FSM and a CN representation, if they can be encoded by the same SCG. Hence, the important MBD properties, e.g., completeness and soundness of diagnostics given a model Φ , can be inherited by an appropriate FSM model.

In summary, these properties indicate that this approach acquires the control semantics of FSMs, and the diagnostic semantics of CNs. On the practical side, although we can generate and use algorithms for either FSM or CN models, further research is necessary to identify the model type most appropriate to a particular class of applications, since model/algorithm efficiency is domain-dependent. In addition, we need to analyze the tradeoffs associated with the extra modeling requirements of the SCG graphical framework versus the linear (rather than worst-case exponential, as in an FSM) growth associated with system model composition.

References

- [Darwiche and Provan, 1996] A. Darwiche and G. Provan. Exploiting system structure in model-based diagnosis of discrete event systems. In *Proc. 7th Intl. Workshop on Principles of Diagnosis*, pages 95–105, 1996.
- [Darwiche, 1998] Adnan Darwiche. Model-based diagnosis using structured system descriptions. *Journal of Artificial Intelligence Research*, 8:165–222, 1998.
- [Chen and Lin, 2000] Y.-L. Chen and F. Lin. Modeling of Discrete Event Systems using Finite State Machines with Parameters. In *Proc. IEEE Conf on Control Applications*, pages 941–946, Anchorage, AK, September 2000.
- [Provan and Chen, 2003] G. Provan and Y.-L. Chen. A general modeling framework for model-based reasoning and discrete event systems analysis. Technical Report SCTR-03-21, Rockwell Scientific Company, March 2003.
- [Sampath *et al.*, 1995] M. Sampath, R. Sangupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis. Diagnosability of discrete event systems. *IEEE Transactions on Automatic Control*, 40(9):1555–1575, September 1995.

Assertion Application in Theorem Proving and Proof Planning

Quoc Bao Vo¹, Christoph Benzmüller¹ and Serge Autexier²

¹FR Informatik, Universität des Saarlandes, 66041 Saarbrücken, Germany

{bao|chris}@ags.uni-sb.de

²DFKI GmbH, 66123 Saarbrücken, Germany

autexier@dfki.de

Our work addresses assertion retrieval and application in theorem proving systems or proof planning systems for classical first-order logic. We propose a distributed mediator M between a mathematical knowledge base KB and a theorem proving system TP which is independent of the particular proof and knowledge representation formats of TP and KB and which applies generalized resolution in order to analyze the logical consequences of arbitrary assertions for a proof context at hand. We discuss the connection to proof planning and motivate an application in a project aiming at a tutorial dialogue system for mathematics. This paper is a short version of [9].

1 Proof planning at the assertion level

Due to Huang [6], the notion of assertion comprises mathematical knowledge from a mathematical knowledge base KB such as axioms, definitions, and theorems. Huang argues that an assertion-based representation, i.e. *assertion level*, is just the right level for machine generated proofs to be transformed into before being presented to (human) users. In this paper we argue further that the assertion level can also serve as one of those levels of granularity on which knowledge-based proof planning should be based [7].

We are convinced that by planning directly on the assertion level it will be possible to overcome at least some of the identified limitations and problems of proof planning as discussed in [3; 5] — in particular, those, that are caused by an unfortunate intertwining of proof planning and calculus level theorem proving. The perspective we therefore motivate is to consider the assertion level as a well chosen borderline between proof planning and machine oriented methods. Determining the logical consequences of assertions in a proof context is the task of machine oriented methods (in our case generalized resolution). The tasks on top of this level — for instance, an domain dependent containment of the initial assertions to be considered, the heuristic selection of the most promising among the computed logical consequences, the introduction, constraining and handling of *meta-variables*, etc. — belong to the scope of domain specific proof planning.

In summary, instead of reconstructing natural deduction (ND) proofs to obtain assertion level proofs as suggested by Huang, we propose to directly plan for proofs at the assertion level. This should improve the quality of the resulted proof plans and also facilitate better user interaction.

The development of our ideas revolves around the mathematical assistant system OMEGA [8] and the current initia-

tive in this project to rebuild the system on top of the proof representation framework in [1; 2]. We furthermore employ OMEGA's agent-based search mechanism OANTS [4] for a distributed modeling of our framework and we motivate an application of the approach in a project aiming at a tutorial dialogue system for mathematics.

2 Assertion Application via Generalized Resolution

Depending on the proof context there may be several ways in which an assertion can be used. For instance, the assertion

$$\forall S_1, S_2 : Set. (S_1 \subseteq S_2 \Leftrightarrow \forall x : Element. (x \in S_1 \Rightarrow x \in S_2))$$

allows us to derive: (1) $a \in V$ from $a \in U$ and $U \subseteq V$, (2) $U \not\subseteq V$ from $a \in U$ and $a \notin V$, (3) $\forall x : Element. (x \in U \Rightarrow x \in V)$ from $U \subseteq V$, (4) etc.

Traditional theorem provers or proof planners that operate on calculus level can only achieve such conclusions after a number of proof steps to eliminate the quantifiers and other connectives such as implication and conjunction. We propose an algorithm *Assertion Application* based on a generalized form of resolution in order to achieve such conclusions in one step for arbitrary assertions and proof contexts at hand. The algorithm, which is described in [9], is based on:

1. The representation of formulas as signed formula trees (SFT); see also [1; 2]. This way we achieve access to the literals of the formulas without breaking them apart.
2. The lifting of the resolution idea to directly operate on complementary and unifiable literals in SFTs instead of working on (unintuitive) normal forms.
3. The exhaustive application of (2) to the SFTs generated for the assertion and the formulas of the proof context.

3 Modeling Assertion Agents

In our implementation of mediator M we model assertion application in form of distributed search processes employing the OANTS approach [4]. This agent based formalism is the driving force behind distributed proof search in OMEGA.

The general application scenario we have in mind is a theorem prover TP that is connected to an independent mathematical knowledge base KB. TP is focusing a proof task τ consisting of the formula TP has to prove, viz. the theorem H ,

and a set of assumptions it can use to prove H . TP can determine the relevant parts, i.e. the related theories Th , from KB and hand them over to our assertion module M . The task of M is to compute with respect to proof task τ all possible logical consequences of the available assertions $\mathcal{B}_i$ taken from Th .

We propose to create for each assertion $\mathcal{B}_i$ one associated instance $\mathcal{AG}_{\mathcal{B}_i}$ of a generic *assertion agent* $\mathcal{AG}$. The generic assertion agent $\mathcal{AG}$ is based on our algorithm *AssertionApplication*. Note that this algorithm only depends on the SFT of the focused assertion and a further set of SFTs for the proof context, and both are specified as parameters of *AssertionApplication*. Each assertion agent instance $\mathcal{AG}_{\mathcal{B}_i}$ computes and suggest the logical consequences of $\mathcal{B}_i$ for proof task τ to our module M which passes them further to TP.

Depending on the size of KB there could be too many applicable assertions passed to M and also too many ways an assertion can be applied. One possibility is to restrict the search space by imposing prerequisites for assertion retrieval. For instance, a proof planner may employ its domain specific meta knowledge to formulate and pass respective context sensitive syntactical filter criteria to the mediator for an efficient preselection by syntactic means.

Proof planning, however, has developed more sophisticated ways to guide and constrain possible the possible instantiations and applications of assertions. The investigation on how these techniques can optimally be employed on top of our assertion application module M are further work.

4 An application: The DIALOG project

Our approach to assertion application is motivated by an application in the DIALOG project as part of the Collaborative Research Center on *Resource adaptive cognitive processes* at Saarland University. The goal of this research project is (i) to empirically investigate flexible dialogue management strategies in complex mathematical tutoring dialogues, and (ii) to develop an experimental prototype system gradually embodying the empirical findings. The experimental system will engage in a dialogue in natural language (and perhaps other modes of communication) and help a student to understand and produce mathematical proofs. It is important that such a system is supported by a human oriented mathematical proof development environment and the OMEGA system with its advanced proof presentation and proof planning facilities is a suitable answer to this requirement.

The overall scenario for DIALOG is: A student user is first taking an interactive course on some mathematical domain (e.g., naive set theory) within a web-based learning environment. When finishing some sections the student is asked by the system to test his learning progress by actively applying the studied lesson material by performing an interactive proof exercise. Since the learning environment is equipped with user monitoring and modeling facilities a user model is maintained and dynamically updated containing information on the axioms, definitions, and theorems (hence the assertions) the student has studied so far. Also a tutor model is available for each exercise containing information on the mathematical material that should be employed and tested by it.

In this scenario we expect the mathematical assistant sys-

tem to be capable of (i) stepwise-interactive and/or (ii) automated proof construction at a human oriented level of granularity for the proof exercise at hand using exactly the mathematical information specified in the (a) tutor model or (b) user model. The proofs constructed for (a) reflect what we want to teach and the proofs for (b) what the system expects the user to be capable of. For interactive tutorial dialog the support for a stepwise proof construction with the mathematical assistant system is of course important, while fully automatically generated proofs are needed to be able to also give away complete solutions or to initially generate a discourse structure for the dialog on the chosen exercise. We want to stress that the user model may be updated also during an exercise, hence the set of relevant assertions may dynamically change during an interactive session.

It is easy to motivate the design of our assertion application module for this scenario. Its capabilities for assertion application for a dynamically varying set of assertions are crucial for the project. It is also essential that reasoning is facilitated at a human oriented level of granularity, since we do not want the user to puzzle around with the peculiarities of, for instance, logical derivations in sequent or natural deduction calculus.

5 Conclusion

Mediating mathematical knowledge between (web-based) mathematical knowledge bases, mathematical reasoning systems and human users is a mathematical knowledge management task of increasing importance. We sketched a flexible and distributed solution for the subtasks of suggesting the logical consequences of assertions at an intuitive reasoning level.

References

- [1] S. Autexier. A proof-planning framework with explicit abstractions based on indexed formulas. *ENTCS*, 58(2), 2001.
- [2] S. Autexier. Hierarchical Contextual Reasoning. PhD Thesis, Saarland University, forthcoming.
- [3] C. Benzmüller, A. Meier, E. Melis, M. Pollet, and V. Sorge. Proof planning: A fresh start? In *Proc. of the IJCAR 2001 Workshop: Future Directions in Automated Reasoning*, Siena, Italy, 2001.
- [4] C. Benzmüller and V. Sorge. OANTS – an open approach at combining interactive and automated theorem proving. In M. Kerber and M. Kohlhase, editors, *Symbolic Computation and Automated Reasoning*, A.K.Peters, 2000.
- [5] A. Bundy. A critique of proof planning. In A. C. Kakas and F. Sadri, editors, *Computational Logic. Logic Programming and Beyond*, LNCS 2408, Springer, 2002.
- [6] X. Huang. Reconstructing proofs at the assertion level. newblock In A. Bundy, editor, *Proc. CADE-12*, LNAI 814, Springer, 1994.
- [7] E. Melis and J. Siekmann. Knowledge-based proof planning. *Artificial Intelligence Journal*, 115(1):65–105, 1999.
- [8] J. Siekmann et al. Proof development with OMEGA. In A. Voronkov, editor, *Proc. CADE-18*, LNAI 2392, Springer, 2002.
- [9] Q. B. Vo, C. Benzmüller, and S. Autexier. An approach to assertion application via generalized resolution. Seki Report SR-03-01, Saarland University, February 2003. <http://www.ags.uni-sb.de/~omega/pub/postscript/SR-03-01.ps.gz>

POSTER PAPERS

CASE-BASED REASONING

Case Base Adaptation Using Solution-Space Metrics^{*}

Brian Knight
University of Greenwich
School of Computing & Mathematical Sciences
London, SE10 9LS, UK
b.knight@gre.ac.uk

Fei Ling Woon
Tunku Abdul Rahman College
School of Arts & Science
Kuala Lumpur, Malaysia
f.woon@gre.ac.uk

Abstract

In this paper we propose a generalisation of the k-nearest neighbour (k-NN) retrieval method based on an error function using distance metrics in the solution and problem space. It is an interpolative method which is proposed to be effective for sparse case bases. The method applies equally to nominal, continuous and mixed domains, and does not depend upon an embedding n-dimensional space. In continuous Euclidean problem domains, the method is shown to be a generalisation of the Shepard's Interpolation method. We term the retrieval algorithm the *Generalised Shepard Nearest Neighbour* (GSNN) method. A novel aspect of GSNN is that it provides a general method for interpolation over nominal solution domains. The performance of the retrieval method is examined with reference to the Iris classification problem, and to a simulated sparse nominal value test problem. The introduction of a solution-space metric is shown to out-perform conventional nearest neighbours methods on sparse case bases.

1 Introduction

We present in this study a Case-Based Reasoning (CBR) retrieval method that utilises a distance metric imposed on solution space. The motivation for such a method is to extend a powerful interpolative method, already proven in the real domain, so that it applies equally in the domain of nominal values. Interpolative methods are well studied in the real domain, and can give good results from relatively sparse datasets. However, no general interpolative method exists for nominal (discrete) solution domains.

2 An Error Function

The GSNN method will be applied to a general class of problem and solution domains. A distance metric $d_x(x, x_i)$ is here defined on the problem domain X and $d_y(y, y_i)$ on the solution domain Y . For the problem space, the term $\|x - x_i\|$ in the Shepard's method [1968] is generalised to $d_x(x, x_i)$ over X . For the solution space Y , $d_y(y, y_i)$ is used where $y = f(x)$ is the value of y which minimizes the error function:

$$I(y) = \sum_i^n \|y - y_i\|^2 \|x - x_i\|^{-p} / \sum_i^n \|x - x_i\|^{-p}$$
$$\partial I / \partial y = 0 \quad \text{iff} \quad y \sum_i^n \|x - x_i\|^{-p} = \sum_i^n \|x - x_i\|^{-p} y_i$$

That this is a minimum follows from the positive definite form: $\partial^2 I / \partial y^2 = \sum_i^n \|x - x_i\|^{-p} / \sum_i^n \|x - x_i\|^{-p}$

The function $I(y)$ depends only upon the Euclidean distance over $Y = R$ and $X = R^d$. In order to generalise the method completely, we propose the error function:

$$I(y) = \sum_i^k d_y^2(y, y_i) d_x(x, x_i)^{-p} / \sum_i^k d_x(x, x_i)^{-p} \quad (1)$$

Here, the set $\{x_1, x_2, \dots, x_k\}$ are the k nearest neighbours in the problem space to the point x . $d_x(x, x_i)$ and $d_y(y, y_i)$ are distance on domains $x \in X, y \in Y$. The retrieved value y is the value $y \in Y$ which minimizes the error function I . The GSNN algorithm is given as follows:

$$\hat{f}(x_q) \leftarrow \arg \min_{y \in Y} \sum_{i=1}^k w_i d_y^2(y, f(x_i)) \quad \text{and} \quad w_i = d_x(x_q, x_i)^{-p}.$$

3 Illustrative Example

In this example we illustrate in detail how the method works. We choose the Iris data set [Fisher, 1936]. The problem space is X and $x = (x_1, x_2, x_3, x_4)$ is a point in X . The solution space $Y = \{\text{setosa, versicolour, virginica}\}$. For the problem space we define distance according to a weighted sum of attributes. For the Y space, we define $d_y(y, y')$ by using the distances between cluster centres to represent the distance between the classes. These distances are shown in the following matrix:

^{*} The support by the University of Greenwich and Tunku Abdul Rahman College (TARC) is acknowledged.

	setosa	versicolour	virginica
setosa	0	.35	.49
versicolour	.35	0	.18
virginica	.49	.18	0

We take two cases, one from *setosa* and one from *virginica*:

$$x_1 = (4.4, 2.9, 1.4, 0.2), y_1 = \text{setosa}$$

$$x_2 = (7.2, 3.2, 6, 1.8), y_2 = \text{virginica}$$

We take as target the *versicolour* iris:

$$x = (5.5, 2.3, 4, 1.3), y = ?$$

Taking $p=1$ and $k=2$, the function $I(y)$ is:

$$I(y) = \sum_i^2 d_y^2(y, y_i) d_x(x, x_i)^{-1} / \sum_i^2 d_x^2(x, x_i)^{-1}$$

$$= (0.36)^{-1} d_y(y, \text{setosa})^2 + (0.35)^{-1} d_y(y, \text{virginica})^2 / (0.36)^{-1} + (0.35)^{-1}$$

Since $I(\text{versicolour})$ is minimum, we take $y=\text{versicolour}$ as the estimated value. This example shows an advantage of the interpolation method in situations where cases are sparse, in that it can correctly predict nominal values not represented in the case base itself.

4 Test on a Simulated Case Base

To examine how the GSNN method might work on real case bases, we simulated case bases of varying density and structure, and used the method to estimate simulated target sets. As a basis for the simulation, we adapted the function used by Ramos and Enright [2001] (i.e. $y = \text{Int}(10 \sin 2\pi x_1 * \sin 2\pi x_2)$) to give 21 nominal values, $y_1, \dots, y_{21}$. These 21 nominal values inherited a distance metric from the numeric values: $d(y_i, y_j) = |y_i - y_j|$.

Test 6.1 uses regularly spaced cases at various case densities. This might represent a well organised case base. Test 6.2 uses randomly selected cases, and is intended to represent disorganised sparse case bases. Cases (x_1, x_2, y) are constructed in the domain: $0 \leq x_1, x_2 \leq 1$, over a regular square lattice, with $10^2, 20^2, 30^2$ points.

Size	Methods	k=1	k=2	k=3	k=4
100	GSNN	709	501	453	539
	k-NN		769	799	786
	DWNN		710	706	709
400	GSNN	501	308	188	215
	k-NN		616	684	652
	DWNN		499	478	488
900	GSNN	360	251	181	224
	k-NN		450	471	456
	DWNN		360	344	344

Table 1. Errors in estimating a test set of 1000 targets, for regular case bases.

Table 1 shows the result of Test 6.1. These results confirm that GSNN with $k > 1$ can out-perform both k-NN and DWNN [Mitchell, 1997] for case bases with regular structure. Table 2 shows the results of Test 6.2. The results show that more errors are recorded for random case bases than for

regular case bases of equivalent size, whatever the value of k . Once again, the results show that GSNN out-performed the other nearest neighbour methods.

Size	Methods	k=1	k=2	k=3	k=4
100	GSNN	734	663	653	678
	k-NN		772	843	843
	DWNN		733	737	739
400	GSNN	573	506	511	492
	k-NN		643	695	708
	DWNN		573	583	591
900	GSNN	421	356	359	344
	k-NN		486	547	548
	DWNN		422	435	432

Table 2. Errors in estimating a test set of 1000 targets, for random case bases.

5 Conclusion

In this paper, we have proposed a method for interpolation over nominal values. The method generalises the Shepard's interpolation method by expressing it in terms of the minimization of a function $I(y)$. This function relies only on distance metrics defined over problem and solution spaces. The method has an advantage for CBR in that it is applicable to case bases with nominal values in the problem and solution domain where no natural ordering exists. The examples studied indicate that GSNN could be useful in CBR with a sparse set of cases, and particularly where the cases can be organised. Tests show that GSNN is more efficient as a retrieval engine than other nearest neighbour methods. The inclusion of a solution space metric in the GSNN technique could be useful in two areas of CBR: (i) The selection of an optimum case base, (ii) Case based model building, from experimental or numerical modeling exercises. Investigations using numerical models indicate that GSNN would appear to be a promising approach for the construction of efficient case-based models.

References

- [Fisher, 1936] R. A. Fisher. The Use of Multiple Measurements in Taxonomic Problems. *Annual Eugenics*, 7, Part II, 179-188 (1936).
- [Mitchell, 1997] T. Mitchell. Machine Learning. *McGraw-Hill Series in Computer Science*, WCB/McGraw-Hill, 230 – 247, USA 1997.
- [Ramos and Enright, 2001] G. A. Ramos and W. Enright. Interpolation of Surfaces over Scattered Data. *Visualization, Imaging and Image Processing Conference, VIIP02*. 2001 IASTED
- [Shepard, 1968] D. Shepard. A Two-Dimensional Interpolation Function for Irregularly Spaced Data. *Proceeding of the 23rd National Conference, ACM*, 517-523, 1968.

Coverage-Optimized Retrieval

David McSherry

School of Computing and Information Engineering
University of Ulster, Coleraine BT52 1SA, Northern Ireland
dmg.mcsherry@ulster.ac.uk

Abstract

We present a generalization of similarity-based retrieval in recommender systems which ensures that for any case that is acceptable to the user, the retrieval set contains a case that is at least as good in an objective sense and so also likely to be acceptable. Our approach recognizes that similarity to the target query is only one of several possible criteria according to which a given case might be considered at least as good as another.

1 Introduction

An advantage of case-based reasoning (CBR) in product recommendation is that if none of the available products exactly matches the user's query, she can be shown the products that are most *similar* to her query [Wilke *et al.*, 1998]. However, a known limitation of similarity-based retrieval is that the most similar case may not be the one that is most acceptable to the user [Burkhard, 1998; Smyth and McClave, 2001]. The k -NN strategy of retrieving the k most similar cases only partially compensates for this limitation, as the number of cases that can be presented to the user is necessarily restricted in practice. So the existence of an acceptable case does not guarantee that it will be retrieved.

We present a new approach to retrieval called *coverage-optimized* retrieval (CORE) which ensures that for any case that is acceptable to the user, the retrieval set contains a case that is at least as good in an objective sense and so also likely to be acceptable. Similarity to the target query is only one of several possible criteria according to which a given case might be considered at least as good as another in the approach.

2 The CORE Retrieval Set

The similarity of a given case C to a target query Q over a subset A_Q of the case attributes A is typically defined as:

$$Sim(C, Q) = \sum_{a \in A_Q} w_a sim_a(C, Q)$$

where for each $a \in A_Q$, w_a is a numeric weight representing the importance of a and $sim_a(C, Q)$ is a measure of the similarity of $\pi_a(C)$, the value of a in C , to $\pi_a(Q)$, the preferred value of a . Apart from its similarity to the target query, another factor likely to influence the acceptability of a given case is the *compromises* it involves, or preferences of the user that it fails to satisfy [Burkhard, 1998]. Often in e-commerce domains, one can identify attributes whose values most users would prefer to maximize or minimize [Wilke *et al.*, 1998]. In CORE, we assume that the value specified by the user is a preferred minimum in the case of a *more-is-better* attribute or a preferred maximum in the case of a *less-is-better* attribute.

Definition 1 For any query Q and case C we define:

$$\begin{aligned} d_1(C, Q) &= \{a \in A_Q : \text{nominal}(a), \pi_a(C) \neq \pi_a(Q)\} \\ d_2(C, Q) &= \{a \in A_Q : \text{more-is-better}(a), \pi_a(C) < \pi_a(Q)\} \\ d_3(C, Q) &= \{a \in A_Q : \text{less-is-better}(a), \pi_a(C) > \pi_a(Q)\} \\ d(C, Q) &= \bigcup_{i=1,2,3} d_i(C, Q) \end{aligned}$$

So $d(C, Q)$ is the set of attributes with respect to which C fails to satisfy the user's preferences. Below we define four dominance criteria according to which a given case C_1 might be considered at least as good as another case C_2 :

$$\begin{aligned} D0: Sim(C_1, Q) &\geq Sim(C_2, Q) \\ D1: Sim(C_1, Q) &\geq Sim(C_2, Q) \text{ and } |d(C_1, Q)| \leq |d(C_2, Q)| \\ D2: Sim(C_1, Q) &\geq Sim(C_2, Q) \text{ and } d(C_1, Q) \subseteq d(C_2, Q) \\ D3: sim_a(C_1, Q) &\geq sim_a(C_2, Q) \text{ for all } a \in A_Q \end{aligned}$$

For example, if $sim_a(C_1, Q) \geq sim_a(C_2, Q)$ for all $a \in A_Q$ we say that C_1 dominates C_2 with respect to D3. We say that a given case C_2 is *covered* by a retrieval set RS if $C_2 \in RS$ or there exists $C_1 \in RS$ such that C_1 dominates C_2 . The importance of coverage in this sense is that if an acceptable case that is not retrieved is covered by a retrieved case, then the retrieved case is also likely to be acceptable. Another basic premise in our approach is that the likelihood of the retrieved case also being acceptable increases with the strength of the dominance criterion with respect to which it

dominates the acceptable case. For example, dominance of the acceptable case with respect to D0 (the usual similarity criterion) may not be enough to ensure that the retrieved case is also acceptable. However, if it dominates the acceptable case with respect to D3, then there is no attribute with respect to which it is less similar to the user's query.

In k -NN, all cases are dominated by the most similar case with respect to D0; so even 1-NN provides full coverage of the case library with respect to D0, the weakest of our dominance criteria. The aim in CORE is to construct a retrieval set that provides full coverage of the case library with respect to any of the dominance criteria we have identified. In Figure 1, Q is the target query and *Candidates* is a list of candidate cases for addition to the retrieval set RS . We assume that the candidate cases, initially all cases in the case library, are sorted in order of non-increasing similarity, and that if C_1, C_2 are equally similar cases such that C_1 dominates C_2 but C_2 does not dominate C_1 , then C_1 is listed before C_2 in *Candidates*.

```

algorithm CORE( $Q$ , Candidates,  $RS$ )
begin
   $RS \leftarrow \phi$ 
  while  $|Candidates| > 0$  do
    begin
       $C_1 \leftarrow \text{first}(Candidates)$ 
       $RS \leftarrow \{C_1\} \cup RS$ 
       $\text{cover}(C_1) \leftarrow \{C_1\}$ 
      for all  $C_2 \in \text{rest}(Candidates)$  do
        if  $C_1$  dominates  $C_2$ 
          then  $\text{cover}(C_1) \leftarrow \text{cover}(C_1) \cup \{C_2\}$ 
       $Candidates \leftarrow Candidates - \text{cover}(C_1)$ 
    end
  end

```

Figure 1. Generic algorithm for coverage-optimized retrieval.

We refer to the versions of CORE based on D1, D2 and D3 as CORE-1, CORE-2 and CORE-3 respectively. It is worth noting that with D0 as the dominance criterion, CORE is equivalent to 1-NN, and so CORE is in fact a generalization of similarity-based retrieval.

Theorem 1 *The CORE retrieval set provides full coverage of the case library and no smaller retrieval set can provide full coverage of the case library.*

While it can easily be shown that the maximum possible sizes of the CORE-1 and CORE-2 retrieval sets for a given query Q are $|A_Q| + 1$ and $2^{|A_Q|}$ respectively, the maximum possible size of the CORE-3 retrieval set is not as easily determined. In practice, CORE-1 and CORE-2 retrieval sets are usually much smaller than their maximum possible sizes. Figure 2 shows the maximum, average, and minimum sizes of CORE retrieval sets for full-length queries on the

Travel case library (www.ai-cbr.org). As might be expected, there is a trade-off between the strength of the dominance criterion in terms of which coverage is defined and the size of the retrieval set required to provide full coverage of the case library. Though unable to compete with CORE-1 in terms of coverage efficiency, CORE-2 has an average retrieval-set size of only 7.5 cases.

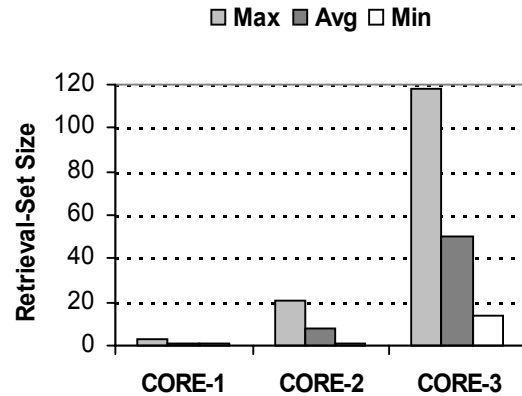


Figure 2. Size of the CORE retrieval set for full-length queries on the Travel case library.

3 Conclusions

Coverage-optimized retrieval (CORE) is a generalization of similarity-based retrieval which ensures that for any case that is acceptable to the user, the retrieval set contains a case that is at least as good according to a given dominance criterion. Our empirical results show that the size of the retrieval set required to provide full coverage of the case library in this sense increases with the strength of the dominance criterion in terms of which coverage is defined. However, CORE-2 offers a good compromise between strength of dominance and retrieval-set size, with an average of only 7.5 cases required to provide full coverage for full-length queries on a case library containing over 1,000 cases.

References

- [Burkhard, 1998] H.-D. Burkhard. Extending some concepts of CBR - foundations of case retrieval nets. In M. Lenz, B. Bartsch-Spörl, H.-D. Burkhard and S. Wess (eds.), *Case-Based Reasoning Technology*, pages 17-50. Berlin-Heidelberg, Springer-Verlag, 1998.
- [Smyth and McClave, 2001] B. Smyth and P. McClave. Similarity vs. diversity. In *Proceedings of the Fourth International Conference on Case-Based Reasoning*, pages 347-361, Vancouver, Canada, 2001. Springer-Verlag.
- [Wilke et al., 1998] W. Wilke, M. Lenz and S. Wess. Intelligent sales support with CBR. In M. Lenz, B. Bartsch-Spörl, H.-D. Burkhard and S. Wess (eds.), *Case-Based Reasoning Technology*, pages 91-113. Berlin-Heidelberg, Springer-Verlag, 1998.

Explicit vs Implicit Profiling - A Case-Study in Electronic Programme Guides

Derry O'Sullivan, Barry Smyth, David Wilson

Smart Media Institute, University College Dublin, Ireland

{dermot.osullivan, barry.smyth, david.wilson}@ucd.ie

Abstract

In this paper, we evaluate the use of implicit interest indicators as the basis for user profiling in the Digital TV domain. Research in more traditional domains, such as Web browsing or Usenet News, indicates that some implicit interest indicators (e.g., read-time and mouse movements) are capable of serving as alternative to explicit profile information such as user ratings. Consequently, the key question we wish to answer relates to the type of implicit indicators that can be identified within the DTV domain and the extent to which they can accurately reflect a user's true preferences.

1 Introduction

Recommender systems combine user profiling, machine learning and information retrieval methods to help users locate information by learning their preferences and making personalized recommendations to individuals [Hammond *et al.*, 1996; Balabanovic and Shoham, 1997; Smyth and Cotter, 2000; Good *et al.*, 1999; Goldberg *et al.*, 1992]. There are two basic profiling strategies. *Explicit profiling* means asking the user direct questions about their preferences, usually in the form of an item rating; e.g. in PTVPlus (www.ptvplus.com), a TV show recommender [Smyth and Cotter, 2001; 2000], users rate individual shows on a five point scale from 'strong dislike' to 'strong like'. Explicit profiling has been used in a range of scenarios, from movies [Resnick *et al.*, 1994] to jobs [Rafter *et al.*, 2000], and the resulting profiles are assumed to be reliable indicators of user preferences. However, users are lazy and do not like responding to explicit requests for feedback. As a result explicit profiles often contain few ratings, which can limit the degree to which they will accurately reflect a user's changing preferences over time.

Implicit profiling methods construct user profiles by inferring user ratings from so-called *interest indicators* based on user's interactions with the system [Claypool *et al.*, 2001; Goecks and Shavlik, 1999]. For example, [Morita and Shinoda, 1994; Konstan *et al.*, 1997; Claypool *et al.*, 2001] report a positive correlation between reading times (for Usenet News and Web pages) and content ratings. [Claypool *et al.*, 2001] shows a correlation between content ratings and both

mouse movements and scrolling although mouse movement was only found to be a useful rating predictor at lower rating levels. Interestingly, [Claypool *et al.*, 2001] finds no correlation between user ratings and the number of clicks to links on a Web page.

In this paper we seek to evaluate the use of implicit profiling in a very different application domain, Digital TV (DTV). DTV users are finding it increasingly difficult to identify relevant TV shows among the hundreds of available channels and recommender systems such as PTVPlus have been forwarded as a potential solution [Smyth and Cotter, 2000]. PTVPlus uses its recommendation engine to generate a set of TV programme recommendations for a target user based on their profiled interests using a combination of collaborative and case-based strategies, and presents these recommendations in the form of a personalized programme guide. So far research in the area of personalisation and recommendation within the DTV domain has focused almost exclusively on the use of explicit profiling. Consequently, the key question we wish to answer relates to the type of implicit interest indicators that can be identified within the DTV domain and the extent to which they can accurately reflect a user's true preferences.

2 Implicit Ratings in the DTV Domain

In this research we use the Físchlár video library system [Lee *et al.*, 2000] developed by the Centre for Digital Video Processing at Dublin City University (www.cdvp.dcu.ie). Físchlár has been fully operational for a number of years and attracts hundreds of users on a regular basis. It is a video library system that allows users to record, playback and browse TV shows through a normal Web browser. Users browse personalized television schedules, provided by the PTVPlus personalization engine [Smyth and Cotter, 2000], to select programmes to be recorded by the system. TV shows are captured in MPEG-1 format and processed in order to index their content for storage and to extract key-frames for content-based browsing. Físchlár users can select programmes to record up to 2 days in advance of broadcast and can access previously recorded programmes for playback or browsing.

As Físchlár takes advantage of the PTVPlus personalized TV listings service, there exists the ability to capture explicit ratings feedback for its user profiles (Figure 1) using the rating icons beside each programme's description. However, unlike PTVPlus, Físchlár users can interact directly with pro-

gramming content, and this provides new profiling opportunities. In particular, the record, playback and browsing actions of users are suitable as implicit interest indicators; e.g., selecting a programme to be recorded is surely a strong potential indicator of interest. Thus, Físchlár's profiling capabilities have been enhanced to capture implicit as well as the explicit profiles, by logging all record, playback and browsing actions (Figure 2).

USER #1762	
Channels	BBC1, BBC2,...
+Progs	Friends, ER, ...
Keywords	Comedy, Soap
Times	PrimeTime

Figure 1: Explicit Profile Example.

USER #1762	
BehaviourPlay	
+Progs	Scrubs, Cheers,...

Figure 2: Implicit Profile Example.

3 Experimental Evaluation

Our aim is to evaluate the usefulness of Físchlár's implicit user profiles relative to its explicit profiles. In particular we wish to determine whether or not these implicit profiles are any more or less accurate at predicting user preferences than the explicit profiles gathered directly from user ratings.

3.1 Test Data

Our experiments draw on 650 live-user profiles from the Físchlár system. In particular we make use of 5 different types of profile:

1. *Explicit* profiles contain explicit ratings that Físchlár users have provided;
2. *Play* profiles contain only those programmes that a user has played back;
3. *Record* profiles contain only those programmes that a user has recorded;
4. *Browse* profiles contain only those profiles that a user has browsed using Físchlár's key-frame browser;
5. *Combined* profiles are a combination of play, record and browse profiles.

3.2 Algorithms

In addition we make use of two different programme recommenders:

1. *CF* is a traditional collaborative filtering recommender as described in [Konstan *et al.*, 1997; O'Sullivan *et al.*, 2002b]; it finds similar users by calculating program overlap between the target user and all others, and then recommends to the target user programs from the most similar users that they have yet not rated.

2. *Sim* is a similarity-based recommender that draws on similarity information mined from user profiles as described in [O'Sullivan *et al.*, 2002b]. It uses data mining techniques to find relationship between items and then harnesses this information both in profile similarity and recommendation ranking (ranking all possible recommendable programmes) to recommend programmes to the user.

Due to space constraints, it is not possible to discuss these recommenders in detail; the interested reader is referred to [O'Sullivan *et al.*, 2002b; 2002a]. Suffice it to say that they each draw on a fundamentally different recommendation strategy, be it collaborative as in the case of CF, or content-based, as in Sim.

3.3 Method

First each profile is randomly split into a training profile and a test profile; this is repeated for a range of different split-ratios for each profile. The training profiles are used as the basis for recommendation; in each case we make 50 programme recommendations for each user. The quality of the resulting recommendations is evaluated with reference to the corresponding test profiles by computing the percentage of recommended programmes that appear in the target test profile. So, for example, we use the training play profiles to generate recommendations that are evaluated with respect to the test play profiles; similarly for record and browse profiles. In this way we can compare the quality of the recommendations generated using implicit play, record, browse and combined indicators to the quality of recommendations generated using explicit ratings profiles.

3.4 Results

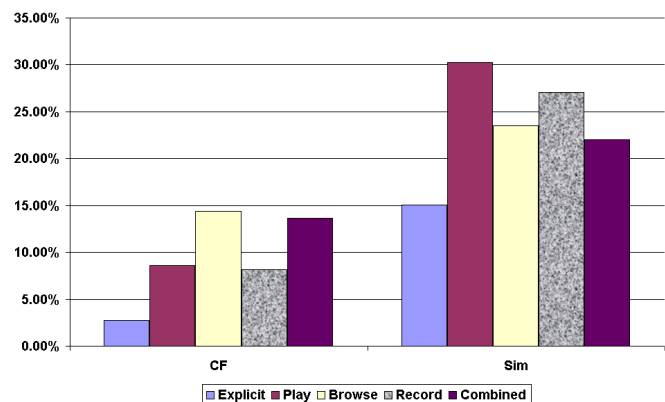


Figure 3: Recommendation accuracy for explicit and implicit (play, record, browse, combined) profile groups.

The results are presented in Figure 3 as an accuracy bar-graph for the various different groups of explicit and implicit profiles (play, record, browse and combined). In each grouping we present the accuracy results obtained by the CF and Sim recommenders. For example, we find that the explicit profiles deliver recommendation accuracies of 2.5% and

15%, for CF and Sim recommenders, respectively; clearly the enhanced Sim recommender offers significant benefits over the standard CF recommender as previously discussed in [O'Sullivan *et al.*, 2002b]. Each of the implicit profiles generate consistently higher recommendation accuracies for both CF and Sim recommenders. For instance, in the case of the play profiles, the recommendation accuracy is 9% and 30% for the CF and Sim recommenders, which is a significant increase on the explicit profile accuracy, especially for the Sim recommender. In fact, the implicit profiles yield recommendation accuracies in excess of 20% for the Sim recommender compared to the 15% obtained for the explicit profiles.

These results indicate that playback, recording, browsing and combined behaviours in Físchlár serve as competent interest indicators when it comes to profiling user preferences. In each case we find an increase in the quality of recommendations made from profiles containing implicit information. Moreover, the scale of this increase is largest in the case of the Sim recommender, due to the fact that this recommender directly mines the profiles in order to generate programme similarity knowledge as the basis for recommendations. The availability of the higher quality implicit profiles helps to improve the similarity rules used by Sim when compared to the similarity rules mined from explicit profiles.

4 Conclusions

We have evaluated the use of implicit interest indicators as the basis for user profiling in the DTV domain. Research in more traditional domains, such as Web browsing or Usenet News, indicates that some implicit interest indicators (e.g., read-time and mouse movements) are capable of serving as reliable alternatives to explicit profile information such as user ratings. Our research indicates that the same is also true in the DTV domain. We have compared the quality of recommendations generated from explicit and implicit profiles and found, through the use of a deployed video library system, that in each case the implicit profiles are capable of generating recommendations that are as good as those generated by the explicit profiles.

Acknowledgements

The support of the Informatics Research Initiative of Enterprise Ireland is gratefully acknowledged. The support and advice of the Centre for Digital Video Processing at Dublin City University is also gratefully acknowledged.

References

- [Balabanovic and Shoham, 1997] M. Balabanovic and Y. Shoham. Fab: Content-Based, Collaborative Recommendation. *Communications of the ACM*, 40(3):66–72, March 1997.
- [Claypool *et al.*, 2001] M. Claypool, P. Le, M. Wased, and D. Brown. Implicit Interest Indicators. In *Intelligent User Interfaces*, pages 33–40, 2001.
- [Goecks and Shavlik, 1999] J. Goecks and J. Shavlik. Automatically Labelling Web Pages based on Normal User Interactions. In *Proceedings of the IJCAI Workshop on Machine Learning for Information Filtering*, Stockholm, Sweden, July 1999.
- [Goldberg *et al.*, 1992] D. Goldberg, D. Nichols, B. Oki, and D. Terry. Using Collaborative Filtering to Weave an Information Tapestry. *Communications of the ACM*, 35(12):61–70, December 1992.
- [Good *et al.*, 1999] N. Good, J. Schafer, J. Konstan, A. Borchers, B. Sarwar, J. Herlocker, and J. Riedl. Combining Collaborative Filtering with Personal Agents for Better Recommendations. In *Proceedings of the Conference of the American Association of Artificial Intelligence (AAAI99)*, Orlando, Florida, USA, July 1999.
- [Hammond *et al.*, 1996] K. Hammond, R. Burke, and K. Schmitt. A Case-Based Approach to Knowledge Navigation. In D.B Leake, editor, *Case-Based Reasoning Experiences Lessons and Future Directions*, pages 125–136. MIT Press, 1996.
- [Konstan *et al.*, 1997] J.A. Konstan, B.N. Miller, D. Maltz, J.L. Herlocker, L.R. Gordon, and J. Riedl. Grouplens: Applying Collaborative Filtering to Usenet News. *Communications of the ACM*, 40(3):77–87, 1997.
- [Lee *et al.*, 2000] H. Lee, A. Smeaton, C. O'Toole, N. Murphy, S. Marlow, and N. O'Connor. The Físchlár Digital Video Recording, Analysis and Browsing System. In *Proceedings of RIAO'2000*, pages 1390–1399, 2000.
- [Morita and Shinoda, 1994] M. Morita and Y. Shinoda. Information Filtering Based on User Behaviour Analysis and Best Match Text Retrieval. In *Proceedings of the 17th ACM Annual International Conference on Research and Development in Information Retrieval*, pages 272–281, 1994.
- [O'Sullivan *et al.*, 2002a] D. O'Sullivan, D. Wilson, and B. Smyth. Improving Case-Based Recommendation: A Collaborative Filtering Approach. In S. Craw and A. Preece, editors, *Proceedings of the 6th European Conference on Case Based Reasoning (ECCBR-02)*, pages 278–291, 2002.
- [O'Sullivan *et al.*, 2002b] D. O'Sullivan, D. Wilson, and B. Smyth. Using Collaborative Filtering Data in Case-Based Recommendation. In *Proceedings of the 15th International FLAIRS Conference*, pages 121–125, 2002.
- [Rafter *et al.*, 2000] R. Rafter, K. Bradley, and B. Smyth. Personalised Retrieval for Online Recruitment Services. In *Proceedings of IRSG'2000*, 2000.
- [Resnick *et al.*, 1994] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: An Open Architecture for Collaborative Filtering of Netnews. In *Proceedings of ACM CSCW'94*, pages 175–186, 1994.
- [Smyth and Cotter, 2000] B. Smyth and P. Cotter. A Personalised TV Listings Service for the Digital TV Age. *Journal of Knowledge-Based Systems*, 13(2-3):53–59, 2000.
- [Smyth and Cotter, 2001] Barry Smyth and Paul Cotter. Personalized Electronic Programme Guides. *Artificial Intelligence Magazine*, 21(2), 2001.

POSTER PAPERS

CONSTRAINTS

A Simulated Annealing Approach to the Travelling Tournament Problem

A. Anagnostopoulos, L. Michel, P. Van Hentenryck, and Y. Vergados

Brown University, Box 1910, Providence, RI 02912

Abstract

Automating the scheduling of sport leagues has received considerable attention in recent years. This paper considers the traveling tournament problem (TTP) proposed in [8; 4] to abstract the salient features of major league baseball (MLB) in the United States. It proposes a simulated annealing algorithm (TTSA) for the TTP that produces significant improvements over previous approaches.

Introduction The scheduling of sport leagues has become an important class of combinatorial optimization applications in recent years since they represent significant sources of revenue and generate extremely challenging optimization problems. This paper considers the travelling tournament problem (TTP) proposed in [4] to abstract the salient features of Major League Baseball (MLB) in the United States. The key to the MLB schedule is a conflict between minimizing travel distances and feasibility constraints on the home/away patterns. Travel distances are a major issue in MLB because of the number of teams and the fact that teams go on “road trips” to visit several opponents before returning home. The feasibility constraints in the MLB restricts the number of successive games that can be played at home or away.

The TTP is an abstraction of the MLB intended to stimulate research in sport scheduling. A solution to the TTP is a double round-robin tournament which satisfies sophisticated feasibility constraints (e.g., no more than three away games in a row) and minimizes the total travel distances of the teams. [4] argues that, without an approach to the TTP, it is unlikely that suitable schedules can be obtained for the MLB. The TTP has raised significant interest in recent years since the challenge instances were proposed. [4] describes both constraint and integer programming approaches to the TTP which generate high-quality solutions. [1] explores a Lagrangian relaxation approach (together with constraint programming techniques) which improves some of the results. Other lower and upper bounds are given in [8], although the details of how they are obtained does not seem to be available.

Problem Description The problem was introduced by Easton, Nemhauser and Trick [8; 4], which contains many interesting discussion on sport scheduling. An input consists of n

teams (n even) and an $n \times n$ symmetric matrix d , such that d_{ij} represents the distance between the homes of teams T_i and T_j . A solution is a schedule in which each team plays with each other twice, once in each team’s home. Such a schedule is called a *double round-robin tournament*. It should be clear that a double round-robin tournament has $2n - 2$ rounds. For a given schedule S , the cost of a team is the total distance that it has to travel starting from its home, playing the scheduled games in S , and returning back home. The cost of a solution is defined as the sum of the cost of every team.

The goal is to find a schedule with minimum cost satisfying the following two constraints: (1) **Atmost Constraints:** No more than three consecutive home or away games are allowed for any team. (2) **Norepeat Constraints:** A game of T_i at T_j ’s home cannot be followed by a game of T_j at T_i ’s home. As a consequence, a double round-robin schedule is feasible if it satisfies the atmost and norepeat constraints and is infeasible otherwise. A schedule is conveniently represented by a table indicating the opponents of the teams. Each line corresponds to a team and each column corresponds to a round. The opponent of team T_i at round r_k is given by the absolute value of element (i, k) . If (i, k) is positive, the game takes place at T_i ’s home, otherwise at T_i ’s opponent home. Consider for instance the schedule S for 6 teams (and thus 10 rounds).

T\R	1	2	3	4	5	6	7	8	9	10
1	6	-2	4	3	-5	-4	-3	5	2	-6
2	5	1	-3	-6	4	3	6	-4	-1	-5
3	-4	5	2	-1	6	-2	1	-6	-5	4
4	3	6	-1	-5	-2	1	5	2	-6	-3
5	-2	-3	6	4	1	-6	-4	-1	3	2
6	-1	-4	-5	2	-3	5	-2	3	4	1

Schedule S specifies that team T_1 has the following schedule. It successively plays against teams T_6 at home, T_2 away, T_4 at home, T_3 at home, T_5 away, T_4 away, T_3 away, T_5 at home, T_2 at home, T_6 away. The travel cost of team T_1 is

$$d_{12} + d_{21} + d_{15} + d_{54} + d_{43} + d_{31} + d_{16} + d_{61}.$$

Observe that long stretches of games at home do not contribute to the travel cost but are limited by the atmost constraints. This kind of tension is precisely why this problem is hard to solve in practice.

The Local Search This paper proposes an advanced simulated annealing algorithm (TTSA) for the TTP. As usual, the algorithm starts from an initial configuration. Its basic step moves from the current configuration c to a configuration in the neighborhood of c . TTSA is based on four main design decisions. (1) Constraints are separated into two groups: hard constraints, which are always satisfied by the configurations, and soft constraints, which may or may not be violated. The hard constraints are the round-robin constraints, while the soft constraints are the no-repeat and at-most constraints. In other words, all configurations in the search represents a double round-robin tournament, which may or may not violate the no-repeat and at-most constraints. Exploring the infeasible region seems to be particularly important for this problem. Obviously, to drive the search toward feasible solutions, TTSA modifies the original objective function to include a penalty term. (2) TTSA is based on a large neighborhood of size $O(n^3)$, where n is the number of teams. Some of these moves can be regarded as a form of ejection chains which is often used in tabu search [6; 7]. (3) TTSA dynamically adjusts the objective function to balance the time spent in the feasible and infeasible regions. This adjustments resembles the strategic oscillation idea [5] successfully in tabu search to solve generalized assignment problems [3], although the details differ since simulated annealing is used as the meta-heuristics. (4) TTSA also uses reheats (e.g., [2]) to escape local minima at low temperatures. The “reheats” increase the temperature again and divide the search in several phases.

The Neighborhood The neighborhood consists of four different moves. Among them, partial swaps of rounds and teams are critical to find high-quality solutions. Consider the move *PartialSwapRounds*(S, T_i, r_k, r_l). This move considers team T_i and swaps its games at rounds r_k and r_l . Then the rest of the schedule for rounds r_k and r_l is updated (in a deterministic way) to produce a double round-robin tournament. Consider the schedule S

T\R	1	2	3	4	5	6	7	8	9	10
1	6	-2	2	3	-5	-4	-3	5	4	-6
2	5	1	-1	-5	4	3	6	-4	-6	-3
3	-4	5	4	-1	6	-2	1	-6	-5	2
4	3	6	-3	-6	-2	1	5	2	-1	-5
5	-2	-3	6	2	1	-6	-4	-1	3	4
6	-1	-4	-5	4	-3	5	-2	3	2	1

and the move *PartialSwapRounds*(S, T_2, r_2, r_9). Obviously swapping the game in rounds r_2 and r_9 would not lead to a round-robin tournament. It is also necessary to swap the games of team 1, 4, and 6 in order to obtain:

T\R	1	2	3	4	5	6	7	8	9	10
1	6	4	2	3	-5	-4	-3	5	-2	-6
2	5	-6	-1	-5	4	3	6	-4	1	-3
3	-4	5	4	-1	6	-2	1	-6	-5	2
4	3	-1	-3	-6	-2	1	5	2	6	-5
5	-2	-3	6	2	1	-6	-4	-1	3	4
6	-1	2	-5	4	-3	5	-2	3	-4	1

n	Old Best	min(D)	max(D)	mean(D)	std(D)
8	39721	39721	39721	39721	0
10	61608	59583	59806	59605.96	53.36
12	118955	112800	114946	113853.00	467.91
14	205894	190368	195456	192931.86	1188.08
16	281660	267194	280925	275015.88	2488.02

Table 1: Solution Quality of TTSA on the TTP

This move can thus be regarded as a form of ejection chain [6; 7]. Finding which games to swap is not difficult: it suffices to find the connected component which contains the games of T_i in rounds r_2 and r_l in the graph where the vertices are the teams and where an edge contains two teams if they play against each other in rounds r_k and r_l . All the teams in this component must have their games swapped. Note that there are $O(n^3)$ such moves.

Experiments TTSA was applied to the National League benchmark described in [4; 8]. Over the course of this research, TTSA was able to match the best solutions for NL4, NL6, and NL8, and to improve the solutions for NL10, NL12, NL14, and NL16 significantly. The best existing solutions were those in [8] and were the best to our knowledge at the time of writing. The last update was on November 1, 2002. TTSA is the first algorithm to go lower than 200,000 for 14 teams and 280,000 for 16 teams. The results are summarized Table 1. Observe that even our worst results on these instances improve the previous best solutions.

Acknowledgements This work is partially supported by by NSF ITR Awards DMI-0121495 and ACI-0121497.

References

- [1] T. Benoist, F. Laburthe, and B. Rottembourg. Lagrange Relaxation and Constraint Programming Collaborative Schemes for Travelling Tournament Problems. In *CP-AI-OR'2001*, 2001.
- [2] D.T. Connelly. General Purpose Simulated Annealing. *Journal of Operations Research*, 43, 1992.
- [3] Juan A. Díaz and Elena Fernández. A tabu search heuristic for the generalized assignment problem. *EJOR*, 132(1):22–38, 2001.
- [4] K. Easton, G. Nemhauser, and M. Trick. The traveling tournament problem description and benchmarks. In *CP'01*, 2001.
- [5] F. Glover and M. Laguna. *Tabu Search*. Kluwer Academic Publishers, 1997.
- [6] M. Laguna, J.P. Kelly, Gonzalez-Velarde, and F. Glover. Tabu search for the multilevel generalized assignment problems. *EJOR*, 42:677–687, 1995.
- [7] E. Pesch and F. Glover. TSP Ejection Chains. *Discrete Applied Mathematics*, 76:165–181, 1997.
- [8] M. Trick. <http://mat.gsia.cmu.edu/TOURN/>, 2002.

Grid-based SensorDCSP*

R. Béjar¹, C. Domshlak², C. Fernández¹, C. Gomes², B. Selman² and M. Valls³

¹Dept. d'Informàtica
Universitat de Lleida
Jaume II, 69. Lleida, Spain

²Dept. of Computer Science
Cornell University
Ithaca, NY 14853, USA

³Dept. de Matemàtica
Universitat de Lleida
Jaume II, 69. Lleida, Spain

Abstract

We introduce Grid-based SensorDCSP, a geometrically structured benchmark problem for the study of distributed CSP algorithms. This domain provides realistic structure of the communication and tracking constraints. We formally define this problem, and perform its worst-case complexity analysis. Likewise, we provide an average case empirical analysis of the AWC algorithm, studying its behavior on tractable and intractable sub-classes of our problem.

1 Introduction

In recent years we have seen an increasing interest in Distributed Constraint Satisfaction Problem (DisCSP) formulations to model combinatorial problems arising in multi-agent environments [Yokoo, 2001]. Recently, an interesting benchmark for DisCSP algorithms, called SensorDCSP, has been introduced in [Fernández *et al.*, 2002]. Inspired by distributed applications arising in networked systems, SensorDCSP involves a network of distributed sensors simultaneously tracking multiple mobile objects. An instance of SensorDCSP is defined by a set of sensors, a set of mobiles which are to be tracked by the sensors, and two sets of visibility and compatibility constraints specifying which sensors can communicate one with another and which sensors can track which mobiles, respectively. The goal is to allocate three sensors to track each mobile, such that all these triplets of sensors are pair-wise disjoint, and the sensors in each such triplet are mutually compatible and can track the mobile they are assigned to. The analysis in [Fernández *et al.*, 2002] addresses the complexity

of SensorDCSP, as well as its empirical evaluation, considering the performance and scalability of alternative DisCSP algorithms on various scenarios in a truly networked environment.

In this work we extend SensorDCSP by providing it with an underlying spatial structure. In SensorDCSP, the constraints of communication between sensors and visibility of mobiles are independent of their physical location. In contrast to SensorDCSP, the spatial limitations for both communication between sensors and visibility of the mobiles are captured by a geometrical structure in our new benchmark, providing a more realistic framework. We refer to our geometrically enriched benchmark as Grid-based SensorDCSP or just *GSensorDCSP*. We provide both an analytical complexity analysis of *GSensorDCSP* and an empirical study of a characteristic DisCSP algorithm on various instances of this problem.

2 GSensorDCSP: Definition and Complexity

GSensorDCSP is a specific variant of the general SensorDCSP, enriched by a spatial structure: we have multiple sensors $S = \{s_1, \dots, s_m\}$, multiple objects (mobiles) $T = \{t_1, \dots, t_n\}$ which are to be tracked by the sensors subject to visibility and compatibility constraints, and the goal is to decide whether there exists an allocation of three sensors to each object, while keeping these triplets of sensors pair-wise disjoint. However, the visibility and compatibility constraints in *GSensorDCSP* relate to the physical limitations of the sensors and the terrain on which the sensors are located.

In *GSensorDCSP*, the sensors are located on the nodes of a

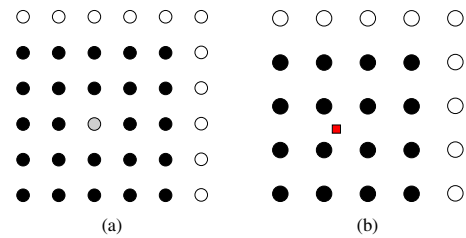


Figure 1: Grid of 6×6 sensors; black sensors stand for the (a) 2-compatibility window of the gray sensor, and (b) 2-visibility window of the square mobile.

*Research supported by AFOSR, grant F49620-01-1-0076 (Intelligent Information Systems Institute) and F49620-01-1-0361 (MURI grant on Cooperative Control of Distributed Autonomous Vehicles in Adversarial Environments), CICYT, TIC2001-1577-C03-03 and DARPA, F30602-00-2-0530 (Controlling Computational Cost: Structure, Phase Transitions and Randomization) and F30602-00-2-0558 (Configuring Wireless Transmission and Decentralized Data Processing for Generic Sensor Networks). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of AFOSR, DARPA, or the U.S. Government.

uniform grid, and the mobiles are located within the surface enclosed by this grid; this way the grid specifies the generally trackable region. The physical limitations of the sensors are modeled by the notions of k_c -compatibility and k_v -visibility. The k_c -compatibility window for a sensor s_i , denoted as $\mathbb{C}^{k_c}(s_i)$, corresponds to the set of all sensors that are at most k_c general (rectilinear and/or diagonal) hops from s_i . For example, the black sensors in Figure 1(a) correspond to 2-compatibility window for the gray sensor. Similarly, the k_v -visibility window for a mobile t_j , denoted as $\mathbb{V}^{k_v}(t_j)$, corresponds to the set of all sensors that are at most k general hops around t_j . For example, the black sensors in Figure 1(b) correspond to 2-visibility window for the square mobile. Finally, if each sensor s_i of a given GSensorDCSP problem instance can communicate only with some sensors within $\mathbb{C}^{k_c}(s_i)$, then the set of compatibility constraints of this instance is called k_c -restricted. The notion of k_v -restricted visibility is defined similarly.

Studying the worst-case complexity of GSensorDCSP we formally show that:

1. Any GSensorDCSP instance with 1-restricted visibility is solvable in low polynomial time, and
2. GSensorDCSP with k_v -restricted visibility is NP-complete for all $k_v \geq 2$.

The former is shown by a reduction to the problem of feasible integer flow in bipartite graphs, while the latter is shown by a non-trivial reduction from 3-SAT and is valid even for the problems with 1-restricted compatibility.

3 Connecting Locality and Constrainedness

While the physical limitations of the sensors in GSensorDCSP are modeled via the locality windows, the terrain limitations are modeled via incomplete compatibility and visibility *within* the windows. Problem instances of any GSensorDCSP sub-class (k_c, k_v) can be ordered according to the *local constrainedness*, i.e., the average number of sensors that a sensor can communicate with and the average number of sensors that can track a mobile. In our experiments, for each pair of locality parameters (k_c, k_v) , we define a random distribution of GSensorDCSP instances via two parameters $P_v, P_c \in (0, 1]$ that control the density of visibility and compatibility: The probability that two sensors s_i and s_j will be compatible is given by P_c if $(s_i, s_j) \in \mathbb{C}^{k_c}(s_i)$, otherwise, it is equal to 0. Clearly, higher value for P_c and P_v correspond to better conditions for communication and tracking, respectively.

For our first experiments with the AWC [Yokoo, 1994] algorithm, we consider different sets of instances with 25 sensors (grid 5×5), and 5 mobiles, with every set generated with different values for the parameters P_c and P_v . The parameters P_c and P_v are ranging from 0.1 to 1 with an increment of 0.1, giving a total number of 100 data sets, where every set contains 50 instances. Given our formal complexity results, we consider several hard subclasses of GSensorDCSP corresponding to $k_v = 2$. For instance, Figure 3(a) shows the ratio of the satisfiable instances (P_{sat}) as a function of P_c and P_v for $k_v = 2$ with $k_c = 1$ and $k_c = 2$. As in the case of general SensorDCSP [Fernández *et al.*, 2002], when both

probabilities are low, the instances generated are mostly unsatisfiable, while for high probabilities most of the instances are satisfiable. Both for $k_c = 1$ and $k_c = 2$, the transition between the satisfiable and unsatisfiable regions occurs within a narrow range of the density parameters. Observe that, for $k_c = 1$ this range corresponds to significantly higher values of P_c and P_v , comparatively to these for $k_c = 2$ and $k_c = 2$. However, the form of the transition for various values of k_c is very similar.

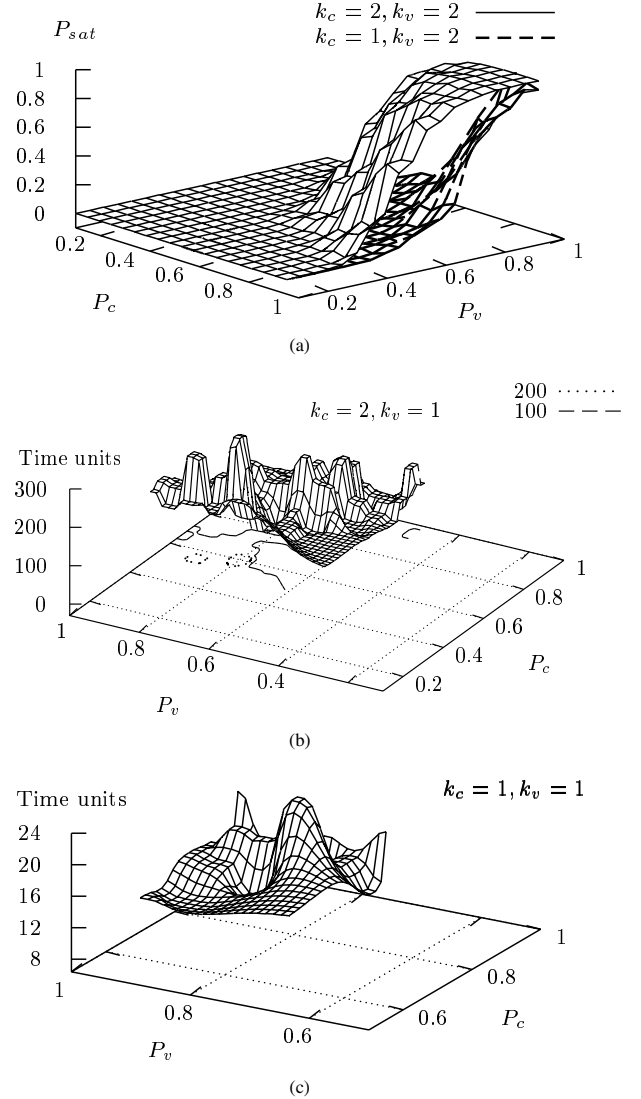


Figure 2: Some experiments with AWC on instances of GSensorDCSP: (a) ratio of satisfiable instances for $k_v = 2$, and mean solution time for (b) $k_v = 2$ and (c) $k_v = 1$, as a function of the density parameters P_c and P_v .

Consistently with the general SensorDCSP, we observe that the phase transition of GSensorDCSP coincides with the region where the hardest instances occur. For instance, Figure 3(b) shows the mean solution time with respect to the density parameters P_v and P_c for the problem instances with 25 sensors, 5 mobiles, $k_v = 3$, and $k_c = 1$. Somewhat

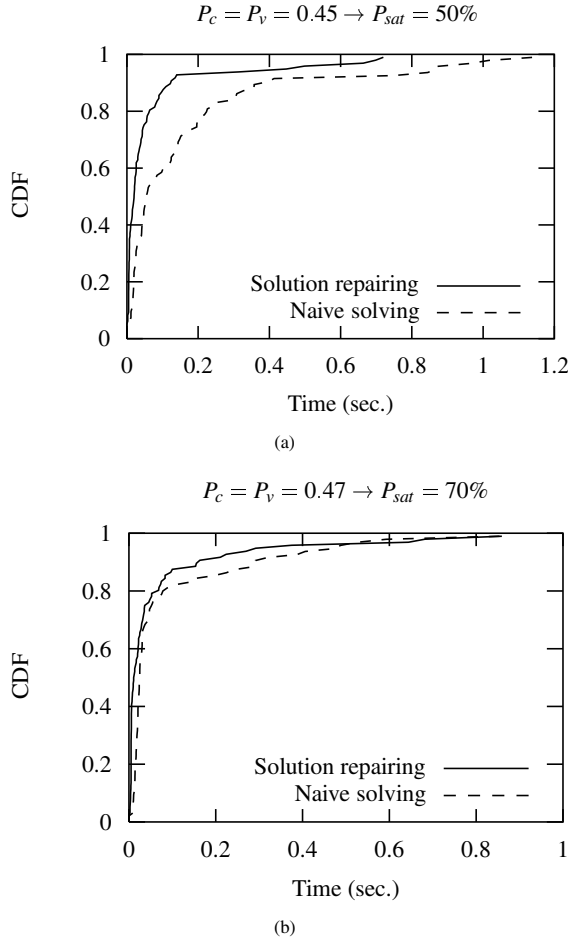


Figure 3: Solving dynamic GSensorDCSP problems with the naive and the solution repairing approaches: (a) performance for a problem generated from an instance in the 70 % P_{sat} (b) performance for a problem generated from a hard instance in the 50% P_{sat} point.

less expected result is depicted in Figure 3(c) for the case of $k_v = 1$ (and $k_c = 1$), which we showed to be polynomial by a reduction to the problem of feasible integral flow in bipartite graphs. Despite the fact that AWC is a general purpose algorithm, Figure 3(c) shows that these tractable instances are relatively easy for AWC. able to “detect and exploit” the tractable without being aware of its existence.

Extending the evaluation, we consider different sets of instances for several orders of the problem (grids of 25–100 sensors), and several levels of decomposition ($k_c, k_v \in \{2, 3, 4, 5\}$). The analysis of these sets of instances clearly shows both exponential dependence of the mean solution time on the size of the grid, and, somewhat more interestingly, growth of the slope of this dependence with k_v .

4 Dynamic GSensorDCSP

As an additional extension of the GSensorDCSP domain, we consider the dynamics of the mobiles, in order to determine under which conditions (number of sensors and mobiles, velocities, trajectories, etc.) generic DisCSP algorithms are able

to track the mobiles in real time. More formally, a dynamic GSensorDCSP problem can be seen as an ordered sequence of static GSensorDCSP problems $\Pi_1, \dots, \Pi_N$. The static problems of the sequence only differ in the positions of the mobiles and hence the visibility between sensors and mobiles. Observe that, if no assumptions are made about the relationship between the positions of the mobiles in Π_i and Π_{i+1} , it is natural to solve these problems independently (naive approach). However, mobile dynamics are typically far from being chaotic. A first approximation is to consider linear trajectories. In this case, changes between the subsequent problems are governed by a clear model. An approach that a priori seems to be promising for dealing with the dynamic problem is to initialize the search for Π_i by the solution already achieved for Π_{i-1} (solution repairing approach).

In order to compare the naive and the solution repairing approach, we have performed experiments where we solve dynamic GSensorDCSP problems with both approaches. Figure 3 depicts the cumulative probability distributions of solving Π_i for two different problems. One of them is obtained from the 50 % P_{sat} point of GSensorDCSP and the other from the 70 % P_{sat} point.

Our results show that we can exploit the previous solution and that, assuming certain reasonable constraints on the mobile movements, we can benefit from solution repairing over the naive approach. More interestingly, it also follows that the relative attractiveness of solution repairing is higher in the region of hardest instances.

5 Conclusions and Future Work

We believe that GSensorDCSP provides a realistic framework for the analysis of the DisCSP algorithms, and we hope it will contribute to the further research in this area. For the dynamic version, a next step would be to consider the effect of several kinds of trajectories in the performance of the solution repairing approach. Finally, a further step towards an even more realistic DisCSP benchmark would be to consider its optimization version: maximizing the number of tracked mobiles. In this extended distributed benchmark, it would be interesting to study the existence of the easy-hard complexity patterns that have been observed in classical optimization problems [Slaney and Walsh, 2002].

References

- [Fernández *et al.*, 2002] C. Fernández, R. Béjar, B. Krishnamachari, and C. P. Gomes. Communication and computation in distributed CSP algorithms. In *Proc. of CP-2002*, pages 664–679, 2002.
- [Slaney and Walsh, 2002] J. Slaney and T. Walsh. Phase transition behavior: from decision to optimization. In *Proceedings of the SAT-2002 Conference*, 2002.
- [Yokoo, 1994] M. Yokoo. Weak-commitment search for solving constraint satisfaction problems. In *Proc. of AAAI-94*, pages 313–318, 1994.
- [Yokoo, 2001] M. Yokoo. *Distributed Constraint Satisfaction: Foundations of Cooperation in Multi-Agent Systems*. Springer-Verlag, 2001.

Dynamic Vehicle Routing with Stochastic Requests

Russell Bent and Pascal Van Hentenryck

Brown University
{rbent,pvh}@cs.brown.edu

Abstract

This paper considers vehicle routing problems (VRP) where customer locations and service times are random variables that are realized dynamically during plan execution. It proposes a multiple scenario approach (MSA) that continuously generates plans consistent with past decisions and anticipating future requests. The approach, which combines AI and OR techniques in novel ways, is compared with the best available heuristics that model long-distance courier mail services [Larsen *et al.*, 2002]. Experimental results shows that MSA may significantly decrease travel times and is robust wrt reasonably noisy distributions.

1 Introduction

The vehicle routing problem (VRP) is a difficult combinatorial optimization problem with many important applications in distribution and transportation systems. It has received considerable attention from the operations research (OR) and AI communities for many years and sophisticated local search methods are quite effective at finding good quality solutions in reasonable amounts of time. In more recent years, technology has advanced so that it is now possible and practical to address dynamic and/or stochastic versions of the problem. These new versions are motivated by the inherent uncertainties that arise in many everyday VRPs and advances in onboard computers and communications systems that allow modification of routing plans even after vehicles are deployed.

Most of the existing work has focused solely on stochastic or dynamic versions of the problem exclusively. In stochastic optimization, the expected cost of a solution is optimized with respect to a recourse function which restores feasibility during plan execution. In dynamic optimization, various data items, such as customer requests, are unknown and are only revealed after some decisions are taken. A more detailed overview of the methods used in both approaches can be found in [Bent and Van Hentenryck, 2002]. How to combine the two approaches is a research topic that is often mentioned (e.g., [Gendreau *et al.*, 1999], among others).

This paper considers the multiple scenario approach (MSA) recently proposed in [Bent and Van Hentenryck,

2002]. MSA combines a variety of AI and OR techniques to address dynamic vehicle routing in the presence of stochastic information.

2 Problem Formulation

In the dynamic VRP, a number of customer requests are available initially, while others become available during the plan execution. In the applications considered in this paper, a request consists of the location of a customer and a service time, both of which are random variables. We assume that the distribution of the requests, or some approximation thereof, is available, which is typically the case in practical applications. For each incoming request, a dynamic algorithm must decide whether to accept or reject it. Once a request is accepted, it must be serviced. Problems are generally characterized by their degree of dynamism (DOD), i.e., the ratio of *unknown customers/total customers* [Larsen *et al.*, 2002]. A more in depth formalization of the VRP model can be found in [Bent and Van Hentenryck, 2002].

3 The Approaches

The Multiple Plan Approach The Multiple Plan Approach (MPA) is a fundamental generalization of many modern approaches. MPA generalizes the approach of [Gendreau *et al.*, 1999] by making it independent of the search procedure. In short, MPA continuously generates plans that are compatible with the current state of information and removes those that are not. In addition, since decisions must be made with respect to a specific plan to guarantee service, a *distinguished* plan is maintained via a ranking function.

More precisely, MPA handles four types of events (1) customer requests, (2) vehicle departures, (3) plan generations, (4) timeouts. A more complete description of the events and MPA can be found in [Bent and Van Hentenryck, 2002].

The Multiple Scenario Approach The Multiple Scenario Approach (MSA) generalizes MPA by considering both existing and potential future requests during plan generation. Future requests are obtained by sampling their distributions. Once a routing plan σ is discovered, MSA stores the routing plan σ^- obtained by removing future requests from σ . As a result, plan σ^- leaves room to accommodate future requests, should they actually materialize. This ability to anticipate the future is the strength of MSA.

Ranking Functions Both MPA and MSA are parametrized by a ranking function f_t , which selects the *distinguished* plan at each time t . We will evaluate two ranking functions for nominating σ^* in this paper. The obvious first choice for f_t would be to select the plan with the smallest travel cost (MPA^d). In [Bent and Van Hentenryck, 2002], it was shown that it is possible to do substantially better in practice on highly-constrained problems by using a consensus function which selects the plan in that most resembles all the stored plans.

Optimization Both MPA and MSA use large neighborhood search (LNS) for optimization, which has been shown to be very effective for vehicle routing [Shaw, 1998; Bent and Van Hentenryck, 2001]. LNS combines the advantages of branch and bound, constraint propagation, and local search.

4 Experimental Results

The Models The starting point of this research was the experimental model in [Larsen *et al.*, 2002], where customers are uniformly distributed in a 10km×10km region and must be served by a single vehicle with uniform speed of 40 km/h. Customers inter-arrival times are governed by an exponential distribution. Problem sets are derived with expected DOD of $\{0, 5, \dots, 100\}$. More details about the model are found in [Larsen *et al.*, 2002]. We generated 15 instances for each DOD configuration, which gives 315 problems for each model described.

Several models are used to evaluate the various approaches. The first two models, M1 and M2, use a single vehicle. Model M1 is the basic model with 40 customers. Model M2 is similar to M1, except that the region is now 40km×40km. The objective function minimizes the travel distance. The next model is a multiple-vehicle model. Model M3 uses 4 vehicles, 160 customers, and a 20km×20km space. Each vehicle can serve at most 50 customers and the vehicle must return to the depot by the time horizon. For the multiple-vehicle model, the objective function consists in minimizing the number of missed customers and minimizing the travel distance. It is possible that some customers be left unserved, since M3 has capacity constraints, as well as a hard deadline.

Single Vehicle Results Reference [Larsen *et al.*, 2002] tested various heuristics on Model M1. Their best heuristic is nearest neighbor (NN), where a pool of unserved customers is maintained and the vehicle is sent to the nearest customer in the pool once it served its current request. Interestingly, in Model M1, the vehicle is able to service customers faster than they arrive. As a consequence, all “reasonable” heuristics converge towards a first come, first serve (FCFS) strategy as the DOD converges to 100%. This same behavior is also exhibited by our approaches. In Model M2, the vehicle is not able to service customers as quickly and the heuristics do not converge to FCFS. Figure 1 shows that MPA^d produces a 4.6% improvement on average and the benefits can be as high as 11.1%. Stochastic information (MSA^d) improves these results slightly. Moreover, MSA^d is never worse than NN. These results essentially indicate that the LNS-based MPA approach provides clear benefits over traditional heuristics. Stochastic information helps, but only marginally.

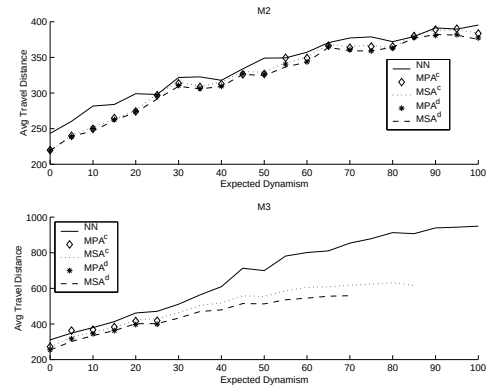


Figure 1: Travel Distance Results

Multiple Vehicle Results First, the NN heuristic was generalized to provide guarantees on servicing customers. Whenever a request arrives, the NN algorithm is simulated to determine if it can accommodate the new request. Surprisingly, the MPA approaches perform very poorly, especially for higher DODs. The “optimized” plans of MPA always use as few vehicles as possible since this generally improves travel distance and feasibility is not a issue until late in the search. This is in sharp contrast with problems with time-windows where feasibility is a major issue and MPA does not experience such problems [Bent and Van Hentenryck, 2002]. The MSA approaches perform roughly the same as NN as far as missed customers are concerned, but the true benefit of MSA is with respect to travel costs as shown in Figure 1. (The figure only shows the results for similar number of unserved requests). Here, MSA decreases travel distances significantly, especially for high DODs. *These models clearly show the benefits of MSA and the value of stochastic information over traditional heuristics and MPA.* These results are even more dramatic when non-uniform distributions are used. Significant gains are observed for DODs as low as 25%. In addition to these results, the algorithm is robust under reasonable amounts of noise, for example when MSA is given incorrect customer arrival rates.

References

- [Bent and Van Hentenryck, 2001] R. Bent and P. Van Hentenryck. A Two-Stage Hybrid Local Search for the Vehicle Routing Problem with Time Windows. To appear in *Transportation Science*
- [Bent and Van Hentenryck, 2002] R. Bent and P. Van Hentenryck. Scenario-Based Planning for Partially Dynamic Vehicle Routing with Stochastic Customers. To appear in *Operations Research* (pending minor revisions).
- [Gendreau *et al.*, 1999] M. Gendreau, F. Guertin, J. Y. Potvin, and E. Taillard. Parallel Tabu Search for Real-Time Vehicle Routing and Dispatching. *Transportation Science*, 33 (4):381–390, 1999.
- [Larsen *et al.*, 2002] A. Larsen, O. Madsen, and M. Solomon. Partially Dynamic Vehicle Routing-Models and Algorithms. *Journal of Operational Research Society*, 53:637–646, 2002.
- [Shaw, 1998] P. Shaw. Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems. In *Principles and Practice of Constraint Programming*, 417–431, 1998.

Solving Finite Domain Constraint Hierarchies by Local Consistency and Tree Search*

Stefano Bistarelli[†]

Dipartimento di Scienze
Università “D’Annunzio” di Pescara, Italy
bista@sci.unich.it

Istituto di Informatica e Telematica, C.N.R.
Pisa, Italy
stefano.bistarelli@iit.cnr.it

Philippe Codognet

Univ. of Paris 6, LIP6, Paris Cedex 05, France
Philippe.Codognet@lip6.fr

H.K.C. Hui and J.H.M. Lee

Dept of Comp. Sci. & Eng.
The Chinese Univ. of Hong Kong, Hong Kong
{kchui,jlee}@cse.cuhk.edu.hk

Abstract

We provide a reformulation of the constraint hierarchies (CHs) framework based on the notion of *error indicators*. Adapting the generalized view of local consistency in semiring-based constraint satisfaction problems (SCSPs), we define *constraint hierarchy k-consistency* (CH-*k*-C) and give a CH-2-C enforcement algorithm. We demonstrate how the CH-2-C algorithm can be seamlessly integrated into the ordinary branch-and-bound algorithm to make it a finite domain CH solver. Experimentation confirms the efficiency and robustness of our proposed solver prototype. Unlike other finite domain CH solvers, our proposed method works for both local and global comparators. In addition, our solver can support arbitrary error functions.

1 Introduction

The Constraint Hierarchy (CH) framework [Borning *et al.*, 1992] is a general framework for the specification and solutions of over-constrained problems. Originating from research in interactive user-interface applications, the CH framework attracts much effort in the design of efficient solvers in the real number domain [Badros *et al.*, 2001; Hosobe *et al.*, 1996]. To extend the benefit of the CH framework to also discrete domain applications, such as timetabling and resource allocation, the paper takes a step towards a general and efficient finite domain CH solver, based on consistency techniques and tree search.

*A more complete and detailed version of this paper can be obtained from the authors.

[†]Part of this research has been carried out while the author was visiting the Department of Computer Science and Engineering, The Chinese University of Hong Kong, China

2 A Reformulation of Constraint Hierarchies

We formulate the CH framework [Borning *et al.*, 1992] (in particular in the definition of comparators and solution set) using error indicators [Bistarelli *et al.*, 1999].

We denote an error value by ξ , possibly with subscripts. Let $I = \{\xi_1, \dots, \xi_N\}$ be a poset (partially ordered set), each element ξ_j of which is an *error indicator*. Given a constraint hierarchy $H = \{H_0, \dots, H_n\}$ where n is the number of non-required levels, and for all $i \in \{0, \dots, n\}$, $H_i = \{c_1^i, \dots, c_{k_i}^i\}$ with k_i being the number of constraints in level i . An error indicator ξ_θ of a valuation θ for a set of variables V is a tuple of error values such that $\xi_\theta = \langle \langle \xi_{\theta_1}^0, \dots, \xi_{\theta_{k_0}}^0 \rangle, \dots, \langle \xi_{\theta_1}^n, \dots, \xi_{\theta_{k_n}}^n \rangle \rangle$ and $\forall a \in \{0, \dots, n\}, \forall b \in \{1, \dots, k_a\}, \xi_{\theta_b}^a = e(c_b^a \theta)$ if $\text{vars}(c_b^a) \subset V$ and $\xi_{\theta_b}^a = 0$ if $\text{vars}(c_b^a) \not\subset V$. Error indicators provide a measure of the “badness” of valuations with respect to H .

The comparator predicate *better* in the original CH formulation is redefined using a *partial order*, denoted by $\prec$. We define $\prec$ to be irreflexive and transitive over I . Hence, it preserves the meaning of *better*. Intuitively, $\xi^i \prec \xi^j$ means ξ^j is “better” than ξ^i in I . In general, $\prec$ will not provide a total ordering. For convenience, we define $\preceq$ such that $\forall \xi^i, \xi^j \in I, \xi^i \preceq \xi^j \rightarrow (\xi^i \prec \xi^j) \vee (\xi^i = \xi^j)$.

We can redefine *l-b* in the original formulation as a partial order $\prec_{l-b}$ as follows. Given any two valuations θ and σ , and the corresponding error indicators ξ_θ and ξ_σ , $\prec_{l-b}$ is defined as:

$$\begin{aligned} \xi_\theta \prec_{l-b} \xi_\sigma &\equiv \exists l > 0 \text{ such that } \forall i \in \{0, \dots, l-1\}, \\ &\forall j \in \{1, \dots, k_i\}, \xi_{\theta_j}^i = \xi_{\sigma_j}^i \\ &\wedge \exists a \in \{1, \dots, k_l\}, \xi_{\theta_a}^l < \xi_{\sigma_a}^l \\ &\wedge \forall b \in \{1, \dots, k_l\}, \xi_{\theta_b}^l \leq \xi_{\sigma_b}^l. \end{aligned}$$

The intuitive meaning of $\xi_\theta \prec_{l-b} \xi_\sigma$ is that valuation σ is *locally-better* than valuation θ .

Similarly, we can define $g\text{-}b \prec_{g\text{-}b}$, and its instances $w\text{-}s\text{-}b \prec_{w\text{-}s\text{-}b}$, $w\text{-}c\text{-}b \prec_{w\text{-}c\text{-}b}$, and $l\text{-}s\text{-}b \prec_{l\text{-}s\text{-}b}$ respectively. Given any two valuations θ and σ , and the corresponding error indicators $\vec{\xi}_\theta$ and $\vec{\xi}_\sigma$:

$$\vec{\xi}_\theta \prec_{g\text{-}b} \vec{\xi}_\sigma \equiv \exists l > 0 \text{ such that } \forall i \in \{0, \dots, l-1\}, \\ g(\langle \xi_{\theta 1}^i, \dots, \xi_{\theta k_i}^i \rangle) = g(\langle \xi_{\sigma 1}^i, \dots, \xi_{\sigma k_i}^i \rangle) \\ \wedge g(\langle \xi_{\sigma 1}^l, \dots, \xi_{\sigma k_l}^l \rangle) < g(\langle \xi_{\theta 1}^l, \dots, \xi_{\theta k_l}^l \rangle),$$

where g is a suitable *combining function* for error values.

Notice that by definition, all local/global comparators ignore constraints in hierarchy levels greater than or equal to l .

We are now ready to define the solution set S_H of a CH H with variables V by:

$$S_0 = \{\theta \mid \text{vars}(\theta) = V, \xi_{\theta i}^0 = 0 \text{ for all } i \in \{1, \dots, k_0\}\} \\ \text{and } S_H = \{\theta \in S_0 \mid \forall \sigma \in S_0, \vec{\xi}_\theta \not\prec \vec{\xi}_\sigma\}.$$

The following lemma gives the monotonicity of the introduced comparators, which are collectively denoted by $\prec_{better}$ and $\preceq_{better}$ in the rest of the paper.

Lemma 1 *Given any two error indicators $\vec{\xi}^i$ and $\vec{\xi}^j$. If for all a, b we have $\xi_b^{i+a} \leq \xi_b^{j+a}$, then $\vec{\xi}^i \preceq_{better} \vec{\xi}^j$.*

Notice that the above lemma let us compare valuation for both local and global comparators (because the $\preceq_{better}$ order implies all the orders induced from any specific comparator) and for arbitrary error functions.

We also introduce the notion of a *hierarchy problem* which is a CH augmented with an *Error Indicator Store*.

Definition 1 *A hierarchy problem $P = \langle H, I_H \rangle$ is a constraint problem, where H is a CH with variables V and I_H is a set containing error indicator stores $\vec{\xi}_{x=d}$ for all variables $x \in V$ and for all $d \in D(x)$. Each $\vec{\xi}_{x=d}^1$ is used for keeping an estimate of the errors of valuations involving $\{x \mapsto d\}$.*

Definition 2 *A valuation θ is a solution of $P = \langle H, I_H \rangle$ if (1) θ is a solution of H and (2) $\vec{\xi}_\theta \preceq_{better} \vec{\xi}_{x=d}$ for all $\vec{\xi}_{x=d} \in I_H$.*

In other words, solutions of $P = \langle H, I_H \rangle$ are solutions of H which have a “worse” error than the estimates provided in I_H . By the definition, the solutions of H always contain those of $\langle H, I_H \rangle$. Equality holds when the error estimates provided in I_H fails to “filter” out any solutions of H .

Theorem 1 *Consider a CH H and the associated hierarchy problem $P = \langle H, I_H \rangle$.*

- $S_P \subseteq S_H$, and
- $S_P = S_H$ if $\vec{\xi}_\theta \preceq_{better} \vec{\xi}_{x=d}$ for all $(x \mapsto d) \in \theta$ and $\theta \in S_H$.

In particular, a hierarchy problem $\langle H, I_H \rangle$ must share the same solution as H if all $\vec{\xi}_{x=d} \in I_H$ contain only the error value 0 (i.e. no error information). This fact is useful in ensuring the correctness of our local consistency algorithm and the completeness of our branch-and-bound solver later.

¹Note the similarity of $\vec{\xi}_{x=d}$ in notation to $\vec{\xi}_\theta$ which denotes the error indicator of a valuation θ . Here, $\vec{\xi}_{x=d}$ is a *store* containing an estimate of the error associated with valuations containing $x \mapsto d$.

3 Summary of the Results

The notion of *constraint hierarchy k-consistency* (CH- k -C), is defined using error indicators which are structures isomorphic to the structure of a given CH used for storing the error information of the CH problem (similar notion was defined by Bistarelli *et al.* [1999]). In particular, we give an algorithm for enforcing CH-2-C of a CH problem. While classical consistency algorithms [Mackworth, 1977] aim to reduce the size of constraint problems, our CH-2-C algorithm works by explicating error information that is originally implicit in CH problems. The space complexity of the CH-2-C algorithm is simply of $O(n_v n_d n_c)$ in the worst case where n_c is the number of labeled constraints, n_v is the number of variables, and n_d is the size of the largest variable domain.

Incorporating a CH-2-C enforcement algorithm in a branch-and-bound algorithm, we obtain a general finite domain CH solver, which works for arbitrary comparators. Search space is pruned by utilizing the error information generated by the CH-2-C algorithm. Unlike other finite domain CH solvers, our proposed solver is applicable to arbitrary comparators.

We compare the performance of our proposed solver with generate-and-test, basic branch-and-bound, and the reified constraint approach by Lua [2000]. Experiments confirm the efficiency and robustness of our research prototype, which brings us one step towards practical finite domain CH solving.

References

- [Badros *et al.*, 2001] G.J. Badros, A. Borning, and P.J. Stuckey. The cassowary linear arithmetic constraint solving algorithm. *ACM Transactions on Computer-Human Interaction*, 8(4):267–306, 2001.
- [Bistarelli *et al.*, 1999] S. Bistarelli, Y. Georget, and J.H.M. Lee. Capturing (fuzzy) constraint hierarchies in semiring-based constraint satisfaction. Unpublished Manuscript, 1999.
- [Borning *et al.*, 1992] A. Borning, B. Freeman-Benson, and M. Wilson. Constraint hierarchies. *Lisp and Symbolic Computation*, 5(3):223–270, 1992.
- [Henz *et al.*, 2000] Martin Henz, Yun Fong Lim, Seet Chong Lua, Xiao Ping Shi, J. Paul Walser, and Roland H. C. Yap. Solving hierarchical constraints over finite domains. In *Sixth International Symposium on Artificial Intelligence and Mathematics*, Fort Lauderdale, Florida, 2000.
- [Hosobe *et al.*, 1996] H. Hosobe, S. Matsuoka, and A. Yonezawa. Generalized local propagation: A framework for solving constraint hierarchies. In *Proceedings of the Second International Conference on Principles and Practice of Constraint Programming*, pages 237–251, 1996.
- [Mackworth, 1977] A.K. Mackworth. Consistency in networks of relations. *AI Journal*, 8(1):99–118, 1977.

Splitting the atom: A new approach to Neighbourhood Interchangeability in Constraint Satisfaction Problems

James Bowen and Chavalit Likitvivatanavong

Cork Constraint Computation Centre, University College Cork, Ireland
{j.bowen, chavalit}@4c.ucc.ie

Abstract

We investigate interchangeability of values in CSPs, based on an approach where a single value in the domain of a variable can be treated as a combination of “sub-values”. An algorithm for removing overlapping sub-values is presented. The resulting CSPs take less time to find all solutions and yield a more compactly-representable, but equivalent, solution set. Experimental results show that, especially in loose problems with large numbers of solutions, dramatic savings in search cost are achieved.

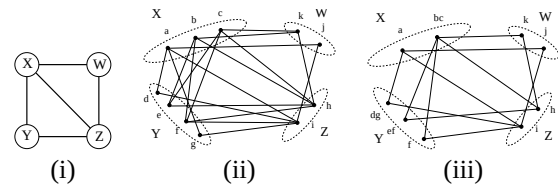
1 Introduction

While most CSP algorithm research is based on the assumption that only one solution to a CSP need be found, the central idea in this paper is motivated by a requirement of certain interactive constraint-based applications: sometimes, before making his next interactive decision, the user needs to know *all* solutions to a sub-CSP. Thus, we need an approach which will reduce the search cost of finding all solutions to a (sub-)CSP and which will enable a more compact representation of the complete solution set. Neighbourhood Interchangeability (NI) [Freuder, 1991] of domain values suggests itself, since merging NI values in a variable domain reduces both thrashing during search and the complexity of enumerating the solution set. Without loss of generality, we restrict our attention here to *binary* CSPs, where the constraints involve two variables.

Our idea is based on a new approach to NI. Normally, each domain value is treated as atomic. Our idea is to “split the atom” – a domain value can be split into several “sub-values” so that interchangeable fragments from other values can then be merged to avoid duplicate search effort during the solving process.

When the values in a domain have been split into fragments, interchangeable fragments of different values can be merged. Figure (i) depicts a constraint network while (ii) shows the microstructure. Figure (iii) shows the result of splitting and merging value fragments, resulting in smaller domains for X and Y .

The approach subsumes NI, as can be seen in X where b and c , which are NI, are merged; similarly, d and g in Y



are merged because they, also, are NI. That the approach surpasses NI can be seen by considering values e and f in Y . Values e and f are not NI because, while both e and f support (b, h) and (c, h) , f also supports (b, i) and (c, i) . However, we can split f into two fragments, one of which supports (b, h) and (c, h) , while the other supports (b, i) and (c, i) . The first fragment can be merged with e and the second can remain isolated.

2 Background and Definitions

We will use a cross-product representation (CPR) [Hubbe and Freuder, 1992] to denote sets of solutions to the star-graph CSPs that we must consider when transforming variable domains. For example, consider the star-graph centred at Y in Figure (ii). The set of solutions to this star-graph in which the centre variable, Y , has the value f can be represented by the following CPR: $\{b, c\} \times \{f\} \times \{h, i\}$.

We define an operator, called “commerge” (commonality extraction and merge), on two CPR sets as follows. The operator extracts, from the two CPR sets, those solutions which have common values for the leaf variables and merges the values for the centre variable. To compute the commerge of two CPR sets, the sets of values for the leaf variables are combined using normal set intersection but the sets of values for the centre variable are combined using normal set union. We denote the commerge operator as $\odot$. Thus, for example, $\{a, b\} \times \{x\} \times \{c\} \odot \{a\} \times \{y\} \times \{c, d\}$ is $\{a\} \times \{x, y\} \times \{c\}$.

We define the “communion” of two CPRs as follows. Two CPRs can be communioned only if the CPRs differ from each other in at most one of the cross-multiplied sets, which cannot be the set for the centre variable. In the communion, all the common cross-multiplied sets are unchanged, but the differing leaf sets are subjected to ordinary set union. We denote the communion operator as $\oplus$. For example, the two CPRs $\{a\} \times \{b\} \times \{c\}$ and $\{a\} \times \{b\} \times \{d\}$ can be communioned because they differ only in the right-most subsets which do not

correspond to the centre variable; the communion of these two CPRs is $\{a\} \times \{b\} \times \{c, d\}$.

We also define a new form of subtraction, the “comminus” of two CPRs, as follows. One CPR can be comminused from another only if they have a common set for the centre variable. Given that provision, the comminus of two CPRs, denoted by $\ominus$, is the same as subtraction. Note that the operator can result in a set of CPRs. Consider, for example, the result of subtracting $\{a\} \times \{x\} \times \{c\}$ (which contains only one solution to a star-graph) from $\{a, b\} \times \{x\} \times \{c, d\}$ (which contains four solutions). The difference between these two CPRs contains three solutions: $a-x-d$, $b-x-c$, $b-x-d$. In CPR form, this set of three solutions needs two CPRs. That is, $\{a, b\} \times \{x\} \times \{c, d\} \ominus \{a\} \times \{x\} \times \{c\}$ is $\{\{a\} \times \{x\} \times \{d\}, \{b\} \times \{x\} \times \{c, d\}\}$ or, alternatively, $\{\{a, b\} \times \{x\} \times \{d\}, \{b\} \times \{x\} \times \{c\}\}$.

3 Transforming CSPs

The procedure for transforming the domain of a single variable is stated in pseudocode below.

Procedure transform(V)

```

queue  $\leftarrow$  all CPR centered in  $V$ ;
result  $\leftarrow \emptyset$ ;
while queue  $\neq \emptyset$  do
  Select and delete any  $S$  from queue;
  match  $\leftarrow$  false;
  while (not match) and (next  $T$  in result exists) do
    if  $S \odot T \neq \emptyset$  then
      match  $\leftarrow$  true;
       $C \leftarrow S \odot T$ ;
       $RS \leftarrow S \ominus C$ ;
       $RT \leftarrow T \ominus C$ ;
      Replace  $T$  in result with  $C$  and  $RT$ ;
      Insert  $RS$  to queue;
  if not match then
    Insert  $S$  to result

```

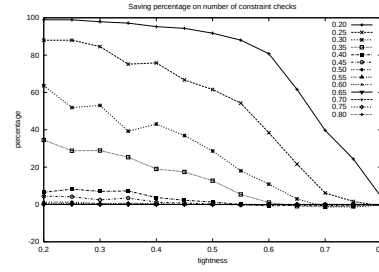
Where possible, communion the CPRs in result;
Update the domain of V and its connecting constraints;

The size of **result** could be very large if there are many fragments of CPRs, making it unlikely that the resulting CPRs can be compressed so that the resulting domain is smaller than the original. Therefore we impose an upper bound on the size of **result**; if **result** exceeds the upper bound, transformation of the variable’s domain is aborted. This cutoff heuristic reduces the transformation time, making the algorithm practical even though the worst-case time complexity is exponential.

The overall transformation process involves transforming each variable domain, one by one, in some order. The order in which variables are processed affects the result. As a heuristic, we have chosen never to accept the result of transforming a variable’s domain if the result of the transformation is that the number of members in the domain is increased, and choose the variable whose transformation leads to the maximum domain size reduction.

3.1 Convergence

We only need to transform the domain of each variable once due to the following theorem:



Theorem 1 *If, after the domain of variable V_1 has been transformed, the domain of a neighbouring variable, V_2 , is also transformed, any subsequent attempt to transform the domain of V_1 will result in no change.*

4 Experimental Results

We tested the algorithm over randomly generated CSPs with eight variables and six values in each domain, varying density from 0.1 to 0.9 with 0.05 increment step, while tightness ranges from 0.1 to 0.8 with the same increment. For each parameter point, the number of constraint checks required to compute all solutions and the number of representations needed to express these solutions are evaluated, averaged over 100 problem instances. The solver is based on MAC-3 with dom/deg variable ordering. The upper bound of the **result** is ten times the size of the variable’s domain.

For the random problem generator, we use model B where density and tightness are fraction instead of probability. Since the transformation process requires arc-consistent CSPs, we also ensure that every problem is arc-consistent by linking each value to at least one value for every constraint in order to guarantee support.

The results are shown in the above graph. While the transformation cannot *guarantee* to reduce the number of constraint checks needed to compute all solutions to a CSP, it can produce huge savings. From the graph, we can see the method works best for problems with low density or low tightness, as either case will lead to a lot of CPR combination. In these cases the work needed for transformed CSPs is less than 1% of the work needed for untransformed CSPs. In those problems where the transformation does not produce large savings, its effect is usually harmless. Indeed, in the worst case we have seen in the experimental data, which occurred only at a few data points, the number of extra constraint checks for a transformed CSP is around 1% more than for the untransformed CSP.

References

- [Freuder, 1991] Eugene C. Freuder. Eliminating interchangeable values in constraint satisfaction problems. In *Proceedings of AAAI’91*, pages 227–233, 1991.
- [Hubbe and Freuder, 1992] Paul D. Hubbe and Eugene C. Freuder. An efficient cross-product representation of the constraint satisfaction problem search space. In *Proceedings of AAAI’92*, pages 421–427, 1992.

Efficient Representation of Adhoc Constraints *

Kenil C.K. Cheng and Jimmy H.M. Lee

Dept of Comp. Sci. & Eng.

The Chinese University of Hong Kong, Hong Kong
{ckcheng, jlee}@cse.cuhk.edu.hk

Peter J. Stuckey

Dept. of Comp. Sci. & Soft. Eng.

University of Melbourne, Australia
pjs@cs.mu.oz.au

1 Introduction

Constraint programming is a promising technique for solving many difficult combinatorial problems. Since real-life constraints can be difficult to describe in symbolic expressions, or provide very weak propagation from their symbolic representation, they are sometimes represented in the form of the sets of solutions or sets of nogoods. This adhoc representation provides strong propagation through domain (generalized arc) consistency techniques. However, the adhoc representation is expensive in terms of memory and computation, when the adhoc constraint is large.

So there is interest in determining less expensive methods for building propagators for adhoc constraints [Frühwirth, 1998; Apt and Monfroy, 1999; Abdennadher and Rigotti, 2000; Barták, 2001; Dao *et al.*, 2002].

In this paper, we propose a new language-independent representation for adhoc constraints, the *box constraint collection*. The idea is to break up an adhoc constraint into pieces and cover these pieces using *box constraints* as tiles. This can be done automatically with a greedy algorithm. With the aid of constructive disjunction and a suitable choice of constraint templates in the collection, our new representation achieves domain consistency.

2 Propagation Based Constraint Solving

In this section we give our terminology for constraint satisfaction problems, and propagation based constraint solving.

An *integer valuation* θ is a mapping of variables to integer values, written $\{x_1 \mapsto d_1, \dots, x_n \mapsto d_n\}$. We extend the valuation θ to map expressions and constraints involving the variables in the natural way. We sometimes treat a valuation $\theta = \{x_1 \mapsto d_1, \dots, x_n \mapsto d_n\}$ as the constraint $x_1 = d_1 \wedge \dots \wedge x_n = d_n$. Let *vars* be the function that returns the set of (free) variables appearing in a constraint or valuation.

A *domain* D is a complete mapping from a fixed (countable) set of variables $\mathcal{V}$ to finite sets of integers. A domain D_1 is *stronger* than a domain D_2 , written $D_1 \sqsubseteq D_2$, if $D_1(x) \subseteq D_2(x)$ for all variables x .

*We thank the anonymous referees for their constructive comments. The work described in this paper was substantially supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region (Project no. CUHK4183/00E).

In an abuse of notation, we define a valuation θ to be an element of a (non-false) domain D , written $\theta \in D$, if $\theta(x_i) \in D(x_i)$ for all $x_i \in \text{vars}(\theta)$.

We are also interested in the notion of an *initial domain*, denoted by D_{init} . The initial domain gives the initial values possible for each variable.

A *constraint* c over variables $x_1, \dots, x_n$, written as $c(x_1, \dots, x_n)$, restricts the values that each variable x_i can take simultaneously. An *adhoc* constraint $c(x_1, \dots, x_n)$ is defined *extensionally* as a set of valuations θ over the variables $x_1, \dots, x_n$. We say $\theta \in c$ is a *solution* of c . For any valuation θ on variables $x_1, \dots, x_n$, with $\theta \notin c$, we call θ a *nogood* of c .

Often we define constraints *intensionally* using some well understood mathematical syntax. For an intensionally defined constraint c we have that $\theta \in c$ iff $\text{vars}(\theta) = \text{vars}(c) \wedge \mathcal{Z} \models_\theta c$. For example the constraint $x_1 = x_2 + 1$ where $D_{init}(x_1) = D_{init}(x_2) = \{1, 2, 3\}$ defines the set of solutions $\{\{x_1 \mapsto 2, x_2 \mapsto 1\}, \{x_1 \mapsto 3, x_2 \mapsto 2\}\}$.

A *constraint satisfaction problem* (CSP) [Tsang, 1993], consists of a set of constraints $c_1, \dots, c_k$ over a set of variables $x_1, \dots, x_n$, where each variable x_i can only take values from its domain $D_{init}(x_i)$, a set of integers. Solving a CSP requires finding a value for each variable from its domain so that no constraint is violated, i.e. all constraints are satisfied.

A *propagator* f is a monotonically decreasing function from domains to domains. The *generalized arc consistent propagator* for a constraint c is defined as $\text{dom}(c)(D)(x) = \{\theta(x) \mid \theta \in D \wedge \theta \in c\}$ where $x \in \text{vars}(c)$, otherwise $\text{dom}(c)(D)(x) = D(x)$. A propagation solver for propagators F repeatedly applies propagators $f \in F$ to a domain D until no further change in D results.

3 Box Constraint Collections

Adhoc constraints are usually implemented as tabled constraints by listing all the solutions or nogoods, incurring space and time overhead. Often we represent a constraint in an adhoc manner because it is difficult (or unwieldy) to describe it using a symbolic expression. However, it may be easier to find symbolic expressions if we examine part of the solution space. Therefore, we propose representing an adhoc constraint c_{adhoc} with a set of simple constraints in DNF.

A *box* $B = \prod_{i=1}^n [a_i..b_i]$ is an n -dimensional hyper-cube, where $[a_i..b_i]$ is a (closed) *interval* of integers a_i and b_i . If

$c(x_1, \dots, x_n)$ is a constraint on variables $x_1, \dots, x_n$, then $\bigwedge_{j=1}^n a_{ij} \leq x_j \leq b_{ij} \wedge c(x_1, \dots, x_n)$ is a *box constraint*, which we write as $B \Rightarrow c$. We restrict the form of constraints $c(x_1, \dots, x_n)$ to certain *templates*. A *box constraint collection* (BCC) is simply a disjunction of box constraints.

The idea is thus to use box constraints in a collection as “tiles” to cover the solution space of an adhoc constraint. The template defining c in a box constraint $B \Rightarrow c$ determines the shape of the tile. Triangles and rectangular boxes are good tile shapes for filling grids. If c is *true*, then $B \Rightarrow c$ is simply the box B . If c is of the form $\sum_{j=1}^n a_j x_j \leq a_0$, then we call $B \Rightarrow c$ a *triangle*.

Lemma 1 Let

$$c_{adhoc}(x_1, \dots, x_n) \equiv \bigvee_{i=1}^m B_i \Rightarrow c_i(x_1, \dots, x_n)$$

and suppose each constraint c_i is implemented by a generalized arc consistent propagator, then using constructive disjunction on this representation achieves generalized arc consistency for c_{adhoc} .

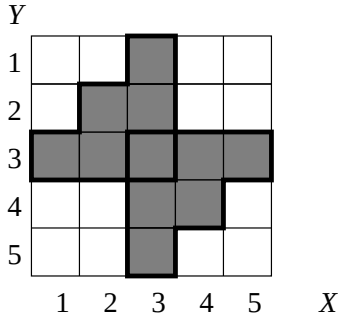


Figure 1: An adhoc constraint c_{tri} made of two triangles

Example 1 A box constraint collection representation of the constraint c_{tri} shown in Figure 1 is

$$[1..3] \times [1..3] \Rightarrow X + Y \geq 4 \vee [3..5] \times [3..5] \Rightarrow X + Y \leq 8$$

□

Due to space limitation, we cannot show how box constraint collections can be compiled into indexicals directly and efficiently.

4 Experiments

We compare the propagation efficiency among *box* (indexical BCCs for boxes only), *tri-box* (indexical BCCs for triangles and boxes) and *rel* (the built-in `relation/3` for binary adhoc constraints) on randomly generated cubic inequalities of the form $d_1 X^3 + d_2 X^2 Y + d_3 X Y^2 + d_4 Y^3 + d_5 X^2 + d_6 X Y + d_7 Y^2 + d_8 X + d_9 Y \leq d_{10}$. The coefficients are randomly chosen between $[-9..9]$. The domain size is 100. For each variable X and Y , we repeat M times picking a subset $S \subseteq D_{init}(x)$ where $|S| = W$, and adding the constraints $x \neq v$ for each $v \in S$. These constraint additions are then removed and the next set S picked. We do our implementation with SICStus Prolog 3.9.1 on a Sun Blade 1000 workstation.

Table 1 summarizes some results. N is the number of solutions. B and T are the number of boxes and triangles. *tri-box* generates no boxes ($B = 0$) in all 3 instances. *gen* is the generation time. *rel* and *prop* (for *box* and *tri-box*) are the time they spend on the propagation test $M = 5000$ and $W = 30$. *tri-box* is the fastest because it compactly represents the non-linear constraints with 1 or 2 triangles. *box*, although is faster than *rel*, it takes a long time to generate because every box covers only a few solutions, and many boxes are needed.

N	<i>rel</i>	<i>box</i>			<i>tri-box</i>		
		B	<i>gen</i>	<i>prop</i>	T	<i>gen</i>	<i>prop</i>
5601	33.78	87	19.17	14.07	2	2.25	6.04
7187	23.58	57	20.97	10.92	1	3.05	3.95
2050	11.59	40	3.43	5.11	2	0.95	4.70

Table 1: Performance comparisons on non-linear constraints

5 Conclusion

We have proposed a new language-independent representation, box constraint collection, for adhoc constraints. With constructive disjunction, our new representation achieves generalized arc consistency, if all constraints inside the collection do.

Future work includes improving the current greedy BCC generation algorithm, and optimizing the indexicals of a box constraint collection.

References

- [Abdennadher and Rigotti, 2000] S. Abdennadher and C. Rigotti. Automatic generation of propagation rules for finite domains. In *CP00*, pages 18–34, 2000.
- [Apt and Monfroy, 1999] K.R. Apt and E. Monfroy. Automatic generation of constraint propagation algorithms for small finite domains. In *CP99*, pages 58–72, 1999.
- [Barták, 2001] R. Barták. Filtering algorithms for tabular constraints. In *CICLOPS 2001*, pages 168–182, 2001.
- [Dao et al., 2002] T.B.H. Dao, A. Lallouet, A. Legtchenko, and L. Martin. Indexical-based solver learning. In *CP02*, pages 541–555, September 2002.
- [Frühwirth, 1998] T. Frühwirth. Theory and practice of constraint handling rules. *Journal of Logic Programming*, 37(1–3):95–138, October 1998.
- [Tsang, 1993] E. Tsang. *Foundations of Constraint Satisfaction*. Academic Press, 1993.

Propagation Redundancy for Permutation Channels *

C.W. Choi and J.H.M. Lee

Dept. of Comp. Sci. & Eng.

The Chinese University of Hong Kong

Shatin, N.T., Hong Kong SAR, China

{cwchoi, jlee}@cse.cuhk.edu.hk

P. J. Stuckey

Dept. of Comp. Sci. & Soft. Eng.

University of Melbourne, 3010

Australia

pjs@cs.mu.oz.au

1 Introduction

Finding a good model of a constraint satisfaction problem (CSP) is a challenging task. A modeller must specify a set of constraints that capture the definitions of the problem, and the model should also have strong propagation. In other words, the model should be able to quickly reduce the domains of the variables of the problem, *and* the implementation of these propagators should be efficient, *and* the search space should not be too large.

A problem can be modelled differently from two viewpoints using two different sets of variables. In redundant modelling [Cheng *et al.*, 1999], we connect the two different models with channelling constraints, which relates valuations in the two different models stronger propagation behaviour can be observed. However, the additional variables and constraints impose extra computation overhead may outweigh the gain of reduction in search space.

In this paper we consider redundant models connected by permutation channels, which commonly arise when the underlying problem is some form of assignment problem. Since each model is complete and only admits the solutions of the problem, each model is logically redundant with respect to the other model plus the permutation channel. In order to keep the benefits of redundant modelling without paying all the costs, We give a theorem which allows us to determine when we can eliminate constraints in the mutually redundant models that do not give extra propagation. Due to space limitations, we state the theorem without proof.

2 Reasoning about Domain Propagation

We consider integer constraint solving with constraint propagation and tree search.

An *integer valuation* θ is a mapping of variables to integer values, written $\{x_1 \mapsto d_1, \dots, x_n \mapsto d_n\}$. Let *vars* be the function that returns the set of variables appearing in a constraint or valuation. A *constraint* c defines a set of valuations $\text{solns}(c)$ each mapping the same set of variables $\text{vars}(c)$. We

call $\text{solns}(c)$ the *solutions* of c . A constraint c is *logically redundant* with respect to a set of constraints C if $\models C \rightarrow c$.

A *domain* D is a complete mapping from a fixed (countable) set of variables $\mathcal{V}$ to finite sets of integers. A *false domain* D is a domain with $D(x) = \emptyset$ for some x . A domain D_1 is *stronger* than a domain D_2 , written $D_1 \sqsubseteq D_2$, if D_1 is a false domain or $D_1(x) \subseteq D_2(x)$ for all variables x . The initial domain D_{init} gives the initial values possible for each variable, allows us to restrict attention to domains D such that $D \sqsubseteq D_{\text{init}}$.

We adopt the notion of *propagation solver* and *domain consistency*¹ from Schulte and Stuckey [2001]. A *propagator* f is a monotonically decreasing function from domains to domains. A *propagation solver* for a set of propagators F and current domain D , $\text{soln}(F, D)$, repeatedly applies all the propagators in F starting from domain D until there is no further change in resulting domain. A domain D is *domain consistent* if D is the least domain containing all solutions of c in D . Define the *domain consistency propagator* $\text{dom}(c)$ for a constraint c such that $\text{soln}(\text{dom}(c), D)$ is always domain consistent for c .

For all domains $D \sqsubseteq D_{\text{init}}$, a set of propagator F_2 is made *propagation redundant* by a set of propagators F_1 , written $F_1 \gg F_2$, if $\text{soln}(F_1, D) \sqsubseteq \text{soln}(F_2, D)$, and is *equivalent* to F_1 , written $F_1 \approx F_2$, if $\text{soln}(F_1, D) = \text{soln}(F_2, D)$.

It is well known that in general the domain propagation of a conjunction of constraints is not equivalent to applying the domain propagators individually. But there are cases where propagation of a conjunction is equivalent to propagation on the individual conjuncts.

Lemma 1 *If c_1 and c_2 share at most one variable x , then $\{\text{dom}(c_1), \text{dom}(c_2)\} \approx \{\text{dom}(c_1 \wedge c_2)\}$.*

An *atomic constraint* is one of $x_i = d$ or $x_i \neq d$ where $x_i \in \mathcal{V}$ and d is an integer. An atomic constraint represents the basic changes in domain that occur during propagation.

A *propagation rule* is of the form $C \mapsto c$ where C is a conjunction of atomic constraints, c is an atomic constraint and $\not\models C \rightarrow c$. Note our propagation rules are similar to the “membership rules” of Apt and Monfroy [2001] except we allow equations on the right hand side.

A propagator f *implements* a propagation rule $C \mapsto c$ if for each $D \sqsubseteq D_{\text{init}}$ whenever $\models D \rightarrow C$, then $\models f(D) \rightarrow c$.

¹Equivalently, hyper-arc or generalized arc consistent.

*We thank Barbara Smith and the anonymous referees for their constructive comments. The work described in this paper was substantially supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region (Project no. CUHK4204/01E).

We can characterize a propagator f in terms of the propagation rules that it implements. Let $rules(f)$ be the set of rules implemented by f . Then $prop(f) \subseteq rules(f)$ are a set of propagation rules such that every $r \in rules(f)$ is subsumed by a rule $r' \in prop(f)$.

3 Permutation Channels

A common form of redundant modelling is when we consider two viewpoints to a permutation problem. We can view the problem as finding a bipartite matching between two sets of objects of the same size. For notational convenience, let the two viewpoints as having the set of variables $X = \{x_0, \dots, x_n\}$, and $Y = \{y_0, \dots, y_n\}$ respectively.

The *permutation channel* $C_{\bowtie}$ is defined by the conjunction of constraints $\bigwedge_{i=0}^n \bigwedge_{j=0}^n (x_i = j \Leftrightarrow y_j = i)$. The *permutation channel propagator* $F_{\bowtie}$ maintains domain consistency of each individual bi-implication, that is $\bigcup_{i=0}^n \bigcup_{j=0}^n \{dom(x_i = j \Leftrightarrow y_j = i)\}$.

Smith [2000] first observes that the permutation channel makes each of the disequations between variables in either model propagation redundant. Walsh [2001] proves this holds for other notions of consistency.

Lemma 2 (Walsh, 2001) $F_{\bowtie} \gg \{dom(x_i \neq x_k)\}$

Related to $C_{\bowtie}$ is the *permutation channel function* $\bowtie$ which is a bijection between atomic constraints in X to atomic constraints in Y , $\bowtie(x_i = j) = (y_j = i)$, and $\bowtie(x_i \neq j) = (y_j \neq i)$. We extend $\bowtie$ to map conjunctions of constraints in the obvious manner $\bowtie(C_1 \wedge C_2) = \bowtie(C_1) \wedge \bowtie(C_2)$.

The fundamental theorem states that a constraint in Y is propagation redundant if there exist a constraint in X when conjuncts with $C_{\bowtie}$ logically imply every propagation rules implemented by the constraint in Y . Since $\bowtie$ is bijective, the theorem is valid when X and Y are reversed.

Theorem 3 Let f_Y be a propagator on Y , and c_X be a constraint on X . If $\models (D_{init} \wedge c_X \wedge \bowtie(C)) \rightarrow \bowtie(c)$ for all $(C \mapsto c) \in prop(f_Y)$, then $\{dom(c_X)\} \cup F_{\bowtie} \gg \{f_Y\}$.

Example 4 Smith [2000] suggests two ways to model the Langford's problem as a permutation problem and how to combine them with the permutation channel. She points out that the so-called *minimal combined model*, which includes only X model and the permutation channel, gives as much pruning as the full combined model. This is proved in an ad hoc manner by Choi and Lee [2002]. We prove this formally using our generic approaches.²

The disequation constraints are propagation redundant by Lemma 2. Consider the separation constraints $c_Y \equiv y_j = 3i \Leftrightarrow y_{j+(i+2)} = 3i + 1$. The propagation rules for $dom(c_Y)$ are (r1) $y_j = 3i \mapsto y_{j+(i+2)} = 3i + 1$, (r2) $y_{j+(i+2)} = 3i + 1 \mapsto y_j = 3i$, (r3) $y_j \neq 3i \mapsto y_{j+(i+2)} \neq 3i + 1$, and (r4) $y_{j+(i+2)} \neq 3i + 1 \mapsto y_j \neq 3i$. We have that for $c_X \equiv x_{3i+1} = x_{3i} + (i+2)$, $\models (D_{init} \wedge c_X \wedge \bowtie(C)) \rightarrow \bowtie(c)$ for all propagation rules above. For example (r1) is mapped to $x_{3i} = j \mapsto x_{3i+1} = j + i + 2$. Hence the conditions of Theorem 3 hold and $dom(c_Y)$ is propagation redundant.

²The complete description of the two permutation models for the Langford's Problem can be found in [Choi and Lee, 2002].

Note the importance of *equational* propagation rules such as $y_j = 3i \mapsto y_{j+(i+2)} = 3i + 1$. If we only allowed disequations on the right hand side, we would replace this with rules $y_j = 3i \mapsto y_{j+(i+2)} \neq k, 0 \leq k \neq 3i + 1 \leq n$. It is impossible to prove that the translated versions are implied by $D_{init} \wedge c_X$.

We can similarly show that each of the constraints $c'_Y \equiv y_j = 3i \Leftrightarrow y_{j+2(i+2)} = 3i + 2$ are propagation redundant using $c_X \wedge c'_X$, where $c'_X \equiv x_{3i+2} = x_{3i+1} + (i+2)$ by Theorem 3. Although model M_X does not include a domain propagator for $c_X \wedge c'_X$, we can still show propagation redundancy since $\{dom(c_X), dom(c'_X)\} \approx \{dom(c_X \wedge c'_X)\}$ by Lemma 1.

Similar reasoning applies to show that each constraint $y_j \neq 3i$ where $0 \leq i \leq 8$ and $27 - 2(i+2) \leq j \leq 26$ is made propagation redundant by $(x_{3i+1} = x_{3i} + (i+2)) \wedge (x_{3i+2} = x_{3i+1} + (i+2)) \wedge (0 \leq x_{3i+2} \leq 26)$. $\square$

4 Conclusion

We have extend our approach to other types of channelling constraints and lead to significantly faster models that do not increase the search space. Although we have illustrated the use of the theorems herein by hand, the approach can clearly be automated. We can constructs the propagation rules automatically using the approach of Abdennadher and Rigotti [2002]. We are interested in extending the work to reason bounds propagation. Another direction is to study a weaker notion of propagation redundancy which allows removal of constraints without affecting the search space given a specific search heuristic.

References

- [Abdennadher and Rigotti, 2002] S. Abdennadher and C. Rigotti. Automatic generation of rule-based solvers for intentionally defined constraints. *IJTAI*, 11(2):283–302, 2002.
- [Apt and Monfroy, 2001] K. Apt and E. Monfroy. Constraint programming viewed as rule-based programming. *TPLP*, 1(6):713–750, 2001.
- [Cheng et al., 1999] B.M.W. Cheng, K.M.F. Choi, J.H.M. Lee, and J.C.K. Wu. Increasing constraint propagation by redundant modelling: an experience report. *Constraints*, 4(2):167–192, 1999.
- [Choi and Lee, 2002] C.W. Choi and J.H.M. Lee. On the pruning behaviour of minimal combined models for permutation CSPs. In *International Workshop on Reformulating Constraint Satisfaction Problems*, 2002.
- [Schulte and Stuckey, 2001] C. Schulte and P. J. Stuckey. When do bounds and domain propagation lead to the same search space. In *PPDP 2001*, pages 115–126, 2001.
- [Smith, 2000] B. M. Smith. Modelling a permutation problem. Research Report 2000.18, School of Computer Studies, University of Leeds, 2000.
- [Walsh, 2001] T. Walsh. Permutation problems and channelling constraints. In *LPAR 2001*, pages 377–391, 2001.

Channeling Constraints and Value Ordering in the QuasiGroup Completion Problem

Iván Dotú and Alvaro del Val and Manuel Cebrián

Departamento de Ingeniería Informática
Universidad Autónoma de Madrid

1 Introduction

The Quasigroup Completion Problem (QCP) is a very challenging benchmark among combinatorial problems, and a focus of much recent interest in the area of constraint programming. It has numerous practical applications [Gomes *et al.*, 2002]; it has been put forward as a benchmark which can bridge the gap between purely random instances and highly structured problems [Gomes *et al.*, 2002]; and its structure as a multiple permutation problem [Walsh, 2001] is common to many other important problems in constraint satisfaction.

[Gomes *et al.*, 2002] reports that QCPs of order 40 could not be solved by pure constraint programming approaches, but could sometimes be solved by hybrid approaches combining constraint programming with mixed integer programming techniques from operations research. In this paper, we show that the pure constraint satisfaction approach can solve many problems of order 45 in the transition phase, which corresponds to the peak of difficulty. Our solution combines a number of known ideas –the use of redundant modeling [Cheng *et al.*, 1996] with primal and dual models of the problem connected by channeling constraints [Walsh, 2001]– with some novel aspects, as well as a new and very effective value ordering heuristic.

2 Models for QCPs

A quasigroup is an ordered pair $(Q, \cdot)$, where Q is a set and $\cdot$ is a binary operation on Q such that the equations $a \cdot x = b$ and $y \cdot a = b$ are uniquely solvable for every pair of elements a, b in Q [Gomes *et al.*, 2002]. The order n of the quasigroup is the cardinality of the set Q . A quasigroup can be seen as an $n \times n$ multiplication table defining a Latin Square, which must be filled with “colors” so that the colors of each row are all distinct, and similarly for columns. A QCP is the problem of coloring a partially filled Latin square, which is known to be NP-Complete [Colbourn, 1984].

Each row and column of a Latin Square defines a permutation problem, i.e. a constraint satisfaction problem with the same number of variables as values, where a solution is a permutation of the values [Walsh, 2001]. The QCP is a multiple permutation problem with $2n$ intersecting permutation constraints (n row permutation constraints and n column permutation constraints). In order to represent those permutation problems we have implemented three models:

- *Primal Model*: This model has a variable $x_{i,j}$ for each cell of the latin square, with i and j its coordinates. Their possible values are the colors of the cell.
- *Row Dual Model*: This is the first dual model, it represents the colors in each row; the dual variable $r_{i,k}$ is the k -th color in the i -th row, and the values it can take are the columns where it can be placed.
- *Column Dual Model*: This one models the colors in each column; the dual variable $c_{j,k}$ represents the k -th color in the j -th column, with rows as its possible values.

The primal and dual not-equal constraints specify that the colors in a row or column must all be different. While these are sufficient to capture the problem (and in fact redundantly so), it is known that these models can be improved by connecting them through *channeling constraints*:

- *Row Channeling Constraints*: Constraints for the n row permutation constraints, linking the primal model with the row dual model: $x_{i,j} = k \Leftrightarrow r_{i,k} = j$.
- *Column Channeling Constraints*: Corresponding to the n column permutation constraints, they link the primal and dual column models: $x_{i,j} = k \Leftrightarrow c_{j,k} = i$.
- *Triangular Channeling Constraints*: These (novel) constraints link both dual models, closing a “triangle” among the three models: $c_{j,k} = i \Leftrightarrow r_{i,k} = j$.

It’s easy to show that the first two kinds of channeling constraint fully capture a QCP (see also [Walsh, 2001]), making the primal and dual constraints redundant.

3 Value Ordering

Key to our results is a new *value ordering* heuristic, which we might call the *min-domain value selection heuristic*. Once a primal or dual variable is selected, we need to choose a value for it. Since any such value corresponds to one specific variable from each of the two other models, we select the value whose corresponding two variables have a minimal “combined” domain. Specifically, say we have chosen $x_{i,j}$. Then we choose a color k from its currently active domain for which the sum of the current domain sizes of $r_{i,k}$ and $c_{j,k}$ is minimal among the currently available colors for $x_{i,j}$.

4 Results

We focused our experiments on solvable instances. To make our results comparable with those reported in [Gomes *et al.*,

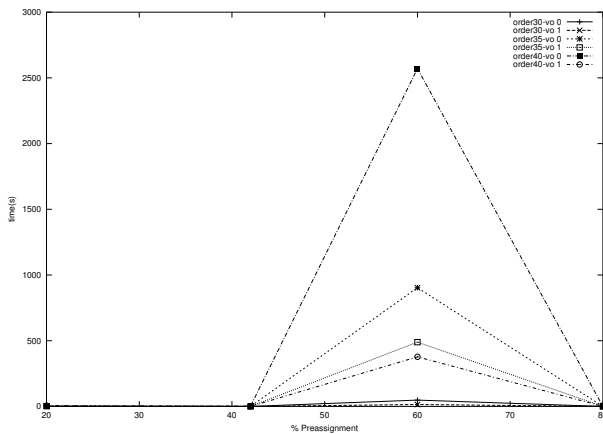


Figure 1: Mean solution time on QCPs of order 30, 35 and 40 with (vo1) and without value ordering (vo0).

2002], we used the same generator¹, and considered only “balanced” instances, which are known to be the hardest. We used an (AC3-based) Generalized Arc Consistency algorithm based on van Beek’s CPLAN implementation (<http://ai.uwaterloo.ca/~vanbeek/software/software.html>), with a few improvements. We considered various (complete) combinations of all the above mentioned constraints, and found that adding the primal and dual “not-equal” constraints never helped performance in the presence of row and column channeling constraints, and yielded even worse results in their absence. Most of the results discussed below therefore use a model with only row and column channeling constraints; they also use “MAC” (maintaining arc consistency [Bessiere, Regin, 1996]). Since our preliminary tests showed that CBJ and nogood learning seem to never help in these problems, and are often a source of significant overhead, we discarded them. We also considered various variable selection heuristics. We found the *min-domain* heuristic, which selects a variable with smallest domain to be uniformly the best among the alternatives we tried, which included more sophisticated heuristics such as *dom+degree*, *dom/degree*, and variants of the above which took into account the (primal or dual) model to which variables belong, e.g. selecting only among primal variables, etc.

With only row and column channeling constraints, our implementation outperforms classical approaches, in that we can solve a considerable amount of instances of order 40 in a reasonable time. As mentioned above, this was very recently considered as out of reach for pure constraint programming approaches [Gomes *et al.*, 2002], which ours is (in fact, we don’t even use global or non-binary constraints). The results when this model was used with the min-domain value ordering heuristic were quite surprising, as it outperformed previous tests in three orders of magnitude in some cases. For example, for the instance bqwh-35-405-5.pls (balanced instance of order 35 and 60% preassigned) it took 2905 secs without value ordering and only 0.40 secs with it. In Figure 1 we present a comparison of our results with and without value

¹Generator *lsencode*, kindly provided by Carla Gomes.

ordering for quasigroups of order 30,35 and 40, and 60% preassignment, close to the phase transition [Gomes *et al.*, 2002].

Encouraged by this performance, we generated some instances of order 45 and 60% preassignment. Again the value ordering heuristic turns out to be so efficient that we were able to solve a great number of instances in a reasonable amount of time. The next table shows median and mean time in seconds (the latter taken only over solved instances) and percent of solved instances, using value ordering for problems in the transition phase.

order	mean	median	% solved	timeout
30	148.84	174.11	68%	1000
35	533.43	163.48	84%	3600
40	690.85	842.11	68%	5000
45	1090.10	2971.40	56%	6000

Finally, we are currently exploring the effect of adding “triangular” channeling constraints; our preliminary results suggest that they can improve performance by an additional 30–35%, using in this case only forward checking (e.g. from 902 to 666 seconds in one order 45/60% preassigned instance). The idea is that they provably allow forward checking to perform as much pruning as arc-consistency, while saving many useless checks; in fact, as many as 60% fewer checks, even though the savings in time are not as great due to effects we are currently exploring.

5 Conclusions and Future Work

We have shown that a pure CSP approach can handle quasigroup completion problems significantly larger than was thought possible, using appropriate models and value ordering heuristics, and even in the absence of global alldiff constraints. Yet our results are preliminary, and no doubt it would be interesting to consider other value ordering heuristics and ways to make the triangular channeling constraints most effective.

References

- [Gomes *et al.*, 2002] C. Gomes and D. B. Shmoys. The promise of LP to Boost CSP Techniques for Combinatorial Problems. In *Proc. of CPAIOR’02*, Le Croisic, March 2002.
- [Gomes *et al.*, 2002] C. Gomes and D. B. Shmoys. Completing Quasigroups or Latin Squares: A Structured Graph Coloring Problem In *Proc. Computational Symposium on Graph Coloring and Generalizations*, 2002.
- [Walsh, 2001] T. Walsh. Permutation Problems and Channeling Constraints. TR 26, APES Group, 2001.
- [Colbourn, 1984] C. Colbourn. The complexity of completing partial latin squares. In *Discrete applied Mathematics*, (8):25–30, 1984.
- [Cheng *et al.*, 1996] B.M.W. Cheng, J.H.M. Lee, and J.C.K. Wu. Speeding up constraint propagation by redundant modeling. In *2nd Int. Conf. on Principles and Practice of Constraint Programming*, pp. 91–103, 1996.
- [Bessiere, Regin, 1996] C.Bessiere and J.C.Regina Mac and combined heuristics: two reason to forsake FC (and CBJ?) on hard problems. In *CP’96*, pp. 61–75, Cambridge MA.

Making the Breakout Algorithm Complete Using Systematic Search

Carlos Eisenberg and Boi Faltings

Swiss Federal Institute of Technology (EPFL)

IN-Ecublens, 1015 Lausanne, Switzerland

{eisenberg,faltings}@lia.di.epfl.ch

Abstract

Local search algorithms have been very successful for solving constraint satisfaction problems (CSP). However, a major weakness has been that local search is unable to detect unsolvability and is thus not suitable for tightly and overconstrained problems. We present a hybrid solving scheme where we combine a local search algorithm - the breakout algorithm, with a systematic search algorithm - backtracking. The breakout algorithm is used for identifying hard or unsolvable subproblems and the backtracking algorithm proves the solvability of these subproblems. The resulting hybrid algorithm is complete and is tested on randomly generated graph 3-colouring problems. The algorithm performs extremely well for all areas of the phase transition and outperforms the individual methods by several orders of magnitude.

1 Introduction

The breakout algorithm is an efficient, local search algorithm for solving Constraint Satisfaction Problems (CSPs). The roots of the algorithm go back to [Minton *et al.*, 1992] and to [Morris, 1993]. [Minton *et al.*, 1992], developed, a local search algorithm, called min-conflict heuristic, which iteratively repairs a given assignment until all conflicts are eliminated. One drawback of the method is that it can get caught in local, non solution minima. Morris eliminated this drawback by extending the algorithm with a breakout method, which allows to escape from local, non solution-minima. The strengths of the breakout algorithm are simplicity, robustness, low memory requirement and high efficiency for solving under constrained problems. [Minton *et al.*, 1992] demonstrated the performance superiority of the min-conflict heuristic by solving large-scale scheduling problems; it performed by orders of magnitude better than traditional backtracking techniques. However, the major weak point of the algorithm is its incompleteness.

In this paper we present a hybrid algorithm that is complete where we combine a local search algorithm, the breakout algorithm, with a complete search algorithm, backtracking. The combination of the two algorithms compensates each others weakness to deal with under and tightly constrained

problems. We discover that the algorithm combination also leads to synergies. The weight information from the breakout algorithm can locate and order hard or unsolvable subproblems and guide backtracking by a fail-first variable order. In addition, we introduce a weight sum constraint that can be used to identify unsolvable subproblems of a certain size.

2 The Scheme

When a problem contains an unsolvable subproblem, we observe that the average constraint weight of a constraint belonging to an unsolvable subproblem is higher than the average constraint weight of the problem. In addition, the smaller the unsolvable subproblem, the higher the average constraint weight. This observation is supported by Lemma 1:

Lemma 1: *After increasing the weights n times, the sum of the constraint weights w_{P_k} of an unsolvable subproblem P_k with k constraints must be greater than or equal to $n + k$.*

Proof: If a subproblem is unsolvable, then in every breakout step, one or more of the subproblem constraints is always violated, thus every time the algorithm is caught in a local non solution minimum, one or more of the corresponding weights must be increased. The lower bound of w_{P_k} can be derived by assuming that in every breakout step only one constraint is violated.

By applying Lemma 1 we can define a weight sum constraint, which is extremely useful for tracing unsolvable subproblems of different sizes.

Definition 1 (Weight sum constraint for an unsolvable subproblem P_k): *According to Lemma 1, in an unsolvable subproblem P_k consisting of k constraints after n breakout steps, the following condition*

$$\sum_{i=1}^k w(c_i) \geq n + k$$

is satisfied, where c_i are all the constraints of the constraint set C_{P_k} of the subproblem P_k .

The weight sum constraint is a powerful tool for pruning the search for unsolvable subproblems. For example, if we search for an unsolvable subproblem of size 3, we only have to consider the constraints whose weight sum is greater $n + 3$.

We also observe that the average constraint weight of an unsolvable subproblem of size k is higher than that of an unsolvable subproblem of size $k + s$ ($s > 0$). This is due to

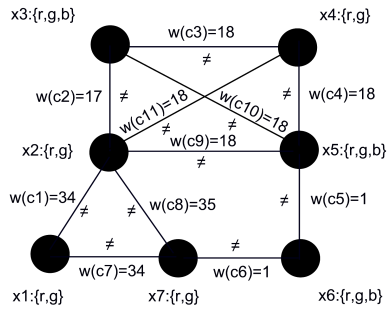


Figure 1: The weight graph of an unsolvable graph colouring problem after 100 breakout steps, containing two unsolvable subproblems of size 3 (c1,c7,c8) and 6 (c2,c3,c4,c9,c10,c11).

the fact that violations cycle over the constraint graph and the less constraints are available, the greater the average constraint weight value. This is demonstrated in Figure 1, containing two unsolvable subproblems of size 3 and 6. After $n=100$ breakout steps, the average constraint weight of the subproblem of size 3 is approximately $n/3 + 1 = 34$ and for the subproblem of size 6, it is $n/6 + 1 = 18$. Sorting the variables according to increasing constraint weights, orders the subproblem of size 3 before the subproblem of size 4, and thus separates the two.

These observations inspired us to use the constraint weight information for localizing potentially unsolvable subproblems and to derive a fail-first variable ordering heuristic for backtracking. In this heuristic, variables are sorted so that the constraints with the highest weights are treated first.

2.1 Hybrid Solver BOBT

We have implemented this scheme into a hybrid algorithm (BOBT), where we combine the breakout algorithm (BO) with backtracking (BT). In this algorithm, we first execute BO and terminate if no solution is available after a bounded number of breakout steps. Then we derive a fail-first variable order from the constraint weights and the graph structure and start backtrack search. BT will then either prove that the problem is unsolvable, or return a solution. This method guarantees the completeness of the hybrid algorithm. In the case where we find an unsolvable subproblem, we terminate the algorithm and give a failure explanation by returning the minimal unsolvable subproblem. The minimal unsolvable subproblem is computed from the unsolvable subproblem using the technique described in [Faltings, 2002].

3 Experiments and Results

For testing the scheme, we generated a large set of 10,000 random graph 3-colouring problems according to the method described in [Davenport et al. 1995]. The generated problem graphs consisted of 30 variables with a connectivity of 2-6. The ratio of solvable to unsolvable problems is 1:1. Figure 2 shows the results of the experiments. We show the number of constraint checks for the three algorithms, BO, BT and BOBT on a logarithmic scale over the graph connectivity. For unsolvable problems, we bound the maximum number of iterations for the BO to $4.37 \cdot 10^5$.

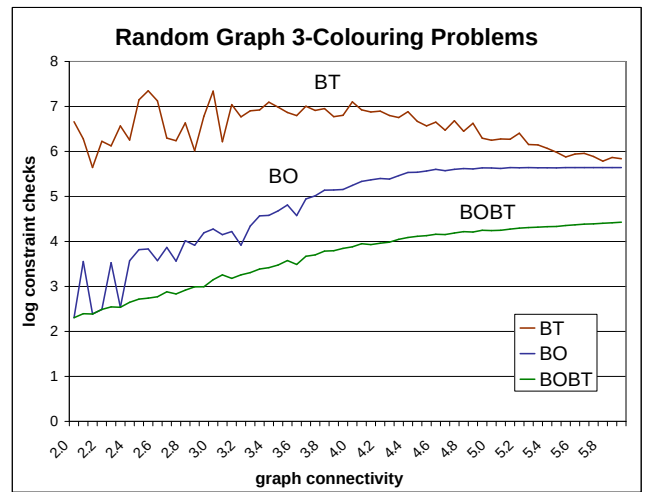


Figure 2: Number of constraint checks on a logarithmic scale for solving 10,000 random generated, 30 node graph 3-colouring problems, solved with BO, BT and BOBT.

4 Conclusion

The main contribution of this paper is the presented hybrid scheme that combines the breakout algorithm with a systematic search method, backtracking, and results in a new algorithm that is complete. In the scheme we use the constraint weight information of the breakout algorithm to identify hard and unsolvable subproblems of increasing sizes and derive in combination with the graph structure a fail-first variable order for a backtracking algorithm. With our results we prove, that the new hybrid search scheme performs extremely well. BOBT needs orders of magnitude less constraint checks than BO and BT. In the future we plan to extend the scheme by identifying the ordered set of all unsolvable sub problems in order to perform a spectral analysis that gives us the distribution of unsolvable subproblems for random graph colouring problems.

References

- [Davenport et al., 1992] Andrew Davenport and Edward Tsang. *An empirical investigation into the exceptionally hard problems*. Technical Report CSM-239, Department of Computer Science, University of Essex, U.K., 1995.
- [Faltings, 2002] Boi Faltings and Macho-Gonzalez. Open Constraint Satisfaction. *Proceedings of the 8th International Conference, CP 2002* Ithaca, NY, USA, Sept. 2002.
- [Minton et al., 1992] Steven Minton and Mark D. Johnston and Andrew B. Philips and Philip Laird. *Minimizing Conflicts: A Heuristic Repair Method for Constraint Satisfaction and Scheduling Problems*. Artificial Intelligence, Vol. 58, P. 161-205, 1992.
- [Morris, 1993] Paul Morris. The breakout method for escaping from local minima. *Proc. of the 11th National Conf. on Artificial Intelligence*, Washington, DC, pp. 40–45, 1993.

Sampling Combinatorial Spaces Using Biased Random Walks

Jordan Erenrich and Bart Selman*

Dept. of Computer Science
Cornell University
{erenrich, selman}@cs.cornell.edu

Abstract

For probabilistic reasoning, one often needs to sample from a combinatorial space. For example, one may need to sample uniformly from the space of all satisfying assignments. Can state-of-the-art search procedures for SAT be used to sample effectively from the space of all solutions? And, if so, how uniformly do they sample? Our initial results find that on the positive side, one can find all solutions to a given instance. Nevertheless, sampling can be highly biased. This research provides a starting point for the development of more balanced procedures.

1 Introduction

There has been much recent progress in the area of algorithms for solving Boolean satisfiability problems. Current methods can handle problem instances with up to a million variables. The emphasis in this work has been on the decision version of the satisfiability problem. A problem instance is either shown satisfiable by producing a satisfiable assignment or by proving that no such solution exists. A natural extension of this problem, for the satisfiable case, is to consider whether one can sample from the set of all satisfying assignments. An effective sampling procedure would be of use in a large range of probabilistic reasoning tasks. In the worst-case, the sampling problem is even harder than the satisfiability problem [Roth, 1996]. However, this leaves open the question how of hard it is to sample solutions from practical problem instances. In particular, can state-of-the-art random walk style procedures for SAT, such as WalkSAT [Selman et al., 1994], be used to sample from the space of satisfying assignments?

We investigated this question, and our key findings are as follows. On reasonable size problem instances, the WalkSAT procedure does reach all satisfying assignments. However, the sampling can be highly non-uniform (several orders of magnitude in the probability of reaching solutions). Quite unexpectedly, we found that sampling on our structured problem instance is much more uniform than on our hard random problem instance. This shows that, in a sense, the search on

a random problem instance is more complex. We also considered the question of how to improve the uniformity of our sampling approach. Our results show, again somewhat surprisingly, that coverage does not become more uniform with more noise injected in the random walk. (Of course, the effectiveness of finding solutions degrades, but apparently, this degradation is proportional for all solutions.)

Because of space limitations, we present only some of our key data. We hope this work will lead to a better understanding of random walk style procedures for sampling solutions, including possible ways for getting more uniform sampling.

Our work should be contrasted with the more standard Markov Chain Monte Carlo (MCMC) approaches, such as simulated annealing (SA). For MCMC methods, one can show that the underlying Markov Chain eventually reaches an equilibrium distribution (although this can take exponential time). Unfortunately, formal guarantees are of little practical value. For example, even for finding a single satisfiable assignment, SA has been shown to be orders of magnitude slower than WalkSAT [Selman et al., 1994] (even ignoring the issue of whether SA reaches the stationary distribution). The new class of local search methods for SAT are based on ideas from biased random walks. These methods are designed to reach some solution as quickly as possible. Such strategies have proven to be quite effective, but they inherently exploit the fact that they operate far away from equilibrium. Nevertheless, as our results show, they can reach all solutions on non-trivial problem instances. Moreover, our work may suggest ways of making the sampling more uniform, possibly leading to a new more powerful class of sampling methods.

2 Sampling Experiments

We consider a hard random 3-SAT instance and a more structured problem from a planning application. In order to generate a random 3-SAT instance, we first created 100 uniform random 3-SAT instances, with 70 variables and 301 clauses. We chose an instance with 2531 solutions for our detailed

Table 1: Summary Statistics from Coverage Experiments

INSTANCE	RUNS	HITS	HITS	COM.-RARE
		RAREST	COMMON	RATIO
RANDOM	$50 \cdot 10^6$	53	$9 \cdot 10^5$	$1.7 \cdot 10^4$
LOGISTIC	$1 \cdot 10^6$	84	$4 \cdot 10^3$	50

*Research supported in part by AFOSR, Darpa, and NSF

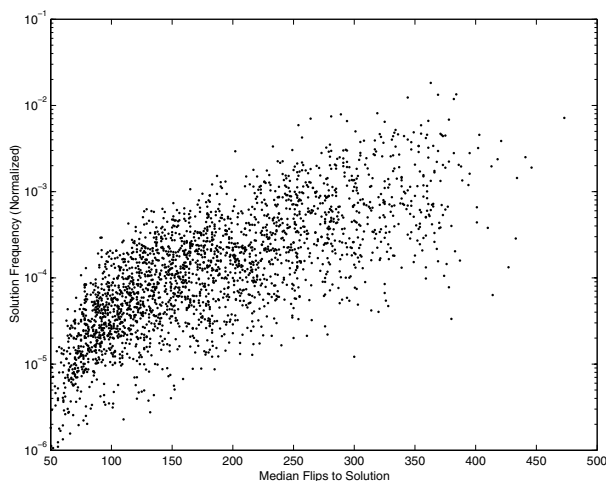


Figure 1: Frequency of finding a specific soln. vs. the median number of flips to find this solution for a 3-SAT problem.

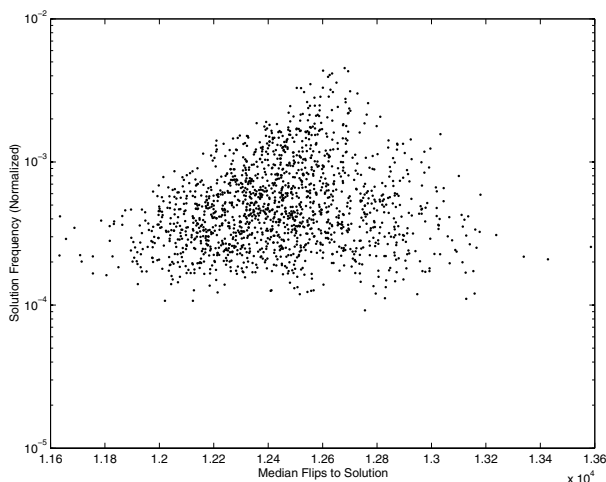


Figure 2: Frequency of finding a specific soln. vs. the median number of flips to find this solution for a logistic problem.

experiments. (Relsat [Bayardo and Pehoushek 2000] was used to systematically generate all solutions.) For the logistics planning formula, we started with *logistics.d.cnf* available from Satlib with over 10^{10} solutions. To facilitate solution set sampling experiments, we added 52 unit clauses, constraining the instance to 1600 solutions.

For the 3-SAT and logistic instances, we performed 50e6 and 1e6 runs of WalkSAT, respectively. All of these runs successfully found a solution, and WalkSAT ultimately found every solution for both instances. Summary statistics for these experiments are available in Table 1.

Figures 1 and 2 display the relationship between “solution frequency” (how often a particular solution is reached) and “speed” (the median number of flips to reach that solution). Each point represents a unique solution. The figure further confirms the results from Table 1: WalkSAT’s solution sampling can be highly non-uniform. Notice the scale

on the y-axis. For the random instance we see a range over 4 orders of magnitude. For the logistic problem, we see that the solution sampling is also biased, albeit much less than on the 3-SAT problem. The figure also indicates that on the 3-SAT problem, rare solutions are typically found with fewer flips than common solutions. No such relation is evident for the logistic problem in Figure 2B.

3 Solution Space Structure

In this section, we attempt to gain insight into the structure of the solution space by analyzing hamming distances between solutions. For both the logistics and 3-SAT problems, we computed the hamming distance between all solutions. Thus, we generated $O(n^2)$ distances per problem, where n is the number of problem solutions.

Figures 3A and B show histograms of these distances for the 3-SAT and logistics problem, respectively. The 3-SAT hamming distance distribution is trimodal, while the logistics problem gives a unimodal distribution. These distributions suggest that the 3-SAT instance’s solution space has multiple clusters of solutions, while the logistic problem’s solution space has only a single cluster of solutions.

Of course, it is inherently difficult to analyze structure within these 70+ dimension solution spaces, and many other underlying structures could explain the results seen in Figure 3. Nevertheless, we believe that these figures are highly suggestive.

4 Obtaining More Uniform Coverage

In the previous experiment with the 3-SAT instance, we observed that rare solutions are generally found with fewer flips than common solutions. Intuitively, one needs to start relatively close to such a rare solution to find it. Following up on this observation, we decreased the cutoff value (number of flips until restart) for Walksat. As shown in Figure 4A, our experiments indicate that a lower cutoff indeed makes the sampling more uniform on the 3-SAT instance. Unfortunately, as a side effect, the median number of flips to reach a solution goes up. Thus, there appears to be a tradeoff between speed and uniformity on the 3-SAT instance. Note that theoretically, a cutoff of 0 (no flips) gives us purely uniform sampling. However, such a strategy is clearly infeasible for any instance of practical interest. (A cutoff below 40 makes it practically infeasible to find all solutions for our random problem instance.) Varying the cutoff used for the logistic problem did not have an observable effect on uniformity of sampling, as expected.

Intuitively, increasing WalkSAT’s noise level would lead to a less greedy and more “erratic” search. It would therefore be reasonable to expect more uniform sampling of solutions. Quite surprisingly however, as shown in Figure 4B, the frequency of the most common and the rarest solution is hardly affected by the noise level. This unexpected finding requires further study. It also means that we need to develop other ways to improve WalkSAT’s sampling. We obtained similar results for the logistic problem, which are not shown due to space constraints.

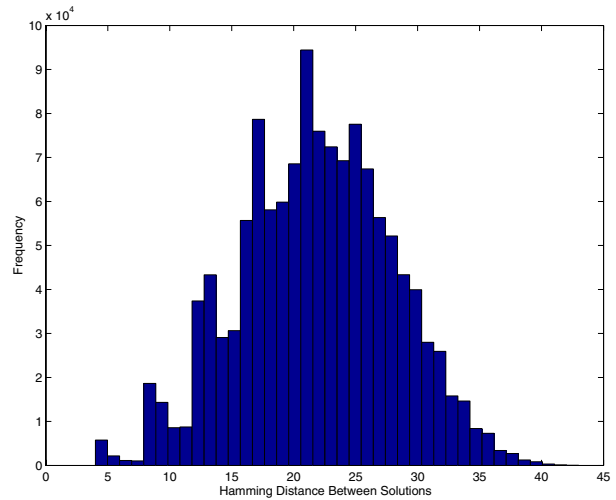
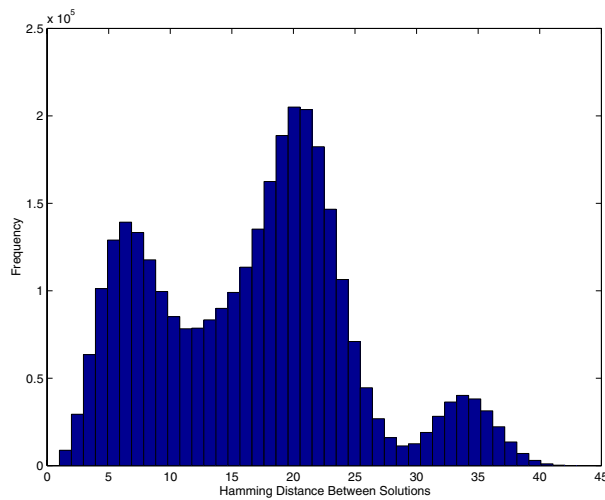


Figure 3: Hamming distances between all solutions for a (A) 3-SAT and (B) logistics problem

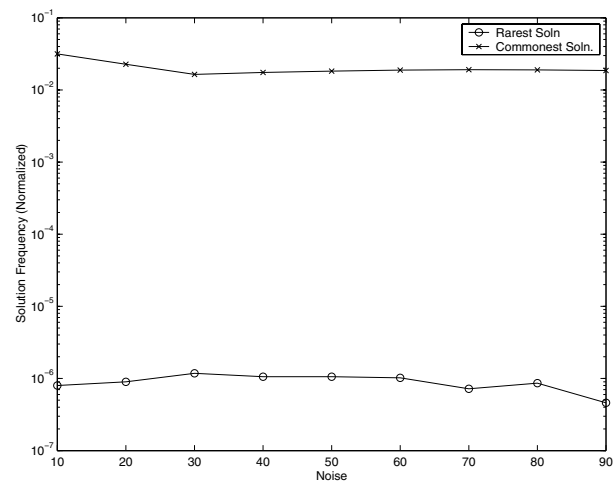
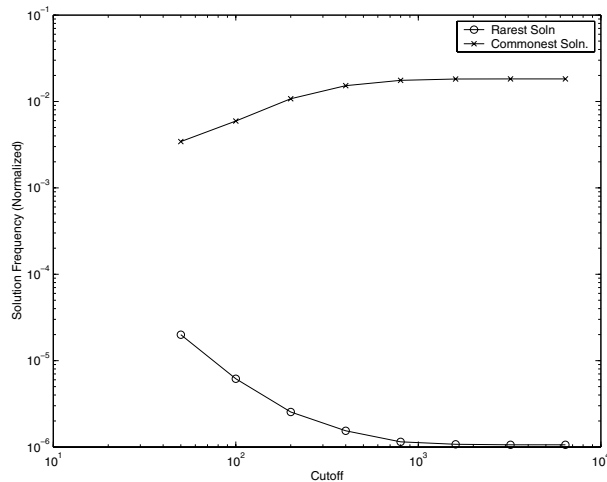


Figure 4: Uniformity of Sampling for a 3-SAT problem: (A) Soln. Frequency vs. Cutoff (B) Soln. Frequency vs. Noise

5 Summary

1. WalkSAT is able to find all solutions for the 3-SAT and constrained logistic problem, within a reasonable number of trials.
2. Solution sampling can be highly non-uniform, particularly for the 3-SAT problem.
3. The 3-SAT instances appear to have a more complicated solution space than the logistic problem.
4. Increasing WalkSAT's noise does not appear to increase the solution sampling uniformity, while it degrades WalkSAT's speed.
5. Challenge: Devise ways to obtain more uniform sampling.

References

- [Bayardo and Pehoushek, 2000] R.J. Bayardo and J.D. Pehoushek. Counting models using connected components. In *Proc. AAAI-00*.
- [McAllester *et al.*, 1997] D. McAllester, B. Selman, and H. Kautz. Evidence for invariants in local search. In *Proc. of AAAI-97*, pp 321-326, 1997.
- [Roth, 1996] D. Roth, On the Hardness of Approximate Reasoning, In *Artificial Intelligence Journal*, Vol. 82, 1-2, pp 273-302, 1996.
- [Selman *et al.*, 1994] B. Selman, H. Kautz, and B. Cohen. Noise Strategies for Local Search. In *Proc. AAAI-94*, pp 337-343, 1994.

Finite Domain Constraint Solver Learning

Arnaud Lallouet, Thi-Bich-Hanh Dao, Andreï Legtchenko, AbdelAli Ed-Dbali

Université d'Orléans – LIFO – BP 6759 – 45067 Orléans – France

firstname.name@lifo.univ-orleans.fr

Abstract

In this paper, we present an abstract framework for learning a finite domain constraint solver modeled by a set of operators enforcing a consistency. The behavior of the consistency to be learned is taken as the set of examples on which the learning process is applied. The best possible expression of this operator in a given language is then searched. We instantiate this framework to the learning of bound-consistency in the indexical language of Gnu-Prolog.

1 Introduction

Constraint Satisfaction Problems (or CSPs) have been widely recognized as a powerful tool to model, formulate and solve artificial intelligence problems as well as industrial ones. A common framework to address this task is the combination of search and filtering by a local consistency. Enforcing a local consistency is usually done by the scheduling of monotonic and contracting operators up to reach their greatest common fix point. The consistency is obtained as the common closure of the set of operators. This closure is efficiently computed by a chaotic iteration of the operators [Apt, 1999]. But usually, the task of finding efficient operators which actually define a consistency is considered as one of the smartest skills of the solver's implementor. The essence of the learning framework consists in considering the behavior of the operator enforcing the desired consistency as a set of examples in order to find an adequate representation of this operator in a given language.

2 Theoretical framework

Let V be a set of variables and $D = (D_X)_{X \in V}$ their finite domains. A *constraint* c is a pair (W, T) where $W \subseteq V$ is the arity of the constraints and $T \subseteq \prod_{X \in W} D_X$ is the set of solutions of c . A *CSP* is a set of constraints. For $W \subseteq V$, a *search state* s is a tuple $(s_X)_{X \in W}$ where $\forall X \in W, s_X \subseteq D_X$. A *singletonic* search state $(\{v_X\})_{X \in W}$ represents a single tuple. The *search space* is $S_W = \prod_{X \in W} \mathcal{P}(D_X)$. The set S_W ordered by point-wise inclusion $\subseteq$ forms a complete lattice.

A *consistency* for a constraint $c = (W, T)$ is an operator $f : S_W \rightarrow S_W$ such that:

- f is monotonic, i.e $s \subseteq s' \Rightarrow f(s) \subseteq f(s')$, in order to ensure the confluence of the reduction mechanism,
- f is contracting, i.e $\forall s \in S_W, f(s) \subseteq s$, in order to reduce variable's domains,
- f is correct w.r.t c , i.e $\forall s \in S_W$, every solutions of c which are present in s remain in $f(s)$,
- f represents c , i.e for every singletonic search state s which does not represent a solution of c , $f(s)$ is an empty state (at least one element of the tuple $f(s)$ is the empty set).

Operators which satisfy the first three conditions are called *pre-consistencies* for c . As an example of consistency, if we suppose that each variable domain D_X is ordered by a total ordering $\leq$ and for $A \subseteq D_X$, we denote by $[A]$ the set $\{a \in D_X \mid \min(A) \leq a \leq \max(A)\}$, then the bound-consistency bc_c is defined by $\forall s \in S_W, \forall X \in W, bc_c(s)_X = s_X \cap [T_X]$, with T_X the projection of T on X .

Let cs_c be the consistency to be learned. Our aim is to build a consistency f which behaves like cs_c as much as possible. Thus f must be contracting, monotonic, *correct w.r.t* cs_c ($\forall s \in S_W, cs_c(s) \subseteq f(s)$) and *singleton complete w.r.t* cs_c ($f(s) \subseteq cs_c(s)$ for any singletonic search state s). However, singleton completeness is difficult to get and even not always possible to express in a given language. In order to transform a pre-consistency into a consistency, let us define a consistency id_c such that $\forall s \in S_W, id_c(s)$ is an empty state if s is a non-solution singletonic state, and $id_c(s) = s$ otherwise. Thus $f \circ id_c$ and $id_c \circ f$ are consistencies for c if f is a pre-consistency for c . Therefore by adding id_c in the set of operators, processed by a chaotic iteration mechanism [Apt, 1999], we only need to build pre-consistencies for c . On the other hand, the correctness condition must be ensured for every $s \in S_W$ which is generally huge. We show that:

Proposition 1 *If f is a monotonic and contracting operator such that $f(s) = s$ for every singletonic state s which represents a solution of c , then f is a pre-consistency for c .*

Therefore, by considering monotonic operators, we can reduce the search space to a sample set E which is a subset of S_W and which contains all singletonic search states. Let $\mathcal{L}$ be the language in which operators are expressed and l be an operator in this language. In order to find the best possible expression, we shall be able to compare two consistencies. This

is usually done with a distance. Let d be such a distance between two consistencies. The learning problem is formulated as follows:

$$\begin{array}{ll} \text{minimize} & d(cs_s, l), \\ \text{subject to} & \forall s \in E, cs_c(s) \subseteq l(s) \subseteq s, \end{array}$$

where $E \subseteq S_W$, E contains all singleton search states of S_W and l is a monotonic operator. Following the machine learning vocabulary, cs_c represents the example space and $\mathcal{L}$ the hypothesis space.

3 Learning indexicals

To instantiate our theoretical framework, we have to define strong language biases in order to limit the combinatorial explosion.

The first question is the language in which operators are expressed. The language of indexicals [van Hentenryck *et al.*, 1991] is chosen, motivated by the ease of integration of the user-defined indexicals in Gnu-Prolog [Diaz and Codognet, 2001]. In this language, an operator is written `X in r`, where X represents the domain of the variable X and r is an expression representing a subset of D_X . If we denote x the unary constraint representing X 's domain, then the indexical represents the operator $x \mapsto x \cap r$.

Then comes the choice of consistency. We learn the bound-consistency, since it allows to limit the example space to intervals instead of arbitrary subsets.

For each variable we learn a reduction indexical and define an indexical for id_c . The reduction indexical for X is of the form `X in minX .. maxX` where `minX`, `maxX` are in some fixed forms. In practice, we use linear, piecewise linear and rational forms. In order to the reduction indexical to be monotonic, the bound `minX` must be anti-monotonic and `maxX` monotonic. This can be ensured by syntactic conditions on the sign of the coefficients for each expression.

The indexicals for id_c could be implemented in two ways: by using Gnu-Prolog indexicals for predefined constraints in which each instance of `min` and `max` is simply replaced by `val`, or by a direct code using `val` and C operators.

As distance between two consistencies, we use the global error on the example space E . By considering that f must be correct w.r.t cs_c , this distance is $\sum_{s \in E} |f(s) \setminus cs_c(s)|$.

Example For lack of space, we present here one example. An user defined global constraint is defined by the following conjunction: $X - Y > 3, X - Y < 30, 3 * X - Y > 50, 5 * Y - X < 120, X < 45, Y > 5$. When treated globally as a two dimensional polyhedra, these constraints yield less indexicals than the above decomposition. On the domain $[0..49] \times [0..49]$, our system generates the following operators:

```
X in 6*min(Y)/<20+18 .. -10/<(1 + max(Y))+45
Y in 10/<(max(X)+1)+6 .. -590/<(max(X)+1)+47
```

Reparation operators are implemented from Gnu-Prolog indexicals. Here is the one for X :

```
X in val(Y)+4 .. 44 & 0 .. val(Y)+29 &
    50+val(Y)/>3+1 .. 44 & 5*val(Y)-119 .. 44
```

When trying all reductions on boxes included in $[16, 25] \times [5, 26]$, the learned operators ran in 290 ms, while the non-decomposed constraints ran in 400 ms. All tests have been done on a Pentium4, 2Ghz, 512MB running Linux.

4 Conclusion

Related work Solver learning has been first introduced by [Apt and Monfroy, 1999] in which they automatically generate a set of rules from the tuples defining a constraint. The complexity of the rules generation limits them to small finite domains such as boolean.

The system PROPMinER [Abdennadher and Rigotti, 2002; 2003] is devoted to the learning of Constraint Handling Rules [Früwirth, 1998]. The produced solver is often very readable, especially when a small number of rules are produced. While being less general in theory since we only deal with finite domains, our method works on domains and constraint arities much larger.

In an earlier paper [Dao *et al.*, 2002], we have presented an indexical learning process. We propose here two main improvements, besides a more general theoretical framework: the possibility of using only a sample of the example space while still ensuring the correctness of the learned solver and the reparation method. It follows that our system is able to handle larger constraint arity and larger domains and therefore yields a better solver.

Summary We have presented a general, language-independent framework for finite domain constraint solver learning and an instantiation to the learning of bound-consistency with Gnu-Prolog indexicals.

Acknowledgment We gratefully thank Michel Bergère and the anonymous referees for their remarks on this paper. This work is supported by french CNRS grant 2JE095.

References

- [Abdennadher and Rigotti, 2002] Slim Abdennadher and Christophe Rigotti. Automatic generation of rule-based solvers for intensionally defined constraints. *International Journal on Artificial Intelligence Tools*, 11(2):283–302, 2002.
- [Abdennadher and Rigotti, 2003] Slim Abdennadher and Christophe Rigotti. Automatic generation of rule-based constraint solvers over finite domains. *Transaction on Computational Logic*, 2003. accepted for publication.
- [Apt and Monfroy, 1999] K. R. Apt and E. Monfroy. Automatic generation of constraint propagation algorithms for small finite domains. In Joxan Jaffar, editor, *International Conference on Principles and Practice of Constraint Programming*, volume 1713 of *LNCS*, pages 58–72. Springer, 1999.
- [Apt, 1999] K. R. Apt. The essence of constraint propagation. *Theoretical Computer Science*, 221(1-2):179–210, 1999.
- [Dao *et al.*, 2002] Thi Bich Hanh Dao, Arnaud Lallouet, Andrei Legtchenko, and Lionel Martin. Indexical-based solver learning. In Pascal van Hentenryck, editor, *International Conference on Principles and Practice of Constraint Programming*, volume 2470 of *LNCS*, pages 541–555. Springer, 2002.
- [Diaz and Codognet, 2001] Daniel Diaz and Philippe Codognet. Design and implementation of the Gnu-Prolog system. *Journal of Functional and Logic Programming*, 2001(6), 2001.
- [Früwirth, 1998] Thom Früwirth. Theory and practice of Constraint Handling Rules. *Journal of Logic Programming*, 37(1-3):95–138, 1998.
- [van Hentenryck *et al.*, 1991] P. van Hentenryck, V. Saraswat, and Y. Deville. Constraint processing in cc(fd). draft, 1991.

Applying interchangeability techniques to the distributed breakout algorithm

Adrian Petcu and Boi Faltings
{adrian.petcu, boi.faltings}@epfl.ch
Artificial Intelligence Laboratory
Swiss Federal Institute of Technology
IN (Ecublens), CH-1015 Lausanne, Switzerland

Abstract

This paper presents two methods for improving the performance of the Distributed Breakout Algorithm using the notion of interchangeability. In particular, we use neighborhood *partial* and *full* interchangeability techniques to keep conflicts localized and avoid spreading them to neighboring areas.

Our experiments on distributed sensor networks show that such techniques can significantly reduce the number of cycles required to solve the problems (therefore also reduce communication and time requirements), especially on difficult problems. Moreover, the improved algorithms are able to solve a higher proportion of the test problems.

1 Introduction

Distributed Constraint Satisfaction (DisCSP) is a powerful paradigm applicable for a wide range of coordination and problem solving tasks in distributed artificial intelligence.

Among the distributed algorithms that were developed for this kind of problems ([Yokoo *et al.*, 1998]), the Distributed Breakout Algorithm (DBA) received quite some interest (e.g. [Zhang *et al.*, 2002]) because of a number of properties that this algorithm exhibits (simple, efficient, low overhead, linear memory requirements, good anytime characteristics).

DBA is an extension of the original centralized Breakout Algorithm ([Morris, 1993]). This algorithm is a local search method, with an innovative technique for escaping from local minima: the constraints have weights, which are dynamically increased to force the agents to adjust their values while in a local minimum. During the execution of the algorithm, each agent proposes improvements to the current state by changing its variable value such that the cost of violated constraints is decreased as much as possible.

While having the interesting properties that we enumerated above, local search algorithms also have a common problem: choosing indiscriminately between the possible values of the local variable (only considering the cost of the immediate constraint violations) can lead to “chain-reactions” (one conflict originating in one part of the constraint graph needlessly propagates throughout the whole graph, only to (hopefully) be resolved in a completely different part of the graph).

We analyzed these phenomena, and drew the conclusion that using interchangeability techniques, one can determine what values from the local domain will not cause such conflict propagations, and use one of those values as the next variable assignment. In this way, we look for a “local resolution” to all conflicts, in the sense that we keep them contained as much as possible, and only involve “external parties” when there is no other way.

We discovered that techniques based on interchangeability [Freuder, 1991] (both *neighborhood partial* and *full interchangeability* [Choueiry *et al.*, 1998]) can improve the performance of this algorithm.

2 Preamble

Problem description and formalization

The distributed sensor network problem ([Gomes *et al.*, 2002]) consists of a sensor field composed of n sensors, and m targets to be tracked. Each sensor has its own visibility *range*. The sensors can communicate among themselves, but not necessarily every sensor with every other sensor.

Some restrictions apply: 3 sensors have to be assigned to each target, and they must be able to communicate among themselves; each sensor can only track one target at a time.

In our approach, one agent corresponds to a target; each agent has 3 local variables (the sensors to be assigned to each target), and the domain of each variable is the set of sensors that can track the respective target.

There are two types of constraints: *intra-agent constraints* (the variables belonging to one agent must be assigned to different sensors, and the sensors assigned to one agent must have a communication link between themselves), and *inter-agent constraints* (no 2 variables from any 2 agents can be assigned the same value - a sensor can track a single target)

Interchangeability background

The concept of interchangeability informally means equivalence between values of a CSP variable:

- *Neighborhood Interchangeability*: 2 values a and b of a variable V_i are *NI* if they are equivalent for every constraint involving V_i ;
- *Neighborhood Partial Interchangeability*: a weaker form of *NI*, defined for a subset of values from the local domain with respect to a set of neighbors, where the

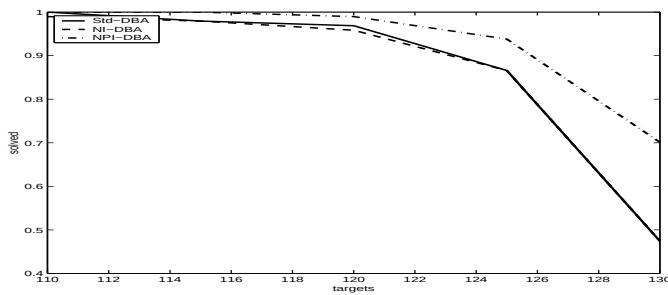


Figure 1: Percentage of solved problems

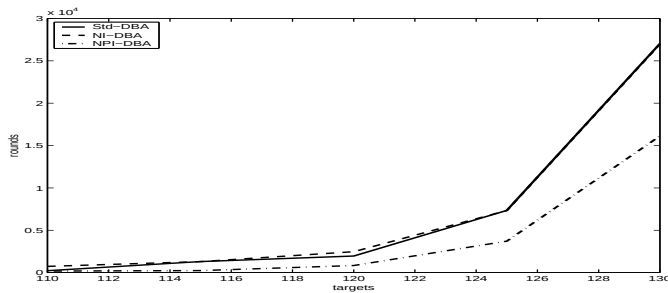


Figure 2: Average no. of rounds spent on each problem size

impact of the change of the local variable is limited to the reference set of neighbors.

3 Algorithms

Due to lack of space, we will present here only a high-level overview of the algorithms that we developed.

NI-DBA: the idea is that if we find the NI-sets for the local variables, we can safely assign values from those sets, being certain this won't cause any conflicts with the neighboring agents. The NI-sets are determined during the pre-processing phase, based on the domains of the neighbors, and are used at runtime like this: if an agent has a conflict with a neighbor, it will search for an improvement in its local domain *giving preference to the values from the NI-set*. This avoids any future conflicts with any neighbor.

NPI-DBA: the NPI-sets are computed at runtime, w.r.t. the set of the neighbors that we already have conflicts with. When searching for a local improvement, we *give preference to the values from the NPI-sets*, thus not risking to cause future conflicts with neighbors that are not already involved, therefore keeping conflicts contained.

4 Evaluation

We made our evaluations in these settings: a sensor field with 400 sensors in total, and randomly generated *solvable* problems with 40, 60, 80, 100, 110, 115, 120, 125 and 130 simultaneous targets (meaning three times as many variables).

We collected these results: problem solved/not solved (a problem is declared unsolved after the number of cycles reaches a threshold of 50000 cycles), and solving effort (time spent and number of cycles required).

For small numbers of targets, all the algorithms performed well; the differences increased with the problem difficulty, and peaked at 130 targets (most difficult problems), where NPI-DBA solved more than 70% of the problems, whereas Standard-DBA solved less than 50% (see Figure 1). Both the average number of rounds and the solving time for standard DBA are bigger than those for NPI-DBA, and close to the ones of NI-DBA. We see in Figure 2 that for difficult problems, the number of required rounds for NPI-DBA is about 40% smaller than the one of Standard DBA. A similar diagram for the time is available, but not included here.

We developed a visual interface that allows us to monitor the solving process, thus giving us clear indications that using the strategies based on NI/NPI greatly inhibits the propagation of conflicts around the constraint graph.

The initialization of the variables was random, as in Standard DBA, in order to keep the algorithms comparable, and see the improvements that the *search strategy* brings. Initialization with values from the NI-sets yields even larger improvements, leading us to believe that both the “informed” initialization of the variables and the subsequent search strategy play a role in the performance of the algorithm.

Overall, our results have shown that NPI-DBA is much better than NI-DBA. This is due to the fact that in dense problems, there is usually little or no NI at all, whereas NPI, being a weaker form of NI is still computable.

5 Conclusions and future work

The techniques presented here can be easily generalized beyond inequality constraints and resource allocation problems.

NPI-DBA clearly outperforms standard DBA for difficult problems, and NI-DBA shows comparable performance. Further speedups can be obtained with “informed” initializations, based on the NI data available after the preprocessing phase.

Further improvements could also be obtained by allowing multiple simultaneous changes of the local variables at each step, and by trying a hierarchical approach, where certain agents are delegated as “local authorities” for solving a particularly difficult local problem. It would be interesting to study in more detail the scalability of these algorithms.

References

- [Freuder, 1991] Eugene C. Freuder ‘Eliminating interchangeable values in CSPs’ In Proc. of AAAI 1991
- [Zhang et al., 2002] Weixiong Zhang and Lars Wittenburg *Distributed Breakout Revisited*. In Proc. of AAAI 2002
- [Gomes et al., 2002] Carla Gomes, Cesar Fernandez, Ramon Bejar *Communication and Computation in DisCSP Algorithms*. In Proc. of CP-2002, Ithaca, New York, USA
- [Choueiry et al., 1998] Berthe Choueiry and G. Noubir: *On the Computation of Local Interchangeability in Discrete Constraint Satisfaction Problems*. In Proc. of AAAI-98.
- [Morris, 1993] Morris, P. *The breakout method for escaping from local minima*. In Proceedings of the Eleventh National Conference on Artificial Intelligence, 40-45
- [Yokoo et al., 1998] Makoto Yokoo *The Distributed Constraint Satisfaction Problem: Formalization and Algorithms*. IEEE Trans. on Knowledge and Data Engineering

EVOC: A Music Generating System using Genetic Algorithms

Timothy Weale and Jennifer Seitzer

University of Dayton

Dayton, Ohio, USA

{timothy.weale, jennifer.seitzer}@notes.udayton.edu

Abstract

In this work, we present a Genetic Algorithm (GA) system “Evolutionary Counterpoint” (EVOC) that generates contrapuntal music. Counterpoint is the construction of a musical piece by superimposing multiple melodies, indirectly forming an underlying harmonic structure. Here, we include a description of the underlying algorithm, fitness function, and overall system modules.

1 Introduction

Counterpoint is the art of combining individual melodic voices to form a harmonious whole [Greenberg, 1997]. Contrapuntal music composition can be thought of as a vast search for the perfect combination of melodies. Given the exhaustive nature of the problem, genetic algorithms are ideal to facilitate the pursuit. In this work, we employ genetic algorithms to this end, and describe our implementation.

One aspect of counterpoint study involves taking a given musical line and creating an accompanying line. To do this correctly, the composer uses a set of guidelines (rules) for composition. These can be as simple as ‘no dissonant sounds’ to as complex as ‘perfect intervals must be approached from opposite voices and one voice must step.’ Using these sets of rules and a given melodic line known as a cantus firmus, one develops accompanying lines that adhere to every guideline. The guidelines are of utmost importance to our system because our fitness function is based on them. In our work, we have made the problem more manageable by using Species I counterpoint, the most structurally primitive form of

counterpoint.

2 Music as a Constraint Problem

Constraint programming is concerned with using constraints to solve problems. Music composition of a counterpoint nature can be seen as a specialized constraint satisfaction problem (CSP). Any CSP has the following characteristics: a set of variables, a domain for the variables, and a set of constraints on the domain [Barták, 1999]. Every note in a given cantus firmus line gives the composer at least one variable to satisfy. Every variable (note) is defined within the domain of every possible musical tone. Finally, the set of counterpoint guidelines gives us the constraints on the domain. The task of the composer (our system, EVOC) is to come up with a correct set of pitches along with its timing schedule. The version of counterpoint rules we are currently applying is limited to Species I in two voices (Figure 2), where given an upper melody as input, the system composes the bass line. We employ the counterpoint rules found in a typical music theory setting [Magnuson] some of which are shown in Figure 1.

3 Main Algorithm

begin EVOC-Algorithm()

input: cantus firmus line

output: Species I melody line

 Randomly generate 8 initial melodies (parents)

repeat

1. create 32 offspring using crossover on parents
2. randomly mutate offspring
3. apply fitness function to offspring
4. select new parents from offspring

until fitness function returns 0

Sample Principles of Species I Counterpoint

1. Leaps of a third, fourth, fifth and ascending minor sixths are available.
2. A distance of no greater than a sixth may be covered in one direction (with the exception of an octave leap).
3. Only two leaps in one direction may be used consecutively.
4. Do not repeat notes consecutively.

Figure 1

Cantus Firmus and Counterpoint Line Using Species I in Two Voices

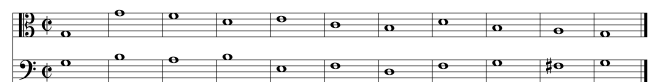


Figure 2

4 System EVOC

The EVOC system consists of two independent program modules each invoked by the Visual Basic interface. Module One reads in the input MIDI file and transforms it into an intermediate text format. Module Two uses this to run the genetic algorithm and generate a counterpoint line. A graphical representation of the logical organization can be found in Figure 3.

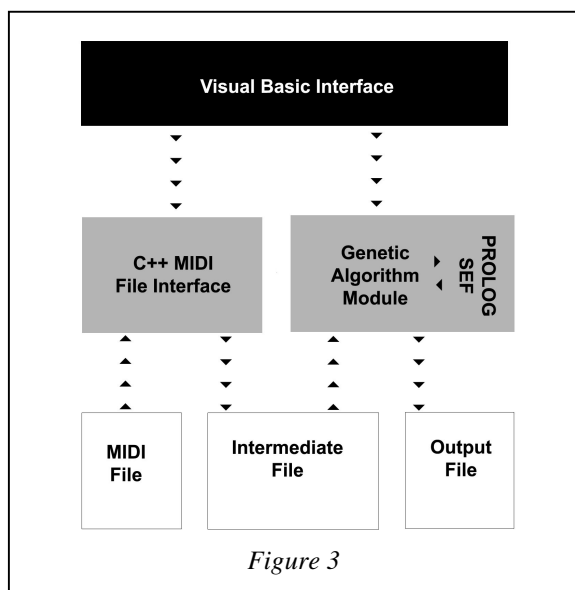


Figure 3

The Visual Basic interface provides a simple, easy way to invoke the other two modules. After loading the interface, the user defines the name of the MIDI file with the given melody line. Using this file name, the additional two modules are invoked.

The MIDI input module is a C++ program that reads in a MIDI file and outputs it as an intermediate file. The melody (as represented by a MIDI file track) is read from the file using file-processing routines implemented in Sapp's IMPROV suite [2002]. From there, the lines are converted to an internal representation, from which the intermediate file is created.

The final module runs the genetic algorithm. This part reads the intermediate file, initializes the algorithm, and employs the main algorithm detailed in Section 3. This algorithm iterates until the fitness function returns zero indicating that a correct melody has been generated. The output of this final module is the generated contrapuntal melody.

4.1 PROLOG Fitness Function

In any GA system, the fitness function indicates which members of a generation survive. For our fitness function, we wrote a theorem prover that embodies the historical Species I counterpoint guidelines. By comparing any given genotype melody with the guidelines, we assign a valuation of validity to the genotype. During se-

lection, the highest rated melodies persist to the next generation.

The evaluation function first constructs a list of *note()* values. These notes have the form: *note(MIDI_value, MIDI_name, measure)*. With these note lists, the PROLOG evaluation is done using multiple scans, with two primary areas of investigation: melodic adherence and harmonic adherence. Compositional errors occur when the constraints of the program (the species counterpoint rules) are broken. When such an error is encountered, an appropriate penalty is assessed against the particular genotype valuation. Serious infractions (e.g., parallel fifths) suffer a -500 penalty, while non-serious violations (e.g., counterbalance issues) may only suffer a -100 penalty.

5 Conclusion

To date, we have an operational system that can generate valid counterpoint lines in Species I. The output is correct and perfectly valid as a functional counterpoint line. We are currently working on improving the rule base as well as the user interface for EVOC.

In this poster, we presented a genetic algorithm and a theorem proving deductive fitness function for the searching and generation of musical counterpoint. It employs a theorem prover as a fitness function that houses fundamental guidelines used by musicians composing these pieces.

References

- [Barták, 1999] Barták, Roman. Constraint Programming: In Pursuit of the Holy Grail. Proceedings of Week of Doctoral Students (WDS99), Part IV, MatFyz Press, Prague, June 1999, pp. 555-564
- [Greenberg, 1997] Bernard S. Greenberg; Bach FAQ Site, 1997 <http://www.bachfaq.org/ctpt.html>. (date verified 11/1/2002)
- [Magnuson] Magnuson, Phillip. *Species Counterpoint*. <http://maestro.udayton.edu>.
- [Sapp, 2002] Sapp, Craig S. *Improv*. <http://improv.sapp.org>.
- [Schottstaedt, 1989] Schottstaedt, W. Automatic Counterpoint. Current Directions in Computer Music. MIT Press, Cambridge.

Temporal Reasoning with Preferences and Uncertainty

N. Yorke-Smith

IC-Parc

Imperial College London

London, U.K.

nys@icparc.ic.ac.uk*

K. B. Venable and F. Rossi

Dept. of Mathematics

University of Padova

Padova, Italy

{kvenable, frossi}@math.unipd.it†

Abstract

Temporal Constraint Satisfaction Problems allow for reasoning with events happening over time. Their expressiveness has been extended independently in two directions: to account for uncontrollable events, and, more recently, to account for soft temporal preferences. The motivation for both extensions is from real-life temporal problems; and indeed such problems may well necessitate *both* preferences and uncertainty. This paper proposes the study of temporal problems with both preferences and uncertainty, and puts forward some methods for their resolution.

1 Motivation and Background

Research on temporal reasoning, once exposed to the difficulties of real-life problems, can be found lacking both expressiveness and flexibility. Planning and scheduling for satellite observations, for instance, involves not only quantitative temporal constraints between events and qualitative temporal ordering of events, but also soft temporal preferences and contingent events over which the agent has no control. For example, on one hand, slewing and scanning activities should not overlap, but may if necessary. On the other hand, the duration of failure recovery procedures is not under the direct control of the satellite executive. To address the lack of expressiveness, preferences can be added to the framework; to address the lack of flexibility to contingency, handling of uncertainty can be added. Some real-world problems, however, have need for both. It is this requirement that motivates us.

In a temporal constraint problem, as defined in Dechter *et al.* [1991], variables denote timepoints and constraints represent the possible temporal relations between them. Such constraints are quantitative, describing restrictions on either durations or distances of events, in terms of intervals over the timeline. In general such problems are NP-complete. However, if each temporal constraint has just one interval — hence the constraints have form $l_{ij} \leq x_j - x_i \leq u_{ij}$, where x_i denote the timepoints — then we have a *Simple Temporal Problem* (STP) that can be solved in polynomial time.

*Partially supported by the EPSRC under grant GR/N64373/01.

†Partially funded by the Italian MIUR projects NAPOLI and COVER, and the ASI (Italian Space Agency) project ARISCOM.

To address the lack of expressiveness in standard STPs, the *Simple Temporal Problems with Preferences* (STPP) framework merge STPs with semiring-based soft constraints [Rossi *et al.*, 2002]. Soft temporal constraints are specified by means of a *preference function* on the constraint interval, $f : [l_{ij}, u_{ij}] \rightarrow A$, where A is a set of preference values. The set A is part of a semiring. In general, STPPs are NP-complete. However, if the preference functions are semi-convex (that is, they have at most one peak), the constraints are combined via an idempotent operation (like max or min), and the preference values are totally ordered, then finding an optimal solution of a STPP is a polynomial problem. Two solvers for STPPs with the features above are presented by Rossi *et al.* [2002]. The first, Path-solver, enforces path consistency in the constraint network, then takes the subinterval on each constraint corresponding to the best preference level. This gives a standard STP, which is then solved for the first solution by backtrack-free search. The second solver, Chop-solver, is less general but more efficient. It finds the maximum level y at which the preference functions can be ‘chopped’, i.e. reduced to the set $\{x : x \in I, f(x) \geq y\}$. This set is a simple interval for each I . Hence we obtain a standard STP, STP_y . By binary search, the solver finds the maximal y for which STP_y is consistent. The solutions of this STP are the optimal solutions of the original STPP.

To address the lack of flexibility in execution of standard STPs, Vidal and Fargier [1999] introduced *Simple Temporal Problems under Uncertainty* (STPUs). Here, again as in a STP, the activities have durations given by intervals. The timepoints, however, fall into two classes: requirement and contingent. The former, as in a STP, are decided by the agent, but the latter are decided by ‘nature’: the agent has no control over when the activity will end; he observes rather than executes. The only information known prior to observation is that nature will respect the interval on the duration. *Controllability* of a STPU is the analogue of consistency of a STP. Controllable implies the agent has a means to execute the timepoints under his control, subject to all constraints. Three notions are proposed. A STPU is *strongly controllable* if there is a fixed execution strategy that works in all realisations (that is, an observation of all contingent timepoints). Checking strong controllability is in P [Vidal and Fargier, 1999]. A STPU is *dynamically controllable* if there is a online execution strategy that depends only on observed timepoints in the

past and that can always be extended to a complete schedule whatever may happen in the future. Checking dynamic controllability is also in P [Morris *et al.*, 2001]. A STPU is *weakly controllable* if there exists at least one execution strategy for every realisation. Checking weak controllability is co-NP [Vidal and Fargier, 1999]. The three notions are ordered by their strength: strong $\Rightarrow$ dynamic $\Rightarrow$ weak.

In this paper we introduce a framework which allows one to handle both preferences and uncertainty in temporal problems. This is done by just merging the two pre-existing models of STPPs and STPUs. We also adopt the notion of controllability, to be used instead of consistency because of the presence of uncertainty, and we adapt it to handle preferences. We then investigate the complexity of checking such notions of controllability, and we develop algorithms to perform such checks. The main results are for strong and weak controllability, which we show to maintain the same complexity as in STPUs under the same hypothesis that make STPPs polynomially solvable. As future work, we plan to study dynamic controllability, and to implement our algorithms to compare their performance experimentally.

2 Simple Temporal Problems with Preferences and Uncertainty (STPPU)

A STPPU is a STPP with timepoints partitioned into two classes, requirement and contingent, just as in a STPU. Since some timepoints are not controllable by the agent, the notion of consistency of a STP(P) is replaced by controllability, just as in a STPU, with the three notions suitably extended. While all three notions of controllability used for STPUs can be adapted to STPPUs, for this short paper we will focus on just weak and strong controllability. A STPPU is *optimally strongly controllable* if there is a fixed execution strategy that works in all realisations and is optimal in each of them. Rather than optimality, one could be interested in just reaching a certain quality level: a STPPU is α -*strongly controllable*, where α is a preference level, if there is a fixed execution strategy that works in all realisations and is above α in each of them. A STPPU is *optimally* (resp. α -) *weakly controllable* if there exists at least one optimal (resp., above α) execution strategy for every realisation.

3 Checking Optimal and α Strong Controllability

It can be proved that checking either optimal or α strong controllability is in P. The method we propose relies on two known algorithms. The first is Path-solver, which enforces path consistency on a STPP. The second is Strong-Controllability(STPU) [Vidal and Fargier, 1999], which checks if a STPU is strongly controllable. The main idea is to apply Strong-Controllability(STPU) to a special STPU, which we will call P_{opt} , that can be constructed starting from the STPP P' obtained by applying Path-solver to a STPPU P . More precisely, the algorithm takes as input a STPPU P and performs the following steps: (1) forgetting the uncertainty, treat P as a STPP and apply Path-solver to it, obtaining the STPP P' ; (2) collapse the non-contingent intervals of P' to

those parts which have the highest preference level, and then neglect the preferences, obtaining a STPU P_{opt} ; (3) apply Strong-Controllability to P_{opt} . Theoretical results that we have proven, but omit for lack of space, show the correctness of the algorithm: (1) If a STPPU is optimally strong controllable, then the STPU obtained by neglecting preferences is strongly controllable; (2) A STPPU P is optimally strong controllable if and only if the STPU P_{opt} is strongly controllable.

We have also generated a different algorithm by combining Strong-Controllability with Chop-solver. This algorithm is less general but more efficient. Both algorithms we propose are polynomial, with a time complexity $O(n^3 \times R \times l)$, where n is the problem size, R is the range of the largest interval, and l is the number of preference levels.

For α -strong controllability, we rely on theoretical results similar to those above, and we propose two algorithms. Given α , the first one checks α -SC by: cutting the intervals to those parts whose preference is above α , and applying strong controllability checking to the resulting STPU. The time complexity is $O(n^3 \times R)$. The second algorithm finds the highest α at which P is α -SC, by performing a binary search for the highest α , checking strong controllability at each step. Its time complexity is $O(p \times n^3 \times R)$, where p is proportional to the precision we want in returning the result.

4 Checking Optimal and α Weak Controllability

Optimal weak controllability of a STPPU is equivalent to weak controllability of the corresponding preference-stripped STPU. Thus we can use the existing STPU algorithms for checking weak controllability. For checking α -weak controllability, we propose two approaches. The first chops the STPPU at level α by considering only those parts of its intervals which have preference level above α (semi-convexity guarantees that the result is always a single interval), and then applies Weak-Controllability to the STPU so obtained. The second approach generates the realisations that arise by considering the contingent intervals reduced to just their upper or lower bound (other realisations need not be considered) [Vidal and Fargier, 1999]), and checks the consistency of all such realisations. Both algorithms are exponential in the number of contingent constraints.

References

- [Dechter *et al.*, 1991] Rina Dechter, Itay Meiri, and Judea Pearl. Temporal constraint networks. *Artificial Intelligence*, 49:61–95, 1991.
- [Morris *et al.*, 2001] Paul Morris, Nicola Muscettola, and Thierry Vidal. Dynamic control of plans with temporal uncertainty. In *Proc. of IJCAI'01*, pages 494–502, Seattle, WA, 2001.
- [Rossi *et al.*, 2002] F. Rossi, A. Sperduti, K.B. Venable, L. Khatib, P. Morris, and R. Morris. Learning and solving soft temporal constraints: An experimental study. In *Proc. of CP'02*, pages 249–263, Ithaca, NY, 2002.
- [Vidal and Fargier, 1999] Thierry Vidal and Hélène Fargier. Handling contingency in temporal constraint networks: From consistency to controllabilities. *Journal of Experimental and Theoretical Artificial Intelligence*, 11(1):23–45, 1999.

POSTER PAPERS

KNOWLEDGE REPRESENTATION

BDIO_{CTL}

Obligations and the Specification of Agent Behavior

Jan Broersen
ICS, Utrecht University
The Netherlands
broersen@cs.uu.nl

Mehdi Dastani
ICS, Utrecht University
The Netherlands
mehdi@cs.uu.nl

Leendert van der Torre
CWI Amsterdam
The Netherlands
torre@cwi.nl

1 Introduction

Recently several agent architectures have been proposed that incorporate obligations. However, agent specification or verification languages that take obligations into account have received less attention. Our research question is how properties involving obligations can be specified or verified in an extension of Rao and Georgeff's BDI_{CTL}. In Section 2 we extend BDI_{CTL} with so-called Standard Deontic Logic, and in Section 3 and 4 we introduce various single agent and multiagent properties.

2 BDIO_{CTL}

We use an equivalent reformulation of Rao and Georgeff's formalism [1991] presented by Schild [2000], which we extend with Standard Deontic Logic (SDL, Von Wright [1951]). We only consider the semantics.

Definition 1 (Syntax BDIO_{CTL}) Assume n agents. The admissible formulae of BDIO_{CTL} are categorized into two classes, state formulae and path formulae.

- S1 Each primitive proposition is a state formula.
- S2 If α and β are state formulae, then so are $\alpha \wedge \beta$ and $\neg\alpha$.
- S3 If α is a path formula, $E\alpha$ and $A\alpha$ are state formulae.
- S4 If α is a state formula and $1 \leq i \leq n$, then $B_i(\alpha)$, $D_i(\alpha)$, $I_i(\alpha)$, $O_i(\alpha)$ are state formulae as well.
- P If α and β are state formulae, then $X\alpha$ and $\alpha U \beta$ are path formulae.

The semantics of BDIO_{CTL} involves two dimensions. The truth of a formula depends on both the world w and the temporal state s . A pair $\langle w, s \rangle$ is called a situation in which BDIO_{CTL} formulae are evaluated.

Definition 2 (Situation structure BDIO_{CTL}) Assume n agents. A structure $M = \langle \Delta, \mathcal{R}, \mathcal{B}_1, \mathcal{D}_1, \mathcal{I}_1, \mathcal{O}_1, \dots, \mathcal{O}_n, L \rangle$ forms a situation structure if Δ is a set of situations, $\mathcal{R} \subseteq \Delta \times \Delta$ is a binary relation such that $w = w'$ whenever $\langle w, s \rangle \mathcal{R} \langle w', s' \rangle$, $Z_i \subseteq \Delta \times \Delta$ for $Z \in \{\mathcal{B}, \mathcal{D}, \mathcal{I}, \mathcal{O}\}$ and $1 \leq i \leq n$ are binary relations such that $s = s'$ whenever $\langle w, s \rangle Z_i \langle w', s' \rangle$, and L an interpretation function that assigns a particular set of situations to each primitive proposition. $L(p)$ contains all those situations in which p holds.

A speciality of CTL is that some formulae – called path formulae – are not interpreted relative to a particular situation. What is relevant here are full paths. The reference to M is omitted whenever it is understood.

Definition 3 (Semantics BDIO_{CTL}) Assume n agents. A full path in situation structure M is a sequence $\chi = \delta_0, \delta_1, \delta_2, \dots$ such that for every $i \geq 0$, δ_i is an element of Δ and $\delta_i \mathcal{R} \delta_{i+1}$, and if χ is finite with δ_n its final situation, then there is no situation δ_{n+1} in Δ such that $\delta_n \mathcal{R} \delta_{n+1}$. We say that a full path starts at δ iff $\delta_0 = \delta$. If $\chi = \delta_0, \delta_1, \delta_2, \dots$ is a full path in M , then we denote δ_i by χ^i ($i \geq 0$).

Let M be a situation structure, δ a situation, and χ a full path. The semantic relation $\models$ for BDIO_{CTL} is then defined as follows:

- S1 $\delta \models p$ iff $\delta \in L(p)$ and p is a primitive proposition
- S2 $\delta \models \alpha \wedge \beta$ iff $\delta \models \alpha$ and $\delta \models \beta$
- $\delta \models \neg\alpha$ iff $\delta \not\models \alpha$ does not hold
- S3 $\delta \models E\alpha$ iff $\exists$ full path χ in M starting at δ s.t. $\chi \models \alpha$
- $\delta \models A\alpha$ iff $\forall$ full path χ in M starting at δ , $\chi \models \alpha$
- S4 $\delta \models B_i(\alpha)$ iff for every $\delta' \in \Delta$ such that $\delta \mathcal{B}_i \delta'$, $\delta' \models \alpha$
- $\delta \models D_i(\alpha)$ iff for every $\delta' \in \Delta$ such that $\delta \mathcal{D}_i \delta'$, $\delta' \models \alpha$
- $\delta \models I_i(\alpha)$ iff for every $\delta' \in \Delta$ such that $\delta \mathcal{I}_i \delta'$, $\delta' \models \alpha$
- $\delta \models O_i(\alpha)$ iff for every $\delta' \in \Delta$ such that $\delta \mathcal{O}_i \delta'$, $\delta' \models \alpha$
- P $\chi \models X\alpha$ iff $\chi^1 \models \alpha$
- $\chi \models \alpha U \beta$ iff there is at least one $i \geq 0$ such that $\chi^i \models \beta$ and for all j ($0 \leq j < i$), $\chi^j \models \alpha$

As usual 'globally α ' (on a path) is defined as $G(\alpha) = \alpha U \perp$ and 'at some future moment α holds' as $F(\alpha) = \neg G(\neg\alpha)$.

We do not discuss the correspondence between modal properties and conditions on binary relations $\mathcal{B}_i$, $\mathcal{D}_i$, $\mathcal{I}_i$, and $\mathcal{O}_i$, but discuss possible specification properties expressible as modal formulae.

Rao and Georgeff discuss realism properties

$$B_i p \rightarrow D_i p \quad D_i p \rightarrow \neg B_i \neg p$$

$$D_i E F p \rightarrow B_i E F p$$

and commitment strategies, e.g.:

$$I_i A F p \rightarrow A(I_i A F p U B_i p)$$

$$I_i A F p \rightarrow A(I_i A F p U (B_i p \vee \neg B_i E F p))$$

3 Single agent properties

The following properties characterize the relation between mental attitudes of a single agent.

3.1 Regimentation

A main question when developing a normative system is whether the norms can be violated or not, i.e. whether the norms are soft or hard constraints. In the latter case, the norms are said to be regimented. Regimented norms correspond to preventative control systems in computer security. For example, in the metro in Paris it is not possible to travel without a ticket, because there is a preventative control system, whereas it is possible to travel without a ticket on the French trains, because there is a detective control system. Norm regimentation for agent i is characterized by:

$$O_i p \rightarrow p$$

Strong and weak epistemic norm regimentation are:

$$O_i p \rightarrow B_i p \quad O_i p \rightarrow \neg B_i \neg p$$

And intentional norm regimentation is:

$$O_i p \rightarrow I_i p \quad O_i p \rightarrow \neg I_i \neg p$$

A variant of the latter regimentation is conditional to a conflict between agent i 's internal and external motivations. For example, if an agent is obliged to work but desires to go to the beach, then it intends to go to work. Or at least it does not intend to go to the beach. An agent is called strongly or weakly respectful if:

$$(O_i p \wedge D_i \neg p) \rightarrow I_i p \quad (O_i p \wedge D_i \neg p) \rightarrow \neg I_i \neg p$$

Moreover, a respectful agent can internalize its obligations in the sense that they turn into desires, or at least it cannot decide to violate the obligation. For example, if an agent is obliged to work then it also desires to work, or at least it cannot desire not to work.

$$O_i p \rightarrow D_i p \quad O_i p \rightarrow \neg D_i \neg p$$

Instead of respectful, agents may also be egocentric, which can be characterized by similar properties:

$$D_i p \rightarrow O_i p \quad D_i p \rightarrow \neg O_i \neg p$$

3.2 Persistence

Obligations typically persist until a deadline, e.g. deliver the goods before noon, or they persist forever, e.g. don't kill. We denote a deadline obligation by $O_i(p, d)$, where achievement of the proposition d is the deadline for the obligation to achieve p . A deadline obligation $O_i(p, d)$ persists until it is fulfilled or becomes obsolete because the deadline is reached:

$$O_i(p, d) =_{def} A(O_i p U(p \vee d))$$

A deadline obligation $O_i(p, p)$, for which the only deadline is the achievement of the obligation itself, is called an 'achievement obligation'. We may characterize that $O_i p$ is an achievement obligation by:

$$O_i p \rightarrow A(O_i p U p)$$

Alternatively, we may characterize that $O_i p$ persists forever, i.e. that it is a 'maintenance obligation', by:

$$O_i p \rightarrow AGO_i p$$

4 Multi-agent Obligations

In a multi-agent setting agents interact with each other, thereby creating obligations. For example, in an electronic market where agents interact to buy and sell goods, sending a confirmation to buy an item creates the obligation of payment by the buyer and the obligation of shipment of the item by the seller. Social systems may be designed in which obligations are related to the mental attitudes of other agents. For example, there may be communities in which agent i may adopt all the obligations of agent j .

$$B_i(O_j(\alpha)) \rightarrow O_i(\alpha)$$

The following property characterizes that agent i adopts the desires of agent j as its obligations. For example, if agent j desires to eat then agent i is obliged to see to it that he gets something to eat.

$$B_i(D_j(\alpha)) \rightarrow O_i(\alpha)$$

In a master slave relationship, the intentions of the master become the obligations of the slave.

$$B_i(I_j(\alpha)) \rightarrow O_i(\alpha)$$

Agent i is a dictator if for every other agent j it holds that:

$$I_i(O_j(\alpha)) \rightarrow O_j(\alpha)$$

Finally, a particular kind of dictator i is one whose desires immediately turn into obligations of another agent j .

$$D_i(O_j(\alpha)) \rightarrow O_j(\alpha)$$

5 Summary

In this paper we have introduced the BDIO_{CTL} logic; a combination of Rao and Georgeff's BDI_{CTL} formalism and standard deontic logic SDL. We have defined several specification and verification properties in this logic. The formalization of other properties is the subject of further research. The option most discussed in deontic logic is whether violations of norms can trigger new obligations, i.e. whether there is contrary-to-duty reasoning. For example, it is often assumed that the legal code does not contain contrary-to-duty norms. Properties related to contrary-to-do reasoning are therefore of particular interest.

References

- [Rao and Georgeff, 1991] A.S. Rao and M.P. Georgeff. Modeling rational agents within a BDI-architecture. In J. Allen, R. Fikes, and E. Sandewall, editors, *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR'91)*, pages 473–484. Morgan Kaufmann Publishers, 1991.
- [Schild, 2000] K. Schild. On the relationship between BDI-logics and standard logics of concurrency. *Autonomous agents and multi-agent systems*, 3:259–283, 2000.
- [Wright, 1951] G.H. von Wright. Deontic logic. *Mind*, 60:1–15, 1951.

Prolegomenon to a Theory of Conservative Belief Revision

James P. Delgrande

School of Computing Science
Simon Fraser University
Burnaby, B.C., CANADA, V5A 1S6
jim@cs.sfu.ca

Abhaya C. Nayak

Department of Computing
Macquarie University
NSW 2109, AUSTRALIA
abhaya@ics.mq.edu.au

Maurice Pagnucco*

School of CSE
The University of New South Wales
NSW, 2052, AUSTRALIA
morri@cse.unsw.edu.au

Abstract

A standard intuition underlying traditional accounts of belief change is the *principle of minimal change*. In this paper we introduce a novel account of belief change in which the agent's belief state is modified minimally to incorporate exactly the new information. Thus a revision by $p \vee q$ will result in a new belief state in which $p \vee q$ is believed, but a stronger proposition (such as $p \wedge q$) is not, regardless of the initial form of the belief state.

A reasoning entity will need to maintain its stock of beliefs in the face of new information. Such belief change is not arbitrary; rather belief change is generally taken to be guided by various *rationality criteria*. One of the most widely advocated rationality criterion is the *principle of minimal change*: that a belief state is modified minimally to incorporate new information [Makinson, 1993]. Perhaps the most evident way in which a change in belief can be said to be minimal is in terms of standard constructions such as systems of spheres [Grove, 1988] i.e., orderings of possible worlds.

In this paper we introduce an account of belief change that is orthogonal to the notion of revision in which “minimal change” is taken with respect to the new information. We examine an account of belief change in which *all* we wish to accept is the new information itself—no more, no less. This is reminiscent of the Gricean principle of *Conversational Implicature*, that in interpreting a speaker we should assume that the speaker means no more, and no less, than what she says. Our approach ensures that, in a sense to be specified, *exactly* the sentence accepted as evidence is incorporated. It proves to be the case that a modified knowledge base is a *conservative extension* (see Section 2) of the sentence for belief change; consequently we term this *conservative belief change*.

1 Motivation and Examples

The following example illustrates the traditional account of integrating new information, in accord with minimal change.

Example 1.1 (Exclusive disjunctive update) *Leslie and Robin are two students who share an apartment above your's. While they get along, they are independent and have*

*their own circles of friends. You initially believe that for the upcoming weekend neither will be in the apartment, say $K \equiv \neg l \wedge \neg r$. However, come the weekend you hear muted but unmistakable sounds of domestic activity. You modify your beliefs minimally to account for this new information, and consequently you conclude just that one of them has not gone away for the weekend, i.e. $K \diamond (l \vee r) \equiv (l \leftrightarrow \neg r)$.*¹

To be sure, this result is not dictated by the standard postulates but it seems to be the most plausible minimal change, given the information available; as well, this phenomenon recurs in the standard distance-based approaches to update (e.g., [Winslett, 1990]), as well as in the belief revision counterparts. The next example illustrates that these results aren't necessarily desirable all the time.

Example 1.2 (Inclusive disjunctive update) *There are two rooms in a warehouse, on the left and on the right. Let l and r denote the fact that the respective rooms are not empty. There are a number of boxes outside the warehouse but the rooms are initially empty, and so $K \equiv \neg l \wedge \neg r$. It subsequently begins to rain, and the boxes are moved inside. One concludes just that the rooms are not empty, i.e. $K \diamond (l \vee r) \equiv (l \vee r)$.*

This example apparently violates the principle of minimal change. As well it conflicts with the aforementioned distance-based approaches, which dictate that the result be $l \leftrightarrow \neg r$, that all the boxes are in one room or the other.

The idea here is that for a revision (or update) by a formula ϕ , *exactly* ϕ is to be incorporated into the knowledge base. Consider $K * (p \vee q)$. If the idea is that all we know about p and q is that $p \vee q$ is true, then we would want the possible combinations of truth values $\{p, q\}$, $\{\neg p, q\}$, and $\{p, \neg q\}$ to be considered possible, and so be consistent with $K * (p \vee q)$. This sense is reminiscent of Gricean conversational implicature [Grice, 1975] wherein a speaker is required to be maximally informative. Thus if a listener is told that $p \vee q$ is true, then the communicator does not know which of p, q are true; if they did, they would have conveyed the stronger information to the listener. A similar notion has been studied by Levesque, and Lakemeyer and Levesque (see [Lakemeyer and Levesque, 2000]) dealing with “only-knowing” or “only-knowing about”. These concepts arise in autoepistemic default reasoning where one may want to assert that all an agent

*Also affiliated with National ICT Australia and the ARC Centre of Excellence in Autonomous Systems.

¹ $\leftrightarrow$ is material biconditional and $\equiv$ is logical equivalence.

knows is ϕ or all that an agent knows about α is ϕ . Technically in our approach this will amount to the result of a belief change being a *conservative extension* (Section 2) of the formula to be incorporated in the knowledge base.

2 Preliminaries

We consider a finitary propositional language $\mathcal{L}$, over a set of atoms, or propositional letters, $\mathbf{P} = \{a, b, c, \dots\}$, truth-functional connectives $\neg, \wedge, \vee, \supset$, and $\leftrightarrow$, and truth-functional constants $\top$ and $\perp$. Interpretations and models are defined in the standard way; M is the set of interpretations of $\mathcal{L}$. $\text{Mod}_{\mathcal{L}}(\phi)$ denotes the set of models of sentence ϕ over language $\mathcal{L}$; the subscript $\mathcal{L}$ may be dropped if the language is clear. For $\phi \in \mathcal{L}$, we will define $\mathcal{L}(\phi)$, the language in which ϕ is expressed, as comprising the minimum set of atoms required to express ϕ (see [Parikh, 1999]). Thus $\mathcal{L}(p \wedge (q \vee \neg q)) = \mathcal{L}(p) = \{p\}$. This extends to sets of sentences in the obvious way. It follows that if $\models \phi \leftrightarrow \psi$ then $\mathcal{L}(\phi) = \mathcal{L}(\psi)$, and if $\models \phi$ then $\mathcal{L}(\phi) = \{\top\}$.

We will make use of the notion of a *conservative extension* of one set of sentences by another.

Definition 2.1 For sets of sentences $\Gamma_1 \subseteq \Gamma_2 \subseteq \mathcal{L}$ we have that Γ_2 is a conservative extension of Γ_1 iff for every $\phi \in \mathcal{L}(\Gamma_1)$, if $\Gamma_2 \models \phi$ then $\Gamma_1 \models \phi$.

Intuitively Γ_2 is a conservative extension of Γ_1 iff Γ_2 extends Γ_1 but tells us nothing more about sentences that are in the language of Γ_1 . Γ_2 may entail sentences in its extended language of course but as far as the language which it shares with Γ_1 is concerned, it says no more than Γ_1 .

A common approach in addressing belief revision has been to provide a set of *rationality postulates* for belief change functions. The *AGM approach* [Gärdenfors, 1988] provides the best-known set of such postulates. The goal is to describe belief change at the *knowledge level*, that is on an abstract level, independent of how beliefs are represented and manipulated. Belief states are modelled by sets of sentences, called *belief sets*, closed under logical consequence. K can be seen as a partial theory of the world. For belief set K and formula ϕ , $K + \phi$ is the deductive closure of $K \cup \{\phi\}$, the *expansion* of K by ϕ . Expansion is intended to be applied when new information is consistent with current beliefs. $K_{\perp}$ is the inconsistent belief set (i.e. $K_{\perp} = \mathcal{L}$).

In *belief revision*, the new information may be inconsistent with the reasoner's beliefs and needs to be incorporated in a consistent manner where possible. See [Gärdenfors, 1988] for the revision postulates. We will make reference to Grove's use of a system of spheres (SOS) model for characterizing AGM revision [Grove, 1988]. A *system of spheres centred on X* is a total, well-founded preorder on the set of interpretations, $\leq_{\text{SOS}}$, in $\mathcal{L}$ such that for $x \in M$ we have that: $x \in X$ iff $x \leq y$ for all $y \in M$. (That is, X is the least set of worlds in the preorder.) We will often omit the subscript from $\leq_{\text{SOS}}$ for readability. Revision is defined for $\text{Mod}_{\mathcal{L}}(K) = X$ by

$$\text{Mod}_{\mathcal{L}}(K * \phi) = \min_{\leq_{\text{SOS}}} \{x \in M \mid x \models \phi\} \quad (1)$$

where $\min\{\}$ denotes the minimal models under $\leq$.

Grove shows that for every belief revision operator satisfying the AGM postulates there is a system of spheres characterising that operator, and vice versa.

3 Conservative Belief Revision

We use $\hat{*}$ to denote the type of belief revision described in Section 1, called “conservative belief revision” or “C-revision.” The idea we wish to capture is that, for $K \hat{*} \phi$, ϕ is *exactly what will be believed in the resulting knowledge base*, relative to the “subject matter” or “context” implicit in ϕ . So for $K \hat{*} ((p \vee q) \wedge r)$ the idea is that $(p \vee q) \wedge r$ constrains the truth values of the atoms in $\{p, q, r\}$, and that exactly $(p \vee q) \wedge r$ will be known about these atoms in the resulting knowledge base. In particular, strengthenings of $p \vee q$, such as p or $p \leftrightarrow \neg q$ will not be true in the resulting knowledge base. This will be the case even when K implies p or $p \leftrightarrow \neg q$; hence a revision may in fact yield a weakening of the knowledge base. This restriction does not necessarily hold for the sentences not in $\mathcal{L}(\phi)$.

The semantic intuition behind our proposal is easily visualised. In Figure 1 we consider a revision where the underlying language is generated from atoms x, y and z . The agent believes $x \wedge \neg y \wedge z$ and encounters evidence $\neg x \vee \neg y$. Accordingly the models are partitioned into four cells corresponding to the interpretations over $\mathcal{L}(\neg x \vee \neg y)$. The best worlds from each of the three cells satisfying $\neg x \vee \neg y$ are chosen to represent the revised knowledge base. Clearly, the belief content of the new knowledge base modulo $\mathcal{L}(\neg x \vee \neg y)$ will be exactly $\neg x \vee \neg y$. Beliefs regarding z will depend on extralogical factors, namely the plausibility of different worlds.

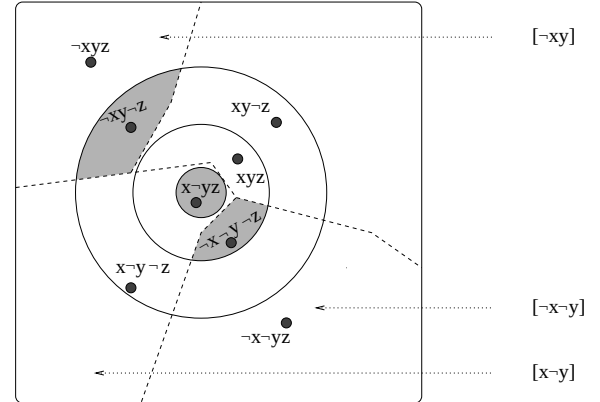


Figure 1: Conservative Revision – Semantics

Now, in determining C-revision, we consider the plausibility of different worlds represented in Figure 1 by the concentric “rings”. The worlds that are more centrally located are more plausible. Accordingly, from the $[x\neg y]$ cell, the world $x\neg yz$ is selected, whereas worlds $\neg xy\neg z$ and $\neg x\neg y\neg z$ are selected from the cells $[\neg xy]$ and $[\neg x\neg y]$ respectively. Since some of these selected worlds satisfy z and some $\neg z$, under this plausibility ordering the belief z is lost. In fact, the new beliefs regarding z can be captured by the beliefs $x \leftrightarrow z$ and $y \vee z$ that are retained from the old knowledge base.

We can formalize this analogously to Grove's system of spheres model for characterizing AGM revision. Given (1), we have the analogous definition for C-revision:

$$\text{Mod}_{\mathcal{L}}(K \hat{*} \phi) = \bigcup_{\sigma \in \text{Mod}_{\mathcal{L}(\phi)}(\phi)} \min_{\leq_{\text{SOS}}} \{\omega \in M \mid \omega \models \sigma\}. \quad (2)$$

The overall result is captured by the following theorem:

Theorem 3.1 For any belief set K and input sentence ϕ , $K \hat{*} \phi$ is a conservative extension of ϕ , i.e., for $\psi \in \mathcal{L}(\phi)$, if $K \hat{*} \phi \models \psi$ then $\phi \models \psi$.

We obtain the following elementary results.

Theorem 3.2

1. $K \hat{*} \phi \subseteq K * \phi$.
2. For ϕ a conjunction of literals, $K \hat{*} \phi = K * \phi$.

This gives rise to the question of whether a specific C-revision function can be captured using the standard definition of revision (1) in some suitably-constructed system of spheres. In general the answer is negative; for a counterexample, consider where $\mathcal{L} = \{p, q\}$ and we are given a C-revision function such that $K \equiv \neg p \wedge \neg q$ and in which $K \hat{*} p = K \hat{*} (p \wedge q)$. This entails the following constraints on the ordering:

$$\{\neg p, \neg q\} < \{p, q\}, < \{p, \neg q\}.$$

However, as is easily verified in this case, $K \hat{*} (p \vee q) \equiv p \vee q$. This however cannot be obtained by standard revision given the above constraints on the ordering, since it would require $\{p, q\}$, $\{\neg p, q\}$ and $\{p, \neg q\}$ at the same level.

While a given system of spheres determines a unique C-revision (as constructed by (2)), the converse in general does not hold. The following example demonstrates this.

Example 3.1 Consider $SOS_1: \dots < xyz < x\neg y\neg z$ and $SOS_2: \dots < x\neg y\neg z < xyz$, where the $\dots$ in the orderings represent an identical subsequence. The C-revision based on these SOS's (using Definition 2) exhibit identical behaviour since no cell of any partition based on a sub-language of $\{x, y, z\}$ will pick up exactly the set $\{xyz, x\neg y\neg z\}$.

Thus we notice an asymmetry between the classical AGM account of belief revision and C-revision. An AGM revision operation $*$, given a fixed belief set K , determines a unique system of spheres. On the other hand, the C-revision operation, given a fixed belief set K , corresponds to a class of systems of spheres. It is of interest to characterise the class of systems of spheres that a given C-revision operation $\hat{*}$ determines. However, we leave this to future work.

We consider next those postulates satisfied by C-revision.

Theorem 3.3 Let K be a belief set, $\phi, \psi \in \mathcal{L}$ and let $\hat{*}$ be defined via (2), then $\hat{*}$ satisfies the following properties:

- ($K \hat{*} 1$) $K \hat{*} \phi$ is a belief set.
- ($K \hat{*} 2$) $\phi \in K \hat{*} \phi$.
- ($K \hat{*} 3$) $K \hat{*} \phi \subseteq K + \phi$.
- ($K \hat{*} 5$) $K \hat{*} \phi = K \perp$ iff $\models \neg \phi$.
- ($K \hat{*} 6$) If $\models \phi \leftrightarrow \psi$, then $K \hat{*} \phi = K \hat{*} \psi$.
- ($K \hat{*} 7$) If $\psi \in K \hat{*} \phi$ then $K \hat{*} \phi = K \hat{*} (\phi \wedge \psi)$.
- ($K \hat{*} 9$) If $\psi \in \mathcal{L}(\phi)$ then $K \hat{*} \phi \models \psi$ implies $\phi \models \psi$.
- ($K \hat{*} 10$) If $\neg \phi \notin K \hat{*} \psi$ and $\neg \psi \notin K \hat{*} \phi$ then $K \hat{*} \phi \subseteq K \hat{*} (\phi \wedge \psi)$.

The numbering is intended to reflect correspondences with the AGM revision postulates. Postulate ($K \hat{*} 9$) is new and states that $K \hat{*} \phi$ is a conservative extension of ϕ . Since C-revision behaves the same as (standard, AGM-style) revision if the formulas involved in a revision are equivalent to sets of literals, AGM postulates 7 and 8 hold in C-revision if ϕ and ψ are equivalent to conjunctions of literals.

There are counterexamples to other AGM postulates.

Observation 1 The following do not hold of $K \hat{*} \phi$.

- ($K * 4$) If $\neg \phi \notin K$, then $K + \phi \subseteq K * \phi$.
- ($K * 7$) $K * (\phi \wedge \psi) \subseteq (K * \phi) + \psi$.
- ($K * 8$) If $\neg \psi \notin K * \phi$, then $(K * \phi) + \psi \subseteq K * (\phi \wedge \psi)$.

4 Conclusion

We have outlined a theory of conservative belief change and presented an analysis of its properties. The main intuitive motivation for this work stems from an attempt to make the most of the information presented by new evidence that a reasoner acquires. As such, our approach focuses much more on the content of the new evidence. Our current analysis suggests that the operator we introduced based on these intuitions possesses some interesting and appealing properties.

With respect to semantics, the distinction between standard AGM revision and C-revision is very much analogous to the distinction between revision and update, and in fact the two distinctions may be seen as duals of each other. For an (AGM) revision, $K * \phi$, we consider the set of all models of K , and revise by selecting the closest models of ϕ to that set. For an update, $K \diamond \phi$, we consider instead each model of K individually, and for each model of K look for the closest models of ϕ ; the update is the union of all such models. Analogous to update, for a C-revision, $K \hat{*} \phi$, we consider each model of ϕ (over $\mathcal{L}(\phi)$), and revise K by this model; the C-revision is the union of all such models. In a similar way in which we motivate C-revision from standard revision, we can define a notion of C-update from standard update. This duality between C- and standard belief change on the one hand, and between revision and update on the other, completes a classification of belief change operators, in terms of whether the models of a knowledge base or formula for change are considered en masse, or individually. It is also relatively straightforward to define syntax-dependent versions of both C-revision and C-update. Also we can look at C-contraction operations (both syntax-independent and dependent versions). However we leave this to future work.

References

- [Gärdenfors, 1988] P. Gärdenfors. *Knowledge in Flux: Modeling the Dynamics of Epistemic States*. MIT Press, 1988.
- [Grice, 1975] H. P. Grice. Logic and Conversation. In *Syntax and Semantics 3: Speech Acts*. Academic Press, 1975.
- [Grove, 1988] A. Grove. Two modellings for theory change. *Journal of Philosophical Logic*, 17:157–170, 1988.
- [Lakemeyer and Levesque, 2000] G. Lakemeyer and H.J. Levesque. *Logic of Knowledge Bases*. MIT Press, 2000.
- [Makinson, 1993] D. Makinson. Five faces of minimality. *Studia Logica*, 52:339–379, 1993.
- [Parikh, 1999] R. Parikh. Beliefs, belief revision, and splitting languages. In L.S. Moss, J. Ginzburg, and M. de Rijke, editors, *Logic, Language and Computation, Vol II*, pages 266–278. CSLI Publications, 1999.
- [Winslett, 1990] M. Winslett. *Updating Logical Databases*. Cambridge University Press, 1990.

Extending DTGOLOG with Options*

A. Ferrein, Ch. Fritz, and G. Lakemeyer

Department of Computer Science

RWTH Aachen

{ferrein, fritz, gerhard}@cs.rwth-aachen.de

1 Introduction

Recently Boutilier et al. (2000) proposed the language DTGOLOG which combines explicit agent programming with decision theory. The motivation is that a user often has some idea about how to go about solving a particular problem yet at the same time does not want to or cannot commit in advance to an exact course of action. Instead, certain choices are left to the agent running the program and determining an optimal action selection policy involves solving a Markov Decision Process (MDP) [Puterman, 1994]. In a sense, a DTGOLOG program can be thought of as a factored representation of an MDP. As an example, Boutilier et al. consider a mail-delivery scenario where the task of delivering mail to a particular person is hand-coded and fixed, while the agent chooses the order in which the various people are served according to some reward function. They note that this approach allows solving problems which are much larger than those solvable using the traditional dynamic-programming approach for MDPs.

Nevertheless, it is not very difficult to find examples where DTGOLOG also shows poor computational behavior. Roughly, the complexity is determined by the number of choices the agent needs to consider when computing a policy and the number of actions with uncertain outcomes (nature's choices) that need to be considered along the way. To better control this potential blow-up we were inspired by work on so-called macro actions or options previously developed in the MDP framework [Hauskrecht et al., 1998; Sutton et al., 1999]. The idea is that the policies for certain subtasks like leaving a room are pre-computed and then simply used when working on the global policy.

This is, roughly, what we propose: for a given domain identify subtasks and compute local policies using standard MDP techniques like value iteration. From those local policies generate two representations: (1) a (deterministic) DTGOLOG program directly encoding the local policy in terms of what action to take while the local policy is applicable and (2) so-called *abstract stochastic actions* which only address the expected outcome of a local policy. It turns out that, while both representations can be used to compute a global policy, the use of *abstract stochastic actions* may result in an exponential speed-up.— In the next section we

briefly introduce MDPs and DTGOLOG. Section 3 describes options in more detail and how to map them into DTGOLOG. We end with a discussion of experimental results.

2 Decision Theory and GOLOG

A fully observable MDP is represented as a tuple $M = \langle S, \mathcal{A}, \text{Pr}, \mathcal{R} \rangle$, where S is a finite set of states, $\mathcal{A}$ is a finite set of actions, Pr is a probability distribution $\text{Pr} : S \times \mathcal{A} \times S \rightarrow [0, 1]$ ($\text{Pr}(s_j | A, s_i)$ denotes the probability of the agent ending up in state s_j after performing action A in state s_i , and $\mathcal{R} : S \rightarrow \mathbb{R}$ is a bounded reward function. The objective is then to construct a policy $\pi : S \rightarrow \mathcal{A}$ that maximizes the expected total reward over some horizon. A simple algorithm for constructing optimal policies is value iteration (cf. [Puterman, 1994]).

DTGOLOG [Boutilier et al., 2000] can be thought of, roughly, as a programming language which allows a user to combine pre-defined primitive actions into complex programs using the usual constructs like sequence, if-then-else, while, and recursive procedures. In addition there are *nondeterministic actions* to model that an agent can choose among alternatives. The semantics is based on the situation calculus and, in particular, the so-called basic action theories described in [Reiter, 2001], which define when primitive actions are executable and how they change and do not change what is true in the world. As a special form of primitive action DTGOLOG allows so-called *stochastic actions*, which have probabilistic outcomes, just as in MDPs. For example, it is possible to define an action r (move to the right), after which a robot has moved one step to the right with probability 0.9, but has moved to the left, up, or down with probability 0.1. MDP-style reward and cost functions are incorporated as well. Given a DTGOLOG program, the idea is then to compute a policy in the sense that each nondeterministic choice is resolved in a decision theoretic manner, that is, by choosing the action with maximal expected reward. Assuming that there are not too many nondeterministic or stochastic actions, a policy can be computed even in cases where an MDP representation seems infeasible due to the size of the state space.

3 Mapping Options into DTGOLOG

As in [Sutton et al., 1999], we define *options* as triples $\langle I, \pi, \beta \rangle$, where I is the set of initial states, π is a policy, and β is the set of terminating states. To illustrate options consider the example in Figure 1 from [Hauskrecht et al., 1998].

*This work was partly supported by the German Science Foundation (DFG), Grant No. LA 747/9-1, and a grant by the NRW Ministry of Education and Research (MSWF).

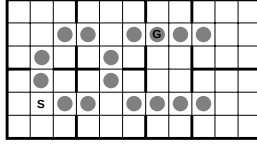


Figure 1: Maze66 from [Hauskrecht et al., 1998].

The task is to find an optimal policy to get from position S to position G . Performing an action has cost 1, the goal position has a high positive reward. The agent can perform the stochastic basic or primitive actions r, l, u, d succeeding with probability 0.9. With probability 0.1 it will be in any other adjacent position. For each room options are defined to leave the room through a certain door (one for each room/door combination). The gray dots correspond to the termination positions of these options. For the left lower room in Figure 1 there are two options O_E and O_N (leave through the east or north door, respectively) with $I_E = \{(x, y) \mid 0 < x \leq 3, 0 < y \leq 3\}$ and $\beta_E = \{(2, 4), (4, 2)\}$, and similarly for O_N . (In order to mark state (4,2) in β_E as success and (2,4) as failure, they are assigned positive and negative reward, respectively.)

Applying standard value iteration techniques, the following can be computed for an option O :

1. the optimal policy π , that is, the most appropriate action A_i for each $s_i \in I$;
2. for each $s_i \in I$ and $s_j \in \beta$, the probability to terminate the option with the outcome s_j when starting in s_i .

These results can now be translated into a form suitable for DTGOLOG. Given $I = \{s_1, \dots, s_n\}$, suppose that each s_i can be uniquely characterized by a logical formula φ_i . (In our example, φ_i simply expresses the coordinates of the location of s_i .) Then the policy π can be translated in a straightforward fashion into the following DTGOLOG program:

```
senseState(O);
while( $\varphi^I$ )
  if  $\varphi_1$  then  $A_1$  else if ... else if  $\varphi_n$  then  $A_n$ ;
  senseState(O);
endwhile
```

where $\varphi^I = \bigvee \varphi_i$ and $\text{senseState}(O)$ is a so-called sensing action. Roughly, executing $\text{senseState}(O)$ establishes the truth values of the φ_i so that they can be tested in the following while and if conditions. (The sensing actions are necessary to account for the MDP assumption of full observability.) All in all, the program simply prescribes that the optimal action (according to π) should be executed as long as the agent is in one of the initial states of O .

Given (2.) above we can also generate for each $s_i \in I$ an abstract stochastic action $o(s_i)$. For example, for the option O_E and state $s_1 = (2, 2)$ we would define an abstract stochastic action $o_e(s_1)$ with nature's choices $o_e^e(s_1)$ and $o_e^n(s_1)$ for leaving through the east or north door, respectively, together with the probabilities of these actually occurring as computed in (2.) ($\text{Prob}(o_e^e(s_1) \mid o_e(s_1)) = 0.99898$, $\text{Prob}(o_e^n(s_1) \mid o_e(s_1)) = 0.00102$).

Just as options in the MDP framework can be treated like primitive actions in MDPs that use these options, the translation into abstract stochastic actions in the DTGOLOG framework has the same effect, that is, we can now write DTGOLOG programs treating abstract stochastic actions just like any other primitive actions. A global policy computed by DTGOLOG for such a program usually mentions abstract actions. These are not readily executable (leaving a room through the

east door is not a primitive action). Hence, in a final step we need to replace the abstract actions by the programs which we derived above.

4 Experimental results

Using our running example we conducted a number of experiments, the goal being at different distances from the initial position (Fig.1 shows the special case of distance 8). The results are given in Figure 2. The x -axis depicts the initial distance to the goal, the y -axis the running time. We compared three different approaches: (A) calculating the optimal policy in DTGOLOG nondeterministically choosing only from the primitive actions, (B) using a set of procedures like the one in the previous section for leaving each room towards a certain neighboring room, choosing from primitive actions only in the goal room, and (C) using options in the form of abstract stochastic actions, choosing from primitive actions only in the goal room.

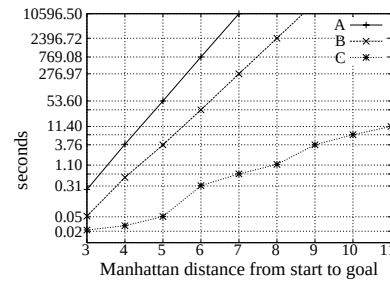


Figure 2: Runtimes of the three test programs in the maze.

Note that the y -axis of the diagram has a logarithmic scale. The speed-up from (A) to (B) shows the benefit using DTGOLOG to constrain the search space by providing fixed programs for certain subtasks. Interestingly, (C), that is, using abstract actions, clearly outperforms (B). Roughly, this is because each abstract action has only two outcomes, whereas the corresponding program provides a very fine-grained view with a huge number of outcomes that need to be considered.

Taking the time of calculation of all options into account (here: 10.51 seconds) the use of options pays off at horizons greater than 5. Also, calculating options can be done off-line and can be neglected in case of frequent reuse.

We remark that while method (A) guarantees optimality, this is not necessarily so for (B) and (C), for essentially the same reasons as in [Hauskrecht et al., 1998]. Certainly in the case of (C), this price seems worth the computational gain. Finally, while we currently assume options as given, Hauskrecht et al. (1998) discuss ways of automatically generating options with good solution qualities, an issue we intend to investigate in the future as well.

References

- [Boutilier et al., 2000] C. Boutilier, R. Reiter, M. Soutchanski, and S. Thrun. Decision-theoretic, high-level agent programming in the situation calculus. In *Proc. AAAI-2000*, 2000.
- [Hauskrecht et al., 1998] M. Hauskrecht, N. Meuleau, L. Kaelbling, T. Dean, and C. Boutilier. Hierarchical solutions of MDPs using macro-actions. In *Proc. UAI 98*, 1998.
- [Puterman, 1994] M. Puterman. *Markov Decision Processes: Discrete Dynamic Programming*. Wiley, New York, 1994.
- [Reiter, 2001] R. Reiter. *Knowledge in Action*. MIT Press, 2001.
- [Sutton et al., 1999] R. Sutton, D. Precup, and S. Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Journal of Artificial Intelligence*, 1999.

Indirect and Conditional Sensing in the Event Calculus

Jeremy Forth

Department of Computing
Imperial College, London, SW7 2BZ, UK
jforth@imperial.ac.uk

Abstract

Controlling the sensing of an environment by an agent has been accepted as necessary for its effective operation. Usually, however, agents operate in partially observable domains where not all parameters of interest are accessible for direct sensing. Sensing actions must then be chosen for what they will reveal indirectly, through an axiomatized model of the domain causal structure. This indirect form of sensing has received somewhat less attention in the literature. This article shows how sensing can be chosen so as to acquire and use indirectly obtained information to meet goals not otherwise possible. Event Calculus is extended with a knowledge formalism, and used to show how inferring unknown information about a domain leads to conditional sensing actions.

1 Integrating Knowledge with Event Calculus

To perform effectively in most practical domains, a rational agent must possess the ability to reason about, and create plans for sensing, conditional, and knowledge-producing actions. This requires the agent to not only reason about the state of objects in the domain, but also reason about the agent's own knowledge about the state of the domain [Moore, 1984]. Acquisition of knowledge may be achieved by direct sensing of the environment, or alternatively, indirectly through inference using existing or newly acquired knowledge.

Providing an agent with the capability to reason about its knowledge of the environment requires that we provide a theory that can represent and reason about that part of itself devoted to describing the environment. We augment domain fluents with meta-level knowledge fluents which at any given time point are semantically attached to the corresponding domain fluent using the knowledge axioms.

The language used to represent knowledge here [Turner, 1990] is a self-referential (amalgamated) language permitting a predicate to take a formula as an argument through the use of a naming relation to convert the formula to a unique term. For simplicity, we adopt the convention of sentences naming themselves. Using such a language, very general expressions

are possible, such as those describing incomplete information.

Shown below are axioms employed in a first-order self-referential theory of knowledge similar to modal logic's S4 system. Event Calculus as adopted here [Shanahan, 1999a] uses the distinguished predicate *HoldsAt* to represent the state of all time varying fluents, including knowledge fluents, all knowledge axioms must be defined within this predicate.

$$\text{HoldsAt}[k(f_1 \rightarrow f_2) \rightarrow [k(f_1) \rightarrow k(f_2)]] \quad (\text{K1})$$

$$\text{HoldsAt}[\forall x k(f) \rightarrow k(\forall x f)] \quad (\text{K2})$$

$$\text{HoldsAt}[k(f) \rightarrow f] \quad (\text{K3})$$

$$\text{HoldsAt}[k(f) \rightarrow k(k(f))] \quad (\text{K4})$$

$$\text{if } \text{pc} \vdash \text{HoldsAt}(f, t) \text{ then } \text{HoldsAt}(k(f), t) \quad (\text{K5})$$

$$\text{HoldsAt}[kw(f) \stackrel{\text{def}}{=} [k(f) \vee k(\neg f)]] \quad (\text{K6})$$

In order to reason about knowledge, we must define inference rules that will apply to formulae within the temporal structure:

$$\text{HoldsAt}(f_1, \rightarrow f_2, t) \rightarrow [\text{HoldsAt}(f_1, t) \rightarrow \text{HoldsAt}(f_2, t)] \quad (\text{KEC1})$$

$$\forall x \text{HoldsAt}(f_1, t) \rightarrow \text{HoldsAt}(\forall x f_1, t) \quad (\text{KEC2})$$

$$\text{HoldsAt}(\neg f, t) \equiv \neg \text{HoldsAt}(f, t) \quad (\text{NEG})$$

2 Acting in Causal Domains

Sensing actions can be axiomatized in a domain independent way by making an assumption that all fluents can be sensed directly. The effect of a pure sensing action is limited, by definition, to changing the agent's knowledge base. There will not be any effect on the domain. Clearly, in many practical situations, sensing does impact the environment, but this will be handled as a refinement to the pure sensing action defined below. A function *sense* is introduced, mapping a

fluent to an action. The sensing action is defined as follows.

$$\text{Initiates}(\text{sense}(f_p), kw(f_p), t) \quad (\text{SA1})$$

$$\begin{aligned} \text{Possible}(\text{sense}(f), t) \leftarrow \\ \text{HoldsAt}(\text{observable}(f), t) \end{aligned} \quad (\text{SP2})$$

Many actions have effects on the domain as well as on the agent's knowledge of the domain's state. An agent may come to know the status of a door by sensing, or alternatively by performing an *OpenDoor* action. Axioms KE1 and KE2 describe what the agent will come to know by performing actions.

$$\begin{aligned} \text{Initiates}(a, k(f_p), t) \leftarrow \\ [\text{Initiates}(a, f_p, t) \wedge \neg \text{Terminates}(a, f_p, t)] \end{aligned} \quad (\text{KE1})$$

$$\begin{aligned} \text{Terminates}(a, k(f_p), t) \leftarrow \\ [\text{Terminates}(a, f_p, t) \wedge \neg \text{Initiates}(a, f_p, t)] \end{aligned} \quad (\text{KE2})$$

Finally, we introduce the function *if* (from fluent and action pairs, to action) to represent conditional actions, executed just if the conditional fluent holds.

$$\begin{aligned} \text{Initiates}(if(f_1, a), f_{2p}, t) \leftarrow \\ [\text{Initiates}(a, f_{2p}, t) \wedge \text{HoldsAt}(f_1, t)] \end{aligned} \quad (\text{CA1})$$

$$\begin{aligned} \text{Terminates}(if(f_1, a), f_{2p}, t) \leftarrow \\ [\text{Terminates}(a, f_{2p}, t) \wedge \text{HoldsAt}(f_1, t)] \end{aligned} \quad (\text{CA2})$$

$$\text{Possible}(if(f, a), t) \leftarrow \text{HoldsAt}(kw(f), t) \quad (\text{CP1})$$

2.1 Example 1

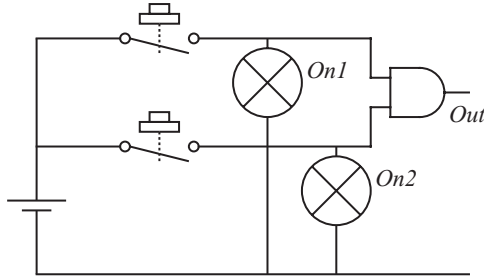


Figure 1: Unobservable output.

To illustrate the use of indirect sensing, we use the domain of Figure 1. Sensing actions are chosen to allow an agent to infer the value of *Out*. Sensing is necessary on just one of the two inputs, as long as this input is found to be inactive. If found to be active, then a second sensing action must take place. Either way, the goal of knowing whether the output is active is met.

There are two primary fluents: *On1* and *On2*, both of which can be directly observed. The output is a derived fluent, which, for the sake of example is specified to be unobservable. The goal is a knowledge goal stating that the robot knows if the output is active or not: $\text{HoldsAt}(kw(\text{Out}), 3)$. To describe the environment's ramification, we use a state constraint, and for the sake of simplicity also provide an

associated knowledge constraint:

$$\begin{aligned} \text{HoldsAt}(\text{Out}, t) \equiv \\ [\text{HoldsAt}(\text{On1}, t) \wedge \text{HoldsAt}(\text{On2}, t)] \end{aligned} \quad (\text{SC2})$$

$$\begin{aligned} \text{HoldsAt}(k(\text{Out}), t) \equiv \\ [\text{HoldsAt}(k(\text{On1}), t) \wedge \text{HoldsAt}(k(\text{On2}), t)] \end{aligned} \quad (\text{KSC2})$$

It is necessary to state that the fluents *On1* and *On2* are observable:

$$\text{HoldsAt}(\text{observable}(\text{On1}), t) \quad (\text{FD1})$$

$$\text{HoldsAt}(\text{observable}(\text{On2}), t) \quad (\text{FD2})$$

A two-action narrative is defined; a sensing action of fluent *On1* followed by a conditional sensing action of fluent *On2*, contingent upon *On1* having been found active:

$$\text{Happens}(\text{sense}(\text{On1}), 1) \quad (\text{N3})$$

$$\text{Happens}(if(\text{On1}, \text{sense}(\text{On2})), 2) \quad (\text{N4})$$

Using appropriate Uniqueness-of-names axioms, and a circumscriptive solution to the frame problem,

$$\begin{aligned} \text{CIRC}[\text{SA1}, \text{CA1-2}; \text{Initiates}, \text{Terminates}], \\ \text{CIRC}[\text{N3-4}; \text{Happens}], \text{ and} \\ \text{CIRC}[\text{CP2}, \text{SP1}, \text{TP1}; \text{Possible}]: \end{aligned}$$

we can prove the goal $\text{HoldsAt}(kw(\text{Out}), 3)$. This result demonstrates that the conditional sensing plan for knowledge acquisition in the domain of Figure 1 will always yield information about the state of the unobservable output. However, depending on the state of the first sensed fluent *On1*, the agent may not have to sense *On2* at all.

3 Conclusion

We have introduced a knowledge formalism into Event Calculus to allow an agent to represent it's knowledge of an environment sufficiently to reason about sensing, conditional actions, and inference of environmental unknowns through state constraints. It is hoped that this will form the basis for a comprehensive account of knowledge producing actions in further work.

References

- [Moore, 1984] Robert C. Moore. A formal theory of knowledge and action. In *Formal Theories of the Commonsense World*. Ablex Publishing Corp., Norwood, New Jersey, 1984.
- [Shanahan, 1999a] Murray Shanahan. The event calculus explained. *Lecture Notes in Computer Science*, 1600:409–434, 1999.
- [Turner, 1990] R. Turner. *Truth and Modality for Knowledge Representation*. Pitman Publishing: London, 1990.

Proactive Dialogue for Interactive Knowledge Capture

Jihie Kim and Yolanda Gil
Information Sciences Institute
University of Southern California
4676 Admiralty Way
Marina del Rey, CA 90292, U.S.A.
jihie@isi.edu, gil@isi.edu

Abstract

Current tools for interactive knowledge capture have little or no learning aptitude. They are mostly oblivious to the process or strategy that the user may be following in entering new knowledge, unaware of their progress during a session, and ignorant of typical skills expected from a good student. We present an approach to make acquisition interfaces more proactive by extending them with: 1) goals that represent what remains to be learned, 2) strategies to achieve these goals and acquire further knowledge, and 3) awareness of the current status of the body of knowledge learned. The resulting interaction shows that the system is aware of its progress towards acquiring the new knowledge, and moves forward by understanding what acquisition goals and strategies to pursue.

1 Introduction

The goal of our work is to develop acquisition interfaces that are proactive learners, able to reason about learning activities and with initiative in participating in the process accordingly. Our approach is to enable acquisition tools to have acquisition and learning goals represented explicitly and declaratively and be aware of the level of competence and confidence of the knowledge they are acquiring. This would enable acquisition tools to reason in terms of the learning/teaching process and to make interaction with the user dynamically generated given the situation at hand. We present our design and implementation of a new acquisition dialogue tool called *SLICK* (Skills for Learning to Interactively Capture Knowledge) that is built based on the tutoring and learning principles we have compiled so far [Kim and Gil, 2002]. *SLICK* makes acquisition tools more proactive by maintaining 1) goals that represent what remains to be learned, 2) strategies to achieve these goals and acquire further knowledge, and 3) awareness of the current status of the body of knowledge learned. The tool has been used for acquiring two very different types of knowledge: biological process models and military plans. The resulting interactions show that the system is aware of its progress towards acquiring the new knowledge, and moves forward by understanding what acquisition goals and strategies to pursue.

Teaching/Learning principle	Tutoring literature
Introduce lesson topics and goals	Atlas-Andes, Meno-Tutor Human tutorial dialog, human learning
Use topics of the lesson as a guide	BE&E, UMFE
Subsumption to existing cognitive structure	human learning, WHY, Atlas-Andes
Immediate feedback	SOPHIE, Auto-Tutor, LISP tutor Human tutorial dialog, human learning
Generate educated guesses	Human tutorial dialog QUADRATIC, PACT
Keep on track	GUIDON, SCHOLAR, TRAIN-Tutor
Indicate lack of understanding	Human tutorial dialog, WHY
Detect and fix "buggy" knowledge	SCHOLAR, Meno-Tutor, WHY Buggy, CIRCSIM, human learning
Learn deep models	PACT, Atlas-Andes
Learn domain language	Atlas-Andes, Meno-Tutor
Keep track of correct answers	Atlas-Andes
Prioritize learning tasks	WHY
Limit the nesting of the lesson to a handful	Atlas
Summarize what was learned	EXCHECK, TRAIN-Tutor, Meno-Tutor
Assess learned knowledge	WEST, Human tutorial dialog

Table 1: Some Tutoring and Learning Principles [Kim and Gil, ITS-2002]

2 Tutoring and Learning Principles for Proactive Learning

We have investigated various tutoring principles used by human tutors and instructional software that are relevant to acquisition tasks (Figure 1). They helped us understand the nature of good teacher-student interactions. We noticed that many useful learning principles could be seen as learning goals and teaching goals that students and teachers seem to pursue at different points throughout a lesson [Kim and Gil, 2002].

We also have investigated how these principles would benefit current acquisition tools [Gil and Kim, 2002]. We have found that the principles have only been used in some aspects of the functionality of acquisition tools, and are exhibited by some but not all the tools.

3 Declarative Representation of Tutoring and Learning Principles

We have used the above principles in the context of building proactive acquisition interfaces. Acquisition tools can be seen as students learning new knowledge from the user (teacher) and they should be able to use some of the strategies that good learners pursue during a tutoring dialogue. The

following presents the capabilities we provide to acquisition interfaces based on these principles.

- Acquisition interfaces should be able to represent *acquisition goals* explicitly. Many of the tutoring principles suggest a more goal-oriented behavior for acquisition interfaces. Having acquisition goals explicitly and declaratively is key to making a tool truly proactive because it could then steer the dialogue with the user to work towards those goals. The goals that are achieved at each point during the dialogue represent the progress made towards acquiring the desired body of knowledge.
- Acquisition interfaces should have *acquisition strategies* in order to understand and actively pursue what is involved in learning about a new topic. Acquisition strategies outline how to achieve acquisition goals. Because so many things are unknown to the system during the lesson, these strategies can only be pursued under the user's guidance and in a mixed-initiative interaction.
- Acquisition tools should have *awareness* of what they have learned already and what they do not know about yet, so that they can better assess their competence and confidence in specific topics, and steer the dialogue with the user in directions that improve their body of knowledge on both counts.

SLICK is developed as a front-end to existing basic acquisition tools by embodying these capabilities. For example, tools that acquire different forms of knowledge (such as problem-solving knowledge vs. concepts) may need different operational goals because they have different subcomponents and functions to build up knowledge bases. Actions done by the user through the basic acquisition tool are intercepted by our system. While the backend tool will update the backend knowledge base and its own user interface, SLICK will update its own structures and user interface.

4 An Implementation of SLICK

SLICK have been used for acquiring two very different types of knowledge: biological process models and military plans. Figure 1 shows the SLICK interface for acquiring military plans (army courses of actions). SLICK is presenting a report on a plan being entered by a military officer, pointing out how the system is understanding the plan. Users describe their plans in terms of the steps (such as attack, seize, destroy, etc.) and the objects involved (military units, terrain features, etc.). As shown in Figure 1-(a), SLICK keeps track of the lesson goal and the user's intention (e.g., expected effect), which can be used to guide the user and to check if the plan is valid. The summary window illustrates the list of involved objects and their tasks, which are essential in viewing how the plan is being built. It highlights the objects with potential problems (such as unassigned units) in red and confident subtopics are shown in blue. The user can check details of each item by clicking on it, as shown in Figure 1-(a). For example, SLICK presents confidence on knowledge items based on the number of times they were involved in test problems.

The user can view progress by checking the issues resolved over time. Towards the end of the lesson (i.e., building a

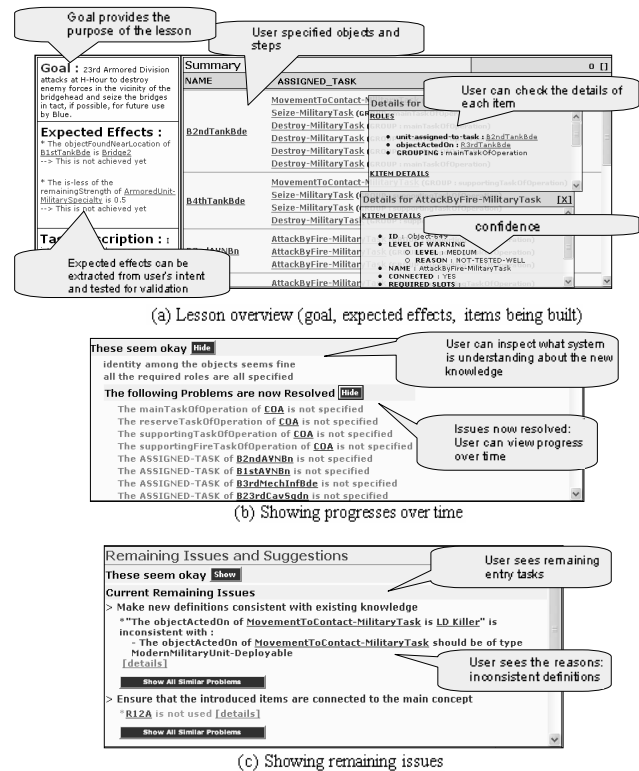


Figure 1: Acquiring military plans with SLICK.

plan), SLICK confirms that all the required roles (such as the information that existing knowledge indicates it must be provided) are specified, and identity among the objects are fine (none of the existing objects appear to be the same) (Figure 1-(b)). When SLICK notices remaining issues, it also collects the sources of the problems to help users understand the problems better. For example, Figure 1-(c) shows that there is an inconsistency between the plan and existing definitions in the KB because in the existing definitions, the 'objectActedOn' should be a military unit (ModernMilitaryUnitDeployable), but currently the user has assigned a terrain feature to it.

Note that SLICK's learning principles have derived this output. For example, in the figure, SLICK reports its understanding of the lesson and the remaining issues in terms of its goals such as "Make new definitions consistent with existing knowledge", "Ensure that the introduced items are connected to the main concept", "Ensure that the required roles are all specified", "Establish identity among the items", etc.

We plan to extend the work with dialogue planning for acquisition tasks, and incorporate a plan recognition module that relates individual user changes with multi-step plans.

References

- [Gil and Kim, 2002] Yolanda Gil and Jihie Kim *Interactive Knowledge Acquisition Tools: A Tutoring Perspective*. Proceedings of CogSci-2002.
- [Kim and Gil, 2002] Jihie Kim and Yolanda Gil *Deriving Acquisition Principles from Tutoring Principles*. Proceedings of ITS-2002.

Coherence of Laws*

Rex Kwok and Norman Y. Foo

Knowledge Systems Group
Department of Artificial Intelligence
School of Computer Science and Engineering
University of New South Wales, NSW 2052
Australia

Abhaya C. Nayak

Knowledge Systems Group
School of MPCE
Macquarie University
NSW 2109
Australia

Abstract

The core of scientific theories are *laws*. These laws often make use of theoretical terms, linguistic entities which do not directly refer to observables. There is therefore no direct way of determining which theoretical assertions are true. This suggests that multiple theories may exist which are incompatible with each other but compatible with all possible observations. Since such theories make the same empirical claims, empirical tests cannot be used to differentiate or rank such theories. One property that has been suggested for evaluating rival theories is *coherence*. This was only understood qualitatively until we [Kwok, et.al. 98] introduced a coherence measure based on the average use of formulas in support sets for observations. The idea was to identify highly coherent theories with those whose formulas that are tightly coupled to account for observations, while low coherence theories contain many disjointed and isolated statements. Our current approach generalizes that insight to accommodate fundamental ideas from the philosophy of science and better mirrors scientific practice. Moreover, this new approach is neutral with respect to the philosophy and practice of science, and is able to explain notions like modularization using coherence.

1 Introduction

This extended summary highlights the main points of the paper, a full version [Kwok, et.al. 03] of which can be obtained electronically. This section motivates the problem and subsequent sections outline the definitions which formalize the intuitions behind coherence, describe some properties that flow from these definitions, and provide examples of their use.

Scientific theories evidently comprise laws that use vocabularies that contain terms which on the one hand refer to observations, and on the other refer to postulated or theoretical entities that are not directly observable. It is in fact quite common for two theories $T1$ and $T2$ that agree on the status of their observational terms to differ in their theoretical terms. One way to compare $T1$ and $T2$ is to say that $T1$ is

more coherent than $T2$ if in accounting for the observations the formulas in $T1$ “work together better” than those in $T2$, or are “more useful” than in $T2$. A persuasive advocate for such properties is Bonjour [Bonjour 85]. In a previous paper [Kwok, et.al. 98] we proposed a definition that amounted to a *quantitative* measure of coherence of theories. In this paper we elaborate on the definition, repairing its deficiencies and extending its range of application.

2 Coherence

Definition 1 (Supports for Observations) Given an input set I and an output set O , a subset Γ of T is a I -relative support for a set O of observations if

1. Γ accounts for O , i.e., $\Gamma \cup I \models \alpha$ for each $\alpha \in O$;
2. Γ is minimal, i.e., for no $\Delta \subset \Gamma$ does Δ account for O .

Let $S(T, I, O)$ denote the set of all I -relative supports for O .

This definition differs from that in [Kwok, et.al. 98] in the relativisation of notion of support to the input set I , which better models scientific practice.

Assumption 1 (Clausal Basis Assumption) All bases of theories are clauses.

Lemma 1 (Some Properties of $S(T, I, O)$) Fix observation sets I and O , and let $T_1 \subseteq T_2$ be theories.

1. Monotonicity in T : $S(T_1, I, O) \subseteq S(T_2, I, O)$
2. $T_1 \cup I \models O$ iff $\exists \Gamma \subseteq T_1$ such that $\Gamma \in S(T_2, I, O)$.

Fix observation set I and theory T . Let $O_1 \subseteq O_2$ be observation sets. Then for every $\Gamma_2 \in S(T, I, O_2)$ there is a $\Gamma_1 \in S(T, I, O_1)$ such that $\Gamma_1 \subseteq \Gamma_2$.

Fix observation set O and theory T . Let $I_1 \subseteq I_2$ be observation sets. Then for every $\Gamma_1 \in S(T, I, O_1)$ there is a $\Gamma_2 \in S(T, I, O_2)$ such that $\Gamma_2 \subseteq \Gamma_1$.

Definition 2 (Utility of a Formula) The Utility of a formula α , if $S(T, I, O) \neq \emptyset$, in a theory T with respect to an I -relative observation set O is:

$$U(\alpha, T, I, O) = \frac{|\{\Gamma : \alpha \in \Gamma \text{ and } \Gamma \in S(T, I, O)\}|}{|S(T, I, O)|}$$

Informally, this is the *relative frequency* of occurrence of α in the support sets for O .

*Supported in part by the Australian Research Council.

Definition 3 (Coherence of a Theory) Let T be a finite theory $\{\alpha_1, \dots, \alpha_n\}$, I a finite sequence of (input) observations $(I_1, I_2, \dots, I_m)$ and O a finite sequence of (output) observations $(O_1, \dots, O_m)$. The I -relative coherence of T with respect to O is:

$$C(T, I, O) = \frac{1}{mn} \sum_{i=1}^n \sum_{j=1}^m U(\alpha_i, T, I_j, O_j)$$

Informally, coherence is the *average* utility of the elements of T in supporting some observations with the help of others. The inputs do not figure directly in the counting because it is the *internal* laws (or rules) of T that we are assessing for how the outputs are supported.

3 A Typical Application

In the full paper there are applications of the above definitions to (i) demolish Craig's Trick [Craig 53], alleged by some to show that theoretical terms are unnecessary, by demonstrating that it yields highly incoherent theories; (ii) argue that Mendel's assumption of two independent theoretical characteristics to account for the observations of his pea plant experiments yields a highly coherent theory; and (iii) indicate that programs that realize the Kolmogorov complexity of sequences are maximally coherent. Here we focus on a typical application and its implication for coherence of modular theories.

3.1 The Black Swan Fix

Prior to western ornithologists exploration of Australia all the swans they had hitherto encountered were white in color. For this focussed domain, there is only one type of object, namely swans, that are of interest. The observational predicates are *swan* and *white*, and we regard the former as the input and the latter as the output. A succinct way to capture induction is the rule 1 in the theory T below:

$$\forall x \text{ swan}(x) \rightarrow \text{white}(x). \quad (1)$$

Notice that T does not have any theoretical terms as we have specified that both the predicates are observational. In Australia they saw black swans. Here is an ad hoc way to revise T minimally if we can enumerate these black swans as additions to the original input set, i.e. these new swans are $sw_1, sw_2, \dots, sw_n$. Call this fix T_n . The revised rules that replace rule 1 are:

$$\forall x [\text{swan}(x) \wedge \bigwedge_{i \leq n} x \neq sw_i] \rightarrow \text{white}(x) \quad (2)$$

and $(2n)$ sentences

$$\bigcup_{i \leq n} \{\text{swan}(sw_i), \text{black}(sw_i)\} \quad (3)$$

Suppose the new observation terms are about swan *color*, i.e., black or white.

T_n has $2n+1$ sentences. For any finite set of k black swans, there are exactly $2k$ sentences in T_n that support their color.

Each such support sentence has utility 1 for a particular swan sw_i , and 0 for other swans.

Hence the coherence of T_n for such k observations is $\frac{2k}{2n+1}$, which is asymptotically 0 with large n . This is an argument against the fix. The "good" fix is what happens in inductive learning when a predicate is invented to summarize the discovery that black swans live in Australia, viz., the new theory T' with 2 sentences

$$\forall x [\text{swan}(x) \wedge \neg \text{Australian}(x) \rightarrow \text{white}(x)] \quad (4)$$

$$\forall x [\text{swan}(x) \wedge \text{Australian}(x) \rightarrow \text{black}(x)] \quad (5)$$

The input set I now comprises pairs of the *swan* atoms and the new observable *Australian* literals. The output O are the two color terms *white* and *black*. Now for any one swan (call it c) observation, its color is supported either by the formulas 4 and $\neg \text{Australian}(c)$, or by the formulas 5 and $\text{Australian}(c)$. Therefore, irrespective of the color the support set for each observation has cardinality 2. Suppose there are k_1 white and k_2 black swans in an observation sequence. It is then easy to see that the utilities of formula 4 is 1 for k_1 observations but 0 for the k_2 observations; formula 5 is the dual of the preceding. The coherence of this theory for any $k_1 + k_2$ swans is therefore $\frac{1}{2}$.

3.2 Modularization and Coherence

In the above example, suppose we partition the output observational terms into *black* (swans) and *white* (swans), denoting the disjoint sets by O_b and O_w respectively. Likewise, we partition the input set into two, I_b and I_w denoting the pairs of hypothesized *Australian* literal and swan atom. Then it is not hard to see that the formula 4 is in all support sets of $S(T', I_w, O_w)$, but is not in any support set of $S(T', I_b, O_b)$. Dually, the formula 5 is in all support sets of $S(T', I_b, O_b)$ but in none of those of $S(T', I_w, O_w)$. The utility of each formula is 1 in their respective support sets, and 0 in the other. This is about as strong as we can get in *modularizing* a theory.

This idea has the following obvious generalization. Suppose an observation set to be accounted for can be partitioned into $\{O_1, \dots, O_n\}$ and the theory Γ has invented theoretical terms $\{\gamma_1, \dots, \gamma_n\}$, such that for each i , γ_i is in every set of $S(\Gamma, I, O_i)$. Then the γ_i *modularize* the theory Γ with respect to the observation partitions.

References

- [Bonjour 85] L. Bonjour. *The Structure of Empirical Knowledge*. Harvard University Press, 1985.
- [Craig 53] W. Craig. On axiomatizability within a system. In *The Journal of Symbolic Logic*, 18, pages 30–32, 1953.
- [Kwok, et.al. 98] R.B.H. Kwok, A.C. Nayak, N. Foo. Coherence Measure Based on Average Use of Formulas. *Proceedings of the Fifth Pacific Rim Conference on Artificial Intelligence*, 553-564, LNCS v.1531, Springer Verlag, 1998.
- [Kwok, et.al. 03] R.B.H. Kwok, N. Foo and A.C. Nayak. Coherence of Theories (full version). Available as Postscript file via FTP <ftp://ftp.cse.unsw.edu.au/pub/users/ksg/Conference/ijcai03-Coherence.ps.gz>.

An Epistemic Logic for Arbitration (Extended Abstract)

Churn-Jung Liao

Institute of Information Science
Academia Sinica, Taipei, 115, Taiwan
liaucj@iis.sinica.edu.tw

Arbitration is the process of settling a conflict between two or more persons [Liberatore and Schaerf, 1995; Revesz, 1993; 1997]. The first version of the arbitration operator between knowledge bases is proposed in [Revesz, 1993] via the so-called model-fitting operators. The postulates for model-fitting operators and the corresponding semantic characterization are presented. Arbitration is defined as a special kind of model-fitting operators.

In [Revesz, 1997], the arbitration operator is further generalized to make it applicable to weighted knowledge bases. A set of postulates is also directly used in characterizing the arbitration between a weighted knowledge base and a regular knowledge base. A weighted knowledge base in [Revesz, 1997] is defined as a mapping $\tilde{K}$ from model sets to non-negative real numbers and a regular knowledge base is just a finite set of propositional sentences. A generalized loyal assignment is then defined as a function that assigns to each weighted knowledge base $\tilde{K}$, a pre-order $\leq_{\tilde{K}}$ between propositional sentences so that some conditions are satisfied for the pre-orders. Finally, the arbitration of a weighted knowledge base $\tilde{K}$ by a regular knowledge base K' is defined as

$$\tilde{K} \triangle K' = \min(K', \leq_{\tilde{K}}),$$

where $\min(K', \leq_{\tilde{K}})$ is the set of $\leq_{\tilde{K}}$ -minimal sentences in K' . However, this kind of arbitration is obviously syntax-dependent. For example, if φ_1 and φ_2 are two propositional sentences such that $\varphi_1 <_{\tilde{K}} \varphi_2$, then $\tilde{K} \triangle \{\varphi_1, \varphi_2\} = \{\varphi_1\} \neq \tilde{K} \triangle \{\varphi_1 \wedge \varphi_2\} = \{\varphi_1 \wedge \varphi_2\}$ even though the two knowledge bases $\{\varphi_1, \varphi_2\}$ and $\{\varphi_1 \wedge \varphi_2\}$ are semantically equivalent.

An alternative, seemingly more natural characterization for arbitration is given in [Liberatore and Schaerf, 1995] without resorting to model-fitting operators. A knowledge base in that work is identified with a set of propositional models. Thus the semantic characterization for this kind of arbitration is given by assigning to each subset of models A a binary relation $\leq_A$ over the set of model sets that satisfies the following conditions (the subscript is omitted when it means all binary relations of the form $\leq_A$):

1. transitivity: if $A \leq B$ and $B \leq C$ then $A \leq C$,
2. if $A \subseteq B$ then $B \leq A$,
3. $A \leq A \cup B$ or $B \leq A \cup B$,
4. $B \leq_A C$ for every C iff $A \cap C \neq \emptyset$,

$$5. A \leq_{C \cup D} B \Leftrightarrow \begin{cases} C \leq_{A \cup B} D \text{ and } A \leq_C B \text{ or} \\ D \leq_{A \cup B} C \text{ and } A \leq_D B. \end{cases}$$

Then the arbitration between two sets of models A and B is defined as

$$A \triangle B = \min(A, \leq_B) \cup \min(B, \leq_A) \quad (1)$$

Note that although the relationship $\leq_A$ is defined between sets of models, in the definition for arbitration, only $\leq_A$ between singletons is used. Thus by slightly abusing the notation, $\leq_A$ may also denote an ordering between models.

While arbitration is originally proposed for the merging of knowledge bases, it is also applicable in multi-agent systems. In particular, we can apply it to the reasoning about belief fusion of multiple agents. Epistemic logic is an important framework for reasoning about knowledge or belief of agents [Fagin *et al.*, 1996], so the incorporation of arbitration operators into epistemic logic will facilitate its application to belief fusion.

To incorporate the arbitration operator of [Liberatore and Schaerf, 1995] into epistemic logic, we must first note that according to (1), the arbitration is commutative but not necessarily associative. Therefore, the arbitration operator should be a binary one between two agents. We can directly add a class of modal operators for arbitration into epistemic logic. However, to be more expressive, we will also consider the interaction between arbitration and other epistemic operators, so we define the set of *arbitration expressions* over the agents recursively as the smallest set containing $\{1, 2, \dots, n\}$ and closed under the binary operators $+$, $\cdot$, and $\triangle$. Here $+$ and $\cdot$ correspond respectively to the distributed belief and the so-called “everybody knows” operators in multi-agent epistemic logic [Fagin *et al.*, 1996].

Let $\mathcal{L}$ denote the language of our logic. The alphabet of $\mathcal{L}$ consists of a countable set $\Phi_0 = \{p, q, r, \dots\}$ of atomic propositions, the propositional constants $\perp$ (falsum or falsity constant) and $\top$ (verum or truth constant), the binary Boolean operator $\vee$ (or), the unary Boolean operator $\neg$ (not), a set $Ag = \{1, 2, \dots, n\}$ of agents, the set of arbitration expressions over Ag , and the auxiliary symbols “[”, “]”, “(”, “)”, and “,”.

The set of well-formed formulas (wffs) of $\mathcal{L}$ is the smallest set containing $\Phi_0 \cup \{\perp, \top\}$ and closed under Boolean operators and the following rule:

- if φ is a wff and a is an arbitration expression, then $[a]\varphi$ is a wff.

As usual, other classical Boolean connectives $\wedge$ (and), $\supset$ (implication), and $\equiv$ (equivalence) can be defined as abbreviations. Also, we will write $\langle a \rangle \varphi$ as the abbreviation for $\neg[a]\neg\varphi$. Note that it has been shown that the only associative arbitration satisfying postulates 7 and 8 of [Liberatore and Schaerf, 1995] is $A\Delta B = A \cup B$, so if Δ is an associative arbitration satisfying those postulates, then $[a\Delta b]\varphi$ is reduced to $[a \cdot b]\varphi$, which is, in turn, equivalent to $[a]\varphi \wedge [b]\varphi$.

For the semantics, a possible model is a tuple

$$(W, (\mathcal{R}_i)_{1 \leq i \leq n}, V, \leq),$$

where

- W is a set of possible worlds,
- each $\mathcal{R}_i$ is a serial binary relation over W^1 ,
- $V : \Phi_0 \times W \rightarrow \{0, 1\}$ is a truth assignment, and
- $\leq$ is a function assigning to each subset of possible worlds A a binary relation $\leq_A \subseteq 2^W \times 2^W$ satisfying the five conditions for the definition of arbitration operator.

Note that the first two conditions imply that $\leq_A$ is a pre-order over 2^W . For each arbitration expression, we can define the binary relations $\mathcal{R}_{a\Delta b}$, $\mathcal{R}_{a \cdot b}$ and $\mathcal{R}_{a+b}$ over W recursively by

$$\mathcal{R}_{a\Delta b}(w) = \min(\mathcal{R}_a(w), \leq_{\mathcal{R}_b(w)}) \cup \min(\mathcal{R}_b(w), \leq_{\mathcal{R}_a(w)}) \quad (2)$$

$$\mathcal{R}_{a+b} = \mathcal{R}_a \cap \mathcal{R}_b \quad (3)$$

$$\mathcal{R}_{a \cdot b} = \mathcal{R}_a \cup \mathcal{R}_b \quad (4)$$

Thus the satisfaction for the wff $[a]\varphi$ is defined as

$$u \models [a]\varphi \text{ iff for all } w \in \mathcal{R}_a(u), w \models \varphi.$$

When the arbitration expression a is a single agent index, the formula $[a]\varphi$ corresponds to $K_a\varphi$ in epistemic logic, which means that the agent a knows or believes φ . We can also abbreviate $[i_1 + (i_2 + \dots (i_{k-1} + i_k))]$ as $[G]$ where $G = \{i_1, i_2, \dots, i_k\}$. $[G]\varphi$ corresponds to the distributed knowledge formula $K_G\varphi$ of [Fagin et al., 1996], which means that the combined knowledge of agents in G implies φ .

Since the set of possible worlds W may be infinite in our logic, the minimal models in (2) may not exist, so we define *coherent models* as those satisfying the limit assumption [Arlo-Costa and Shapiro, 1992] for each binary relations $\leq_A$ such that $A \subseteq W$:

for any nonempty $U \subseteq W$, $\min(U, \leq_A)$ is nonempty.

An axiomatic system for coherent models is presented in figure 1, where a, b , and c are meta-variables for arbitration expressions, i is for agent, φ and ψ are for wffs, and G_1 and G_2 are for groups of agents. Axioms (f)-(l) correspond to the postulates 2-8 of [Liberatore and Schaerf, 1995] and the limit assumption is needed for the soundness of axioms (i) and (l). Though the completeness of such a system is not yet proved, this brief presentation has shown that the modal logic approach can provide a framework for integrating epistemic reasoning and different knowledge merging operators at the object logic level.

¹A binary relation $\mathcal{R}$ over W is called serial if it satisfies: $\forall w \exists u. \mathcal{R}(w, u)$.

1. Axioms:

- (a) P: all tautologies of the propositional calculus
- (b) $([a]\varphi \wedge [a](\varphi \supset \psi)) \supset [a]\psi$
- (c) $\neg[\{i\}]\perp$
- (d) $[G_1]\varphi \supset [G_2]\varphi$ if $G_1 \subset G_2$
- (e) $[a \cdot b]\varphi \equiv ([a]\varphi \wedge [b]\varphi)$
- (f) $[a\Delta b]\varphi \equiv [b\Delta a]\varphi$
- (g) $[a\Delta b]\varphi \supset [a + b]\varphi$
- (h) $\neg[a + b]\perp \supset ([a + b]\varphi \supset [a\Delta b]\varphi)$
- (i) $[a\Delta b]\perp \supset [a]\perp \wedge [b]\perp$
- (j) $([a\Delta(b \cdot c)]\varphi \equiv [a\Delta b]\varphi) \vee ([a\Delta(b \cdot c)]\varphi \equiv [a\Delta c]\varphi) \vee ([a\Delta(b \cdot c)]\varphi \equiv [(a\Delta b) \cdot (a\Delta c)]\varphi)$
- (k) $[a]\varphi \wedge [b]\varphi \supset [a\Delta b]\varphi$
- (l) $\neg[a]\perp \supset \neg[a + (a\Delta b)]\perp$

2. Rules of Inference:

- (a) Modus ponens(MP):

$$\frac{\varphi \quad \varphi \supset \psi}{\psi}$$

- (b) Necessitation(Nec):

$$\frac{\varphi}{[a]\varphi}$$

Figure 1: An axiomatic system for the logic of arbitration

Acknowledgments

We would like to thank two anonymous referees for their helpful comments.

References

- [Arlo-Costa and Shapiro, 1992] H.L. Arlo-Costa and S.J. Shapiro. "Maps between nonmonotonic and conditional logic". In B. Nebel, C. Rich, and W. Swartout, editors, *Proceedings of the 3rd International Conference on Principle of Knowledge Representation and Reasoning*, pages 553–564. Morgan Kaufmann, 1992.
- [Fagin et al., 1996] R. Fagin, J.Y. Halpern, Y. Moses, and M.Y. Vardi. *Reasoning about Knowledge*. MIT Press, 1996.
- [Liberatore and Schaerf, 1995] P. Liberatore and M. Schaerf. "Arbitration: A commutative operator for belief revision". In *Proceedings of the Second World Conference on the Fundamentals of Artificial Intelligence (WOCFAI '95)*, pages 217–228, 1995.
- [Revesz, 1993] P. Z. Revesz. "On the semantics of theory change: Arbitration between old and new information". In *Proceedings of the Twelfth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 71–82, 1993.
- [Revesz, 1997] P.Z. Revesz. "On the semantics of arbitration". *International Journal of Algebra and Computation*, 7(2):133–160, 1997.

Constructing utility models from observed negotiation actions

Angelo Restificar

University of Wisconsin-Milwaukee
Milwaukee, WI 53211 USA
angelo@cs.uwm.edu

Peter Haddawy

Asian Institute of Technology
Pathumthani, Thailand 12120
haddawy@ait.ac.th

Abstract

We propose a novel method for constructing utility models by learning from observed negotiation actions. In particular, we show how offers and counter-offers in negotiation can be transformed into gamble questions providing the basis for eliciting utility functions. Results of experiments and evaluation are briefly described.

1 Introduction

Utility functions encode an agent's preference and risk-taking behavior, and in many negotiation scenarios they are often considered private information. Although various elicitation techniques for decision makers have been widely used (see for example [Keeney and Raiffa, 1976]), they are not readily applicable in the negotiation scenario. For example, an agent can not ask the the opposing party lottery questions to assess its utility function. The use of learning mechanisms in negotiation has been investigated in several recent studies, see for example [Bui *et al.*, 1999; Zeng and Sycara, 1998], and has been shown to be an effective tool in handling uncertainty and incompleteness. None of the previous studies, however, have directly addressed the issue of utility elicitation. In many cases, value functions and utilities are assumed to be partially or completely known as shown in their experiments and evaluation. In this work, we assume that each party's utility is completely unknown to the other and no trusted third party exists. We describe how information provided by observed negotiation actions can be used to construct the utility model of an agent.

An important tool in eliciting utility functions is the use of lottery or gamble questions and the concept of certainty equivalence. Let D be a domain, U be a utility function over D , and let o_i and o_j be outcomes in a gamble G where o_i occurs with a probability p , o_j occurs with a probability $(1 - p)$, and $o_i, o_j \in D$. A certainty equivalent is an amount $\hat{x}$ such that the decision maker (DM) is indifferent between G and $\hat{x}$. Thus, $U(\hat{x}) = pU(o_i) + (1 - p)U(o_j)$ or $\hat{x} = U^{-1}[pU(o_i) + (1 - p)U(o_j)]$.

2 Negotiation actions as gamble questions

We would like to construct a model that gives us preference and risk-taking information given a position x_i and a gamble that insists on a position x_j . A position is either an offer or a counter-offer. Due to space limitation, our discussion is restricted to constructing the buyer's model. We use Zeuthen's concept of probability to risk a conflict [Zeuthen, 1930] as a basis for transforming negotiation transactions into gamble questions. While Zeuthen assumes complete knowledge

of utility functions, our approach attempts to build a utility model by observing negotiation actions.

A conflict occurs when both parties do not reach an agreement. The probability that a party would risk a conflict is proportional to the difference between what the party wants and what it is offered. The closer the other party's offer is to what is desired, the smaller this probability should be. The farther away the other party's offer is from what is desired, the larger the readiness to risk a fight or conflict.

Definition 1 Let U_B and U_S be B 's (buyer) and S 's (seller) utility function, respectively. Let x_B be B 's position and x_S be S 's position. The probability that B will risk a conflict, p_B , and the probability that S will risk a conflict, p_S , are defined as follows:

$$p_B = \begin{cases} \frac{U_B(x_B) - U_B(x_S)}{U_B(x_B)} & \text{if } U_B(x_B) > U_B(x_S) \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

$$p_S = \begin{cases} \frac{U_S(x_S) - U_S(x_B)}{U_S(x_S)} & \text{if } U_S(x_S) > U_S(x_B) \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

By computing the expected utility, B can decide whether to accept S 's offer or insist its own position. In particular, B counter-offers x_B whenever the gamble has a utility higher than x_S , i.e., $U_B(x_S) < (1 - p_S)U_B(x_B) + p_S U_B(\text{conflict})$. B accepts x_S if for B the expected utility of the gamble $(1 - p_S)U_B(x_B) + p_S U_B(\text{conflict})$ can not exceed $U_B(x_S)$. If the utility of a non-agreement (conflict) is higher than x_S then the offer is rejected.

Without losing generality we will assume that U_B is monotonically decreasing while U_S is monotonically increasing. The following theoretical results allow us to generate training instances implied by a counter-offer.

Theorem 1 Let x_B be B 's counter-offer and x_S be S 's offer. Furthermore, let $G = (1 - p_S)U_B(x_B) + p_S U_B(\text{conflict})$ and let $\hat{x}_B$ be B 's certainty equivalent for G . If $(G > U_B(x_S))$ then for any nondegenerate G , $\hat{x}_B \in (x_B, x_S)$.

Theorem 2 (Inferior Offers) Let x_B be B 's counter-offer and x_S be S 's offer. Furthermore, let $G = (1 - p_S)U_B(x_B) + p_S U_B(\text{conflict})$ and p_S^* be S 's readiness to risk a conflict at x , $x_S < x$. If $(G > U_B(x_S))$ then $((1 - p_S^*)U_B(x_B) + p_S^* U_B(\text{conflict})) > U_B(x)$, $\forall x$ such that $x_S < x$.

By Theorem 1, if B decides to make a counter-offer x_B to an offer x_S by S , then B 's certainty equivalent belongs to the interval (x_B, x_S) . Theorem 2 states that if B prefers the gamble to an offer of x_S by S then B would also prefer to gamble than accept any offer by S higher than x_S . Symmetrically, B prefers any offer lower than x_B over a gamble. Counter-offers imply preference between a gamble and a specific offer. Although B 's estimate of S 's probability to risk a

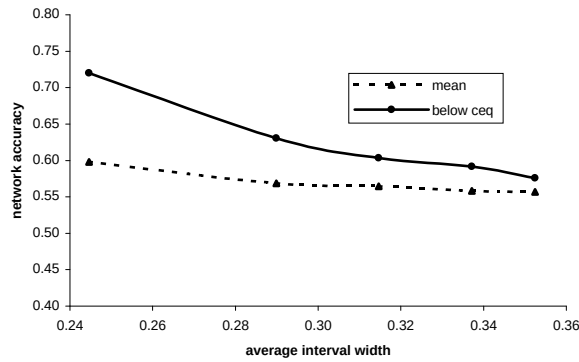


Figure 1: Overall Performance Results

conflict, p_S , is not specifically known, B 's counter-offers are in part based on p_S . Hence, by learning B 's actions, p_S values are indirectly incorporated into the model.

3 Utility model construction and evaluation

We ran experiments using two control utility functions: $\gamma_1(x) = 1 - \frac{e^{0.05x}}{149}$ (risk-averse, decreasing function) and $\gamma_2(x) = e^{-0.0025x}$ (risk-prone, decreasing function). In each negotiation sequence, p_S is either generated randomly or is set using $\{0.50, 0.25, 0.60\}$ as values. The negotiation using $\gamma_1(x)$ is over the domain $D_1 = [50, 100]$ and that of $\gamma_2(x)$ is over $D_2 = [200, 700]$. The value of conflict c is set at the maximum value of the domain. The buyer-seller negotiation strategy, σ , vary among Boulware-Conceder, Conceder-Boulware, and Conceder-Conceder pairs. We define our Boulware strategy as one where the agent concedes only 10% of the time and a Conceder strategy as one where concession is frequent at 90% of the time. Whenever an agent concedes, concession is randomly chosen between 0–50% of the difference between both agents' most recent counter-offers.

The artificial neural network used in our experiments has one hidden layer with four nodes. The input layer contains three input nodes and the output layer contains two nodes. Data fed into the input layer are scaled so that values only range between 0 and 1. Negotiation sequences used for training, tuning, and testing are randomly generated using a chosen strategy pair, a control utility function $\gamma(x)$, a negotiation domain, and a constant conflict value. We used a $k-1$ cross validation method to train and tune the network, where k is the number of negotiation transactions (offer-counter pairs) in each negotiation sequence. Network training is stopped when either no improvement in performance is detected for a successive 2,000 epochs or the number of epochs reaches 20,000. Among the data generated using the intervals, 90% is used for training and 10% is used for tuning. We point out that although the generated data maybe learned using other techniques we have chosen to use neural networks for convenience purposes.

We have randomly generated a total of 97 negotiation sequences, wherein each of B 's counter-offer B 's expected utility of the gamble is greater than that of S 's offer. The total number of negotiation transactions is 477 which gives an average of 5 transactions per negotiation sequence. The training instances are obtained by generating a total of 200 random data points for each observed negotiation transaction; 100 random data points for each of the region below and above the interval. The certainty equivalent, which is obtained from

the control utility function, lies inside each interval. For each of the region below and above the certainty equivalent 100 test points are generated. We then evaluate the approach by comparing how well the model performs when trained using the intervals against the test points from the control utility function.

Intuitively, not all negotiation transactions maybe useful. For example, an offer that is near the maximum domain value and a counter-offer that is near the minimum domain value has an interval width that is close to the width of the domain. Since we are using the interval to estimate the certainty equivalent such a negotiation transaction would be less useful to one in which both the offer and counter-offer are closer to the certainty equivalent. Results of regression analysis suggest that useful interval widths are those that are about 50% of the domain width. We have used this result as a basis to eliminate data points that may not be useful in constructing the utility model.

To test the overall performance, negotiation sequences were grouped into subsets where normalized interval widths are no greater than 0.50, 0.45, 0.40, 0.35, and 0.30. The average interval width for each subset is 0.35, 0.34, 0.31, 0.29, and 0.24, respectively. The average network performance of each of these respective subsets is shown in Figure 1. The overall network performance increases as the average interval width corresponding to the negotiation transactions decreases. The solid curve shows the performance of the utility model in predicting whether an offer is preferred by B to a gamble using only implicit data points below the interval. This is important because B 's counter-offers only correspond to the lower limit of the interval. The dotted curve shows the accuracy of the model in predicting whether an offer is preferred by B to a gamble and whether B prefers a gamble to an offer. The mean accuracy is obtained by averaging the results using implicit data points below and above the interval. The results suggest that for intervals with average width of 0.24 the network can predict about 72% of the time whether an offer is preferred to a gamble by B . For intervals with average width of less than or equal to 0.31, we are able to predict with more than 60% accuracy whether B prefers the offer to a gamble. In addition, the predictive accuracy of the model when implicit data points above and below the interval are used is better than a random guess. We ran four right-tailed z-tests and one right-tailed t-test using the following hypotheses: $H_0 : \mu = 0.50$ and $H_a : \mu > 0.50$. For the t-test the null hypothesis is rejected at $\alpha = 0.005$. In each of the z-tests, the null hypothesis is rejected at $\alpha = 0.001$.

In summary, we have outlined theoretical results that would allow us to construct utility models from negotiation actions. Results from our experiments suggest that our utility model provides significant predictive capability.

References

- [Bui et al., 1999] H. H. Bui, S. Venkatesh, and D. Kieronska. Learning other agents' preferences in multi-agent negotiation using the bayesian classifier. *International Journal of Cooperative Information Systems*, 8:275–294, 1999.
- [Keeney and Raiffa, 1976] Ralph Keeney and Howard Raiffa. *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. Cambridge University Press, 1976.
- [Zeng and Sycara, 1998] Dajun Zeng and Katia Sycara. Bayesian learning in negotiation. *Int. Journal Human-Computer Studies*, 48:125–141, 1998.
- [Zeuthen, 1930] Frederick Zeuthen. *Problems of Monopoly and Economic Warfare*. Routledge and Kegan Paul, Ltd., 1930.

Engineering a complex ontology with time

Jorge Santos

GECAD - Knowledge Engineering and
Decision Support Research Group
Instituto Superior de Engenharia do Porto
Departamento de Engenharia Informática
4200-072 Porto - Portugal
<http://www.dei.isep.ipp.pt/~jsantos>

Steffen Staab

Universität Karlsruhe (TH), Institut AIFB
D-76128 Karlsruhe, Germany
<http://www.aifb.uni-karlsruhe.de/WBS/sst>
&
Ontoprise GmbH, 76227 Karlsruhe, Germany
<http://www.ontoprise.com>

1 Principal Idea

Although most ontologies available on the Web (e.g., consider the DAML ontology library) exhibit only rather simple structures, viz. taxonomies and frame-like links between concepts. There are some domains that frequently needs intricate concept descriptions and interactions — in particular ones about time and space.

Despite the last developments in practical theories about time and in the engineering of concept hierarchies and concept frames, the issue of how to engineer complex ontologies with intricate interactions based on time has not been researched very deeply, yet, rendering the engineering of a new complex domain ontology with time a labor intensive, one-off experience with little methodology.

Here, we summarize our ontology engineering methodology, FONTE (Factorizing ONTOlogy Engineering complexity), which pursues a ‘divide-and-conquer’ strategy for engineering complex ontologies with time. FONTE divides a targeted ontology that is complex and that includes time into two building blocks, viz. a temporal theory and a time-less domain ontology. Each one of the two ontologies can be built independently allowing for a factorization of complexity. The targeted ontology is then assembled from the time-less domain ontology and the temporal theory by an operator $\otimes$.

Thereby, the assembling operator $\otimes$ is very different from existing operators for merging or aligning ontologies [Noy and Musen, 2000; Rahm and Bernstein, 2001]. Merging ontologies is a process that intends to join different ontologies about overlapping domains into a new one and most of its problems and techniques are related to the identification of similar concepts through structure analysis (e.g. graph analysis, path length, common nodes or/and edges and lexical analysis). For instance, car from ontology 1, *O1.car*, and auto from ontology 2 *O2.auto* may be defined to be identical in the merging process because of results of the structure analysis. To formalize the merging and aligning process, Wiederhold proposed a general algebra for composing large applications through merging ontologies of related domains [Wiederhold, 1994] and actually, the operations proposed (*Intersection*, *Union* and *Difference*) are about the similarities and differences of two ontologies.

In contrast, the result of $\otimes$ needs rather to be seen in analogy to the Cartesian product of two entities. For instance, car from ontology 1, *O1.car*, with its frame *O1.licensedInState* is

assembled by $\otimes$ with ontology 2 and its *O2.timeInterval* in a way such that every car in the result ontology has a lifetime as well as multiple *O1.licensedInState*-frames with different, mutually exclusive life spans.

$\otimes$ is operationalized by an iterative, interactive process. It starts off with a human assembly — in the sense just explained — between an ontology *O1*, the time-less domain ontology, and an ontology *O2*, the temporal theory. It is then propelled by a set of rules and a set of constraints. The set of rules drives a semi-automatic process proposing combinations. The set of constraints narrows down the set of plausible proposals to valid ones.

In the past a variety of approaches were proposed for reducing the complexity of engineering a rule-based system, e.g. by task analysis [Schreiber *et al.*, 1999], or an ontology-based system, e.g. by developing with patterns [Clark *et al.*, 2000; Staab *et al.*, 2001; Hou *et al.*, 2002] or developing subontologies and merging them [Noy and Musen, 2000; Rahm and Bernstein, 2001]. As different as these methods are, they may be characterized by subdividing the task of building a large ontology by engineering, re-using and then connecting smaller parts of the overall ontology.

Though FONTE shares its goal with these methodologies is its rather different in its operationalization. FONTE does not aim at a partitioning and re-union (by merge or align with recognition of similarities) of the problem space, but rather by a factorization into primordial concepts and a subsequent combination $\otimes$ that is more akin to a Cartesian product than a union of ontologies.

2 Temporal Ontology

The time and general events ontology used embodies many concepts like *Instant*, *Period* or *Process* routinely found in ‘standard’ ontologies like Time-DAML or SUMO.

Besides the classes *TemporalEntity* and *Eventuality* similar to the ones used in Time-DAML a third class (*TimedThing*) was defined to capture the notion that bridges between temporal concepts and the domain concepts that will be used during the assemble process. In particular, we have included the notion of *Role* as a core concept. While there are concepts that give identity to their instances (i.e. they are *semantically rigid* [Guarino and Welty, 2000]), e.g. while the identity of a particular person depends on being an instance of *Person*, the identity of the same person does not change when

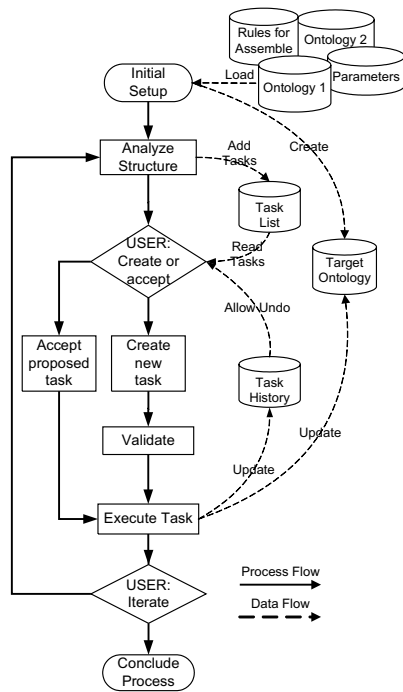


Figure 1: Assembly main process

it ends being a *student* and starts being a *professor*. Thus, the notion of *Role* is important when connecting a temporal theory with a concrete domain.

3 The Assembly Process

The assembly process comprises two main building blocks. First, the specification of temporal aspects for a time-less domain ontology remains dependent on the conceptualization of the ontology engineer. Therefore, it is very important that the engineer may interactively influence the process. Second, in order to facilitate and accelerate the assembly of time-less domain concepts with temporal notions, the interactive process is supported by heuristics asking and pointing the engineer.

The assembling process runs as depicted in Figure 1: It starts by an *Initial Setup*. Some basic operations are performed, namely loading the ontologies to be assembled, loading a set of rules to drive the process and initializing some process parameters. The rules and parameters are defined separately from the tool in order to allow for adaptations to the particular needs of different temporal ontologies. However the rules and parameters do not change when a new domain ontology is to be assembled. The *Target Ontology* initially corresponds to the union of the time-less domain ontology, *O1*, and the temporal theory, *O2*.

In the *Analyze Structure* step a set of tests are performed that restrict the set of possible task instances to plausible ones, which are then proposed by insertion into the *Task List*. As more information becomes available in subsequent iterations, the usefulness of results provided by the structure analysis improves.

In every iteration the engineer decides whether to accept an automatically proposed task instance from the *Task List*. Alternatively, the user may take the initiative and assemble

a new task instance from scratch. Then a set of logical tests (*Validate*) are performed in order to detect the existence of any knowledge anomalies (e.g. circularity or redundancy). In contrast, the acceptance of a proposed task instance does not require further checks as the checks are tested for validity before the user sees them.

By the *Execute Task* step the corresponding changes are made to the target ontology. Thereafter, the user decides either to pursue another iteration or to go to *Conclude Process* and accept the current *Target Ontology* as the final version.

4 Evaluation and Conclusion

In order to evaluate the effectiveness of FONTE, we have numerically evaluated the assembly tasks proposed and executed for an ontology about semantic web research community and the temporal ontology briefly presented in section 2. We have investigated how many assembling steps were proposed and evaluated their adequacy. The study results suggest that indeed FONTE provided a way to factorize the complexity of building large applications leading to more reliable and cheaper final products.

Though, so far, we have only studied the assembly of time into a given ontology, we conjecture that FONTE may also be applied to integrate other important concepts like space, trust, or user access rights — concepts that pervade a given ontology in intricate ways such that a method like FONTE is needed in order to factorize engineering complexity out leading to more consistent and cheaper target ontologies.

Acknowledgments This work was supported by a Marie-Curie Fellowship and by FCT (Portuguese Science and Technology Foundation) through the programs ONTOMAPPER (POSI-41818) and SANSKI (POCTI-41830). We thank our colleagues, particularly N. Silva, G. Stumme, and J. Tane.

References

- [Clark *et al.*, 2000] P. Clark, J. Thompson, and B. Porter. Knowledge patterns. In *KR2000*, pages 591–600, 2000.
- [Guarino and Welty, 2000] N. Guarino and C. Welty. A formal ontology of properties. In *Knowledge Acquisition, Modeling and Management*, pages 97–112, 2000.
- [Hou *et al.*, 2002] Chih-Sheng Johnson Hou, N. F. Noy, and M. A. Musen. A template-based approach toward acquisition of logical sentences. In *Procs.Intelligent Information Processing 2002 - World Computer Congress*, Montreal, Canada, 2002.
- [Noy and Musen, 2000] N. Noy and M. Musen. Prompt: Algorithm and tool for automated ontology merging and alignment. In *Procs.AAAI-2000*, 2000.
- [Rahm and Bernstein, 2001] Erhard Rahm and Philip A. Bernstein. A survey of approaches to automatic schema matching. *VLDB Journal*, 10(4):334–350, 2001.
- [Schreiber *et al.*, 1999] G. Schreiber, Akkermans H., Anjewierden A., R. Hoog, N. Shadbolt, W. Van de Velde, and B. Wielinga. *Knowledge engineering and management. The CommonKADS Methodology*. MIT Press, 1999.
- [Staab *et al.*, 2001] S. Staab, M. Erdmann, and A. Maedche. Engineering ontologies using semantic patterns. In *Procs. IJCAI-01 Workshop on E-Business & the Intelligent Web*, 2001.
- [Wiederhold, 1994] G. Wiederhold. An algebra for ontology composition. In *Proc.Workshop on Formal Methods*, pages 56–61, Monterey, 1994.

A Logic-based Algorithm for Image Sequence Interpretation and Anchoring

Paulo Santos* and Murray Shanahan
Dep. of Electrical and Electronics Engineering
Imperial College, London
{p.santos, m.shanahan}@imperial.ac.uk

Abstract

This paper describes a logic-based framework for interpretation of sequences of scenes captured by a stereo vision system of a mobile robot. An algorithm for anchoring and interpretation of such sequences is also proposed.

1 Introduction

In this work we extend the logic-based spatial reasoning system for scene interpretation proposed in [Santos and Shanahan, 2002] and develop an algorithm that encodes the interpretation process. This algorithm also accounts for the process whereby mappings between logical symbols and sensor data are built up and maintained over time. This is an aspect of the so-called *symbol anchoring* problem [Coradeschi and Saffiotti, 2000].

This paper assumes a stereo vision system embedded in a mobile robot as the source of data about the world. A symbolic representation of the sensor data from the vision system is constructed assuming a *horizontal slice* of each snapshot. A horizontal slice is, in effect, a 2D *depth profile* of the scene before the robot, taken at a particular height. Within these depth profiles, peaks occur that are caused by nearby objects or collections of objects, these peaks are called *depth peaks*. The size and disparity values of single depth peaks, the distance between pairs of peaks and the transitions that occur in these attributes through consecutive pairs of profiles are the building blocks for our spatial reasoning theory.

Within this framework scene understanding is understood as a process of hypothesising the existence and the dynamic relationships between physical objects (and between physical objects and the observer) assuming temporally ordered sequences of depth profiles. This process recalls sensor data assimilation as abduction first proposed in [Shanahan, 1996]. In fact the initial motivation for the present research was to propose a new qualitative background theory about space-time within this framework by using notions from qualitative spatial reasoning theories such as [Randell *et al.*, 1992].

*Supported by CAPES.

2 Depth Profiles

This work assumes sequences of depth profiles as temporally ordered snapshots of the world. A sketch of a depth profile is shown in Figure 1b.

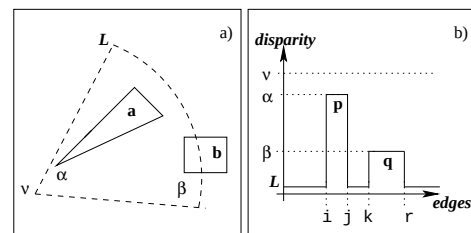


Figure 1: a) Two objects a , b noted from a robot's viewpoint v ; b) Depth profile (relative to the viewpoint v in a) representing the objects a and b respectively by the peaks p and q .

The axis *disparity* in depth profiles is constrained by the furthest point that can be noted by the robot's sensors, this limiting value is represented by L in these charts (Figure 1b). In fact, L is determined by the specification of the robot's sensors.

Differences in the size (and/or disparity) of peaks and transitions of the size (disparity) of a peak in a sequence of profiles encode information about dynamical relations between objects in the world and between the objects and the observer. This is the initial insight upon which the scene interpretation system proposed in this paper is based. The next section presents some relations that represent transitions in depth profiles.

3 Relations on Depth Profile Transitions

This work assumes a relation $peak_of(p, b, v)$, $p_o(p, b, v)$ for short, representing that there is a peak p assigned to the physical body b with respect to the viewpoint v . In this work we are dealing with the viewpoint of a single robot, therefore we abbreviate $p_o(p, b, v)$ to $p_o(p, b)$. The relation $p_o/2$ plays a similar role of the *predicate grounding relation* defined in [Coradeschi and Saffiotti, 2000].

Assuming that the symbols a and b represent physical bodies, p and q depth peaks, and t a time point,

the following relations are investigated in this paper: $extending(p_o(p, b), t)$, states that the disparity value of a peak p is increasing at time t ; $shrinking(p_o(p, b), t)$, states that the disparity value of a peak p is decreasing at time t ; $approaching(p_o(p, a), p_o(q, b), t)$, represents that two peaks p and q are approaching each other at time t ; $receding(p_o(p, a), p_o(q, b), t)$, states that two peaks p and q are receding each other at time t ; $coalescing(p_o(p, a), p_o(q, b), t)$, states that two peaks p and q are coalescing at time t ; $splitting(p_o(p, a), p_o(q, b), t)$, states the case of one peak splitting into two distinct peaks p and q at time t . These relations are hypotheses assumed to be possible explanations for transitions in the attributes of a peak (or set of peaks).

Similarly to [Santos and Shanahan, 2002], the relations described above can be connected to descriptions of transitions on sensor data and, further, to relations about changes in the robot environment by means of sets of axioms. Due to space restrictions, however, we do not present axioms for these relations.

Informally, $coalescing(p_o(p, a), p_o(q, b), t)$ and $splitting(p_o(p, a), p_o(q, b), t)$ can be related, respectively, to the event of an object a **occluding** an object b and of a **appearing from behind** b . Therefore, once a hypothesis on peak transition has been obtained, a relative hypothesis on objects in the world can be inferred from the appropriate axiom stating this connection. This is a central idea underlying our solution for anchoring. In fact, this solution assumes processes of explanation and expectation as described in the following sections.

4 Sensor Data Assimilation

Following the ideas proposed in [Shanahan, 1996] and [Santos and Shanahan, 2002], the task of the abductive process for sensor data assimilation is to infer the relations described in Section 3 as hypotheses given a description (observation) of the sensor data in terms of depth peaks transitions. More formally, assuming that Ψ is a description (in terms of depth profiles) of a sequence of stereo images, and Σ is a background theory comprising axioms connecting the relations in Section 3 to sensor data transitions and to changes in the robot's environment, the task of assimilation as abduction is to find a set of formulae Δ , such that

$$\Sigma, \Delta \models \Psi.$$

5 Expectation

Expected peak transitions are suggested by the conceptual neighbourhood diagram (CND) of the relations described in Section 3, shown in Figure 2¹.

Given an abduced explanation for a transition on a pair of peaks, the expected future relations involving these peaks are the neighbours of this transition in the diagram in Figure 2. As this process generates multiple competing expectations, only those that are verified by further sensor data lead to the

¹For brevity, we are omitting from this diagram the relations $extending/2$ and $shrinking/2$.

generation of new predictions. In practice, prediction in this work is, thus, reduced to a table look-up procedure.

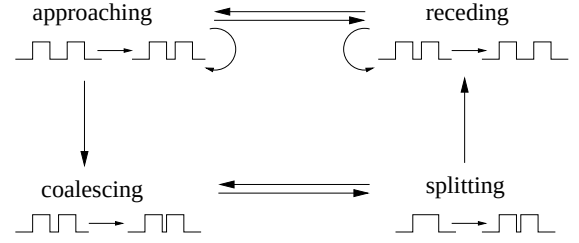


Figure 2: Conceptual neighbourhood diagram.

6 Image segmentation

From a practical point of view, horizontal slices defining depth profiles are comprised of a sequence of measurements made by the considered off-the-shelf vision system. Each horizontal slice is then segmented into *peaks* and *backgrounds* via a simple threshold mechanism on disparity values. This segmentation procedure outputs a sequence of first-order terms of the form: $peak(Lb, Disp, Size)$ and $background((Lb1, Lb2), Disp1, Size1)$, where Lb , $Lb1$ and $Lb2$ are variables for $p_o/2$ terms labelling peaks. In particular $Lb1$ and $Lb2$ are the labels of the peaks bounding the background segment; $Disp$ and $Size$ are the values of disparity and size of the peak Lb ; and, $Disp1$ and $Size1$ are the disparity and size values of the background segment labelled $(Lb1, Lb2)$. If it is the case that a background is bounded by one of the end points of a depth profile (borders of the field of view), a corresponding symbol is assigned to compose its label. In other words, in such cases, either the symbol `borderLeft` or `borderRight` are part of the pair defining the label of a background segment.

Subsequent pairs of horizontal-slice descriptions are input to an algorithm that provides the appropriate interpretation in terms of the dynamic relations discussed in Section 3. This algorithm is summarised in the next section.

7 The anchoring and interpretation algorithm

This section describes an algorithm for interpretation and anchoring, so called *A&I algorithm*, that works by matching peak segments in consecutive pairs of profile descriptions that are depicting the same object in the world. Each profile description is comprised of a finite number of segments peak and background, which are provided by the segmentation routine described in the previous section.

In order to describe the *A&I algorithm* let P_i and P_{i+1} be one consecutive pair of profiles, we assume that P_i has n segments and P_{i+1} has m . In this section we denote a segment r in a profile k by $s_{k,r}$, thus, P_i and P_{i+1} can be denoted as:

$$P_i = \{s_{i,1}, s_{i,2}, \dots, s_{i,n}\}$$

and

$$P_{i+1} = \{s_{i+1,1}, \dots, s_{i+1,m}\}$$

Let $PRED$ and INT be initially empty lists containing, respectively, a set of predictions and a set of interpretations

for the profile pair (P_i, P_{i+1}) . We assume also a symbol o denoting a physical body.

For every subsequent pair of profile descriptions (P_i, P_{i+1}) , the algorithm's task is to match pairs of segments $(s_{i,j}, s_{i+1,r})$ in (P_i, P_{i+1}) that depict the same object in the world. The segments are considered from left to right in the profile descriptions, starting with the first pair of segments $(s_{i,1}, s_{i+1,1})$. The algorithm is summarised as follows.

Begin: (A&I algorithm)

While (P_i and P_{i+1} are non-empty)

1. if $s_{i,j}$ and $s_{i+1,r}$ are peak segments; then
 - (a) if $s_{i,j}$ is bound to a term $p_o(j, o)$, then assign the term $p_o(r, o)$ to the label of $s_{i+1,r}$;
 - (b) else, create two new terms $p_o(j, o)$ and $p_o(r, o)$ and assign them respectively to the labels of $s_{i,j}$ and $s_{i+1,r}$;
 - (c) compare the size and disparity measurements in both segments and put into *INT* the appropriate interpretation according to the relations *extending* and *shrinking*, in Section 3;
 - (d) from these interpretations and the conceptual neighbourhood diagram in Figure 2 obtain the predicted predicates and insert them into *PRED*;
 - (e) return the next pair of segments: $(s_{i,j+1}, s_{i+1,r+1})$;
2. if $s_{i,j}$ and $s_{i+1,r}$ are background segments; then
 - (a) unify their labels;
 - (b) compare the sizes of $s_{i,j}$ and $s_{i+1,r}$ and consider the difference between these values, which gives the difference in the distance between the two peaks bounding $s_{i,j}$ and $s_{i+1,r}$;
 - (c) interpret this difference in terms of the relations *approaching* and *receding*, and insert the appropriate interpretation into *INT*;
 - (d) from this interpretation and the CND in Figure 2 insert the relative predictions into *PRED*;
 - (e) return the next pair of segments: $(s_{i,j+1}, s_{i+1,r+1})$;
3. if $s_{i,j}$ is a background segment and $s_{i+1,r}$ is a peak segment then check in the prediction set *PRED* whether the two peaks bounding $s_{i,j}$ have been expected to be *coalescing* or *splitting*,
 - (a) if so, assume this expectation as explanation for $s_{i,j}$ and $s_{i+1,r}$, inserting it into *INT*. Execute the appropriate unifications on peak variables and terms $p_o/2$ as explained in step 1a and 1b. Then, obtain the predicted predicates, inserting them into *PRED*. Return the pair $(s_{i,j}, s_{i+1,r+1})$ to be considered for interpretation;
 - (b) else explain away the peak $s_{i,j}$ as noise and assume the pair $(s_{i,j+1}, s_{i+1,r})$ for interpretation;
4. if $s_{i,j}$ is a peak segment and $s_{i+1,r}$ a background segment then proceed analogously to the previous case.

End while.

- If P_i is empty, get the next pair of profile descriptions (P_{i+1}, P_{i+2}) .

- else, if P_{i+1} is empty but P_i is not, then explain away the remaining peak segments in P_i and consider the next pair of profile descriptions (P_{i+1}, P_{i+2}) .

The while loop above is repeated until there are no more depth profile descriptions.

End.

Informally, the algorithm above considers consecutive pairs of depth profile descriptions as lists. Pairs of elements from these lists are compared and a match for peak segments is obtained according to the expected transitions in depth profiles. The result of this process is the interpretation of peak transitions, which are related to changes in the objects depicted by peaks.

The algorithm main processing is the while loop that considers, in pairs, every segment in the depth profile descriptions. The running time of this algorithm is, thus, linear on the size of the list containing these descriptions. However, as the number of segments in these lists is related to the number of objects depicted by the considered depth profiles, and there are only a finite number of objects in each scene, the asymptotic upper bound of this algorithm is $\mathcal{O}(1)$.

8 Conclusion

This paper presents a new step towards an abductive framework for sensor data interpretation of sequences of stereovision images obtained by a mobile robot's vision system. We proposed a symbolic representation of the stereovision data based on depth profiles obtained from horizontal slices of snapshots of the world. These profiles encode information about objects in the world as peaks. Each profile was, then, segmented into first-order statements that were input of an algorithm whose task was to unify pairs of segments, providing the interpretation for the occurred transition in depth peaks.

Further research has to consider two main open issues in this framework. First, we have to develop of a qualitative theory about object's shape from multiple horizontal slices of a scene. This theory would enhance the representation of the environment and, therefore, improve the capabilities of the reasoning system. A second issue is the development reasoning modules capable of explaining away noise patches in the image sequences according to common-sense knowledge about the world and about the sensor's limitations.

References

- [Coradeschi and Saffiotti, 2000] S. Coradeschi and A. Saffiotti. Anchoring symbols to sensor data: Preliminary report. In *Proc. of AAIL*, pages 129–135, Austin, U.S., 2000.
- [Randell et al., 1992] D. Randell, Z. Cui, and A. Cohn. A spatial logic based on regions and connection. In *Proc. of the KR*, pages 165–176, Cambridge, U.S., 1992.
- [Santos and Shanahan, 2002] P. Santos and M. Shanahan. Hypothesising object relations from image transitions. In Frank van Harmelen, editor, *Proc. of ECAI*, pages 292–296, Lyon, France, 2002.
- [Shanahan, 1996] M. Shanahan. Robotics and the common sense informatic situation. In *Proc. of ECAI*, pages 684–688, Budapest, Hungary, 1996.

POSTER PAPERS

INFORMATION RETRIEVAL AND DATA MINING

Learning Consumer Photo Categories for Semantic Retrieval

Joo-Hwee Lim

Institute for Infocomm Research
21 Heng Mui Keng Terrace, Singapore 119613
jooohwee@i2r.a-star.edu.sg

Jesse S. Jin

University of New South Wales
Sydney 2052, Australia
jesse@cse.unsw.edu.au

Abstract

In this paper, we develop a computational learning framework to build a hierarchy of 11 consumer photo categories for semantic retrieval. Two levels of visual semantics are learned for image content and image category statistically. We evaluate the average precisions at top retrieved photos on 2400 heterogeneous consumer photos with very good result.

1 Introduction

Research on image categorization has received more attention lately. In particular, the efforts to classify photos based on contents have been devoted to: indoor versus outdoor [Bradshaw, 2000], natural versus man-made [Bradshaw, 2000; Vailaya et al., 2001], and categories of natural scenes [Vailaya et al., 2001]. In general, the classifications were made based on low-level features such as color, edge directions etc and [Vailaya et al., 2001] presented the most comprehensive coverage of the problem with a hierarchy of 8 categories.

In this paper, we deal with a more comprehensive hierarchy of 11 categories (Fig. 1) for 2400 consumer photos. These photos (Fig. 2), including photos of bad quality, present a real spectrum of complexities when compared to the photos used in previous works [Bradshaw, 2000; Vailaya et al., 2001], which are mainly professional photos from the Corel stock photo library. Furthermore, only 20% of our test collection is used for training and we evaluate our approach in terms of average precisions (over 10 runs) of semantic retrieval at top retrieved photos which is important for practical usage. Our approach is unique that we compute *uniform semantic* features at both image content and image category levels using a computational learning framework instead of low-level features crafted for different categories.

At the image content level, salient image regions that exhibit semantic meanings are adopted as training examples to construct *semantic support regions* (SSR) that span a new indexing space. Local image regions of a photo is projected into this space as linear combinations of the SSR and further aggregated spatially to form image content signature for similarity matching. At the

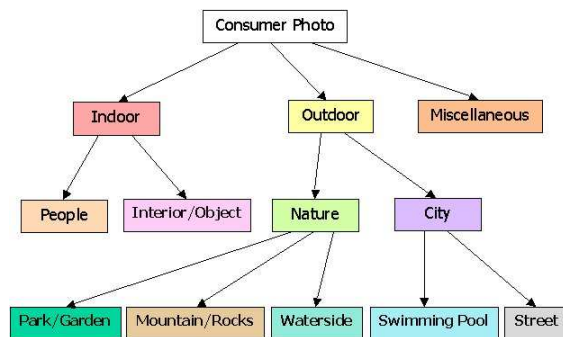


Figure 1: A hierarchy of consumer photo categories

image category level, we learn image category models with small number of labeled photos to compute the relevance measure of photos in a winner-take-all approach.

1.1 Learning Semantic Support Regions

SSR are salient image patches that exhibit semantic meanings to us. A cropped face region, a typical grass patch, and a patch of swimming pool water etc can all be treated as their instances. To compute the SSR from training instances, we adopt support vector machines. We extract suitable features such as color and textures for a local image patch and denote this feature vector as x . A support vector classifier S_i devoted to a class i of SSR is treated as a function on x , $S_i(x)$. Then the posterior probability of class i can be computed as

$$P(S_i|x) = \frac{\exp^{S_i(x)}}{\sum_k \exp^{S_k(x)}}. \quad (1)$$

For the experiments described in this paper, since we are dealing with heterogeneous consumer photos, we adopt color (means and standard deviations of each color channel) and texture features (means and standard deviations of Gabor coefficients) to characterize SSR.

To detect SSR with translation and scale invariance in an image, the image is scanned with windows of different scales. To reconcile the detection maps across different resolutions onto a common basis, we adopt the following principle: If the probability of the most probable class of a region j at resolution k is less than that of a larger

region (at resolution $k+1$) that subsumes region j , then the classification probabilities of region j should be replaced by those of the larger region at resolution $k+1$. To aggregate the classification probabilities of the reconciled detection map for a spatial area X that comprises of x small equal regions with feature vectors $x_1, x_2, \dots, x_n$, we compute the expected value of $\hat{P}(S_i|X)$ over x_j as $\frac{1}{n} \sum_j P(S_i|x_j)$ since $P(x_j)$ are equal for all small regions j . The similarity between two corresponding blocks X_j in image X and Y_j in image Y is computed as cosine between the probability vectors $\hat{P}(S_i|X_j)$ and $\hat{P}(S_i|Y_j)$. The overall similarity between X and Y is the average of the cosine values over all blocks.

1.2 Learning Semantic Image Categories

A photo category M_i is also learned using support vector machines. The input patterns to M_i are the indexes of the images ($Z = \hat{P}(S_i|X_j)$). The support vector learning computes the support images for the categories from a set of labeled photos. Given an unlabeled photo of index Z , the output of a category M_i is $S(M_i, Z)$. With the winner-take-all approach, we compute the winner k as $k = \text{argmax}_i S(M_i, Z)$. Then the relevance measure of Z to category M_i is defined as

$$R(M_i, Z) = S(M_k, Z) \text{ if } i = k; 0 \text{ otherwise} \quad (2)$$

2 Empirical Evaluation

From the 2400 photos, we define ground truth lists for the 11 categories. Their sizes are listed in Table 1 as breadth-first order of Fig. 1 and examples are shown in Fig. 2. We designed 26 classes of SSR: people (face, figure, crowd, skin), sky (clear, cloudy, blue), ground (floor, sand, grass), water (pool, pond, river), foliage (green, floral, branch), mountain (far, rocky), building (old, city, far), interior (wall, wooden, china, fabric, light). We cropped 554 image regions from 138 images and used 375 of them as training data for support vector machines to compute the SSR and the remaining 179 as test data to gauge generalization performance. Among all the kernels tried, a polynomial kernel with degree 2 and constant 1 gave the best result on precision and recall.

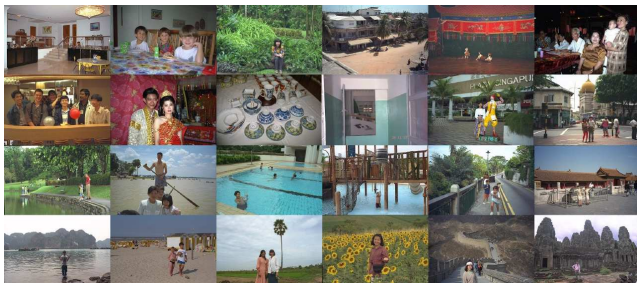


Figure 2: Two sample photos for each category (top-down, left-to-right): indr, outd, misc, inpp, inob, city, natr, pool, strt, wtsd, park, mtrk

We used 20% of the 2400 photos for training. We generated 10 different sets of positive training samples from the ground truth list for each category based on uniform random distribution. The negative training samples of a given category are positive training samples from other categories that do not overlap with the category. The evaluation of retrieval precision is carried out hierarchically with respect to the category tree in Fig. 1. The test data for the category of a child node of a run is the ground truth list of its parent node minus the training samples used for learning the category of the child node in the run. For example, to evaluate the retrieval performance of nature (natr) photos, the ground truth list of outdoor less the training sample for building the nature category is taken as the test data. The learning and retrieval of each category were performed 10 times and the average precisions (over 10 runs) of top retrieved images are given in Table 1.

Table 1: Average precisions at top numbers of photos

Avg.Prec.	Size	Top 20	Top 30	Top 50
indr	994	0.94	0.96	0.96
outd	1218	1.00	1.00	1.00
inpp	860	0.99	0.99	0.99
inob	134	0.84	0.75	0.56
natr	521	0.96	0.96	0.95
city	697	0.95	0.94	0.93
park	304	1.00	0.99	0.98
mtrk	67	0.41	0.27	0.16
wtsd	150	0.92	0.89	0.66
pool	52	0.47	0.32	0.21
strt	645	0.99	0.99	0.99

From Table 1, we observe that up to first 50 images, a user (on average) gets almost all relevant photos of the respective categories except less so for categories interior/object (inob) and waterside (wtsd), and even less so for categories mountain/rocks (mtrk) and swimming pool (pool). The reasons for poorer performance are 2-fold. First these categories have much fewer positive training samples (i.e. 27, 30, 13, 10). Next, they comprise images of varied contents (c.f. Fig. 2: objects plus interior, mountain and rocks, lakeside plus beach, pool with and without water as focus). We believe that with more training samples, their performance would be raised.

References

- [Bradshaw, 2000] B. Bradshaw. Semantic based image retrieval: a probabilistic approach. In *Proc. of ACM Multimedia*, pp. 167-176, 2000.
- [Vailaya et al., 2001] A. Vailaya et al. Bayesian framework for hierarchical semantic classification of vacation images. *IEEE Trans. on Image Processing*, 10(1): 117-130, 2001.

Intelligent Multimodal Stream Processing

Mark Maybury

Information Technology Division

The MITRE Corporation

202 Burlington Road

Bedford, MA 01730, USA

maybury@mitre.org

www.mitre.org/resources/centers/it

Abstract

This poster describes methods to enable intelligent access to multimodal information streams. We illustrate these methods in two integrated systems: the Broadcast News Editor (BNE) which incorporates image, speech, and language processing and the Broadcast News Navigator (BNN) which provides search, visualization and personalized access to broadcast news video. BNN enables users to perform keyword and named entity search, temporally and geospatially visualize entities and stories, cluster stories, discover entity relations, and obtain personalized multimedia summaries. By transforming access from sequential to direct search and providing hierarchical hyperlinked summaries, BNE and BNN enable users to access topics and entity news clusters nearly three times as fast as direct search of video.

1. Intelligent News on Demand

Figure 1 illustrates the BNE and BNN systems that integrate components from MITRE, Oracle, Carnegie Mellon University and Lincoln Laboratory to process imagery, audio, and text to enable news on demand. Key elements of BNE include scene change detection and classification from imagery, silence and speaker change detection from audio, and named entity extraction from speech or closed caption transcripts. A correlation component extracts cues from across these streams to detect the start and end of broadcasts, commercials, and stories. Subsequently, stories are classified and summarized, including selection of key frames, named entities, and representative sentences. Finally, the user can search and explore video stories, provide relevancy feedback, specify media preferences, and obtain personalized displays using the BNN web browser as shown in Figure 4 below. As detailed in [Maybury 2003], BNE and BNN transform video access from sequential to direct search, providing novel navigation and discovery mechanisms such as topical and entity-specific news clusters. Empirical evaluations have shown that users can find stories and answer questions nearly three times as fast as searching digital video with no loss in precision and recall.

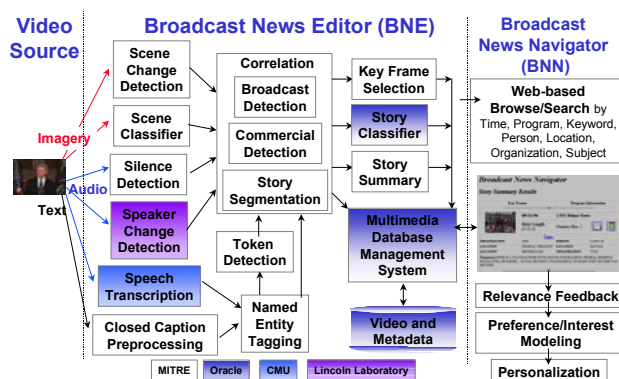


Figure 1. Multistream Broadcast News Understanding: System Architecture

2. Intelligent Multimodal Segmentation

BNN exploits text, audio, and imagery streams and associated cues to detect story and commercial segment boundaries and to select media elements to use for summaries and multimodal displays. Underlying BNN is a set of machine-learned, time-enhanced finite state automata modeling news structure that take into account the above cues and probabilistic, temporal models of event occurrence [Boykin and Merlino 2000]. Program start/end, anchor/report shots, commercials, and/or story shifts are inferred from cross media cues in text (e.g., frequent weather or sports terms, funding and/or copyright notices), discourse cues (e.g., “coming up next”), music (e.g., characteristic jingles), silence (indicating breaks to commercial), and visual cues (e.g., logos, anchor vs. reporter shots). For example, as shown in Figure 2, our frequency analysis of news anchor sign on terms (e.g., “hello”, “welcome”) or sign off terms (e.g., “that’s all”, “thanks for watching”) occurring minutes from program start enables the creation of pattern detectors for such events. We have discovered analyzing months worth of news, for example, that sign off terms occur in 97% of news programs.

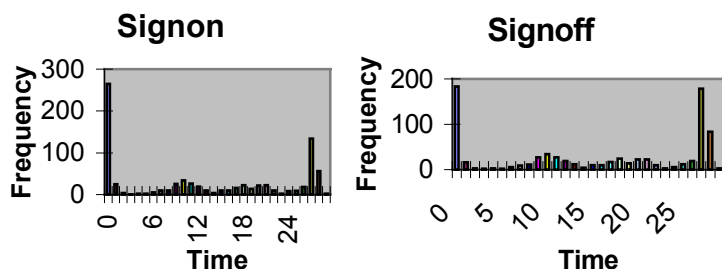


Figure 2. Text Analysis of Program Start and End Terms

Analogously, as shown in Figure 3, BNE classifies 15 frames per second into images associated with program start (e.g., logos), commercial breaks (e.g., blackframes), single and double anchor shots, and reporter shots.

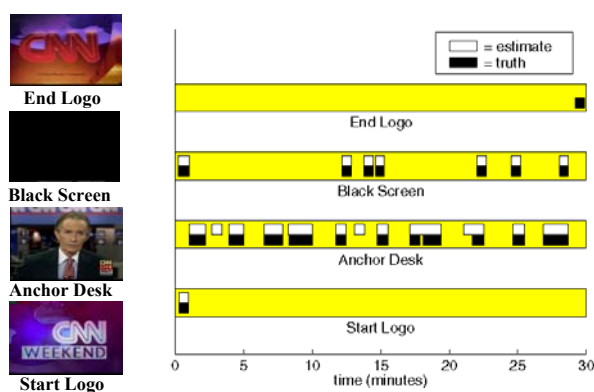


Figure 3. Imagery Analysis

3. Search, Retrieval, and Visualization

After BNE segments stories with multimodal cues, BNN supports the retrieval of stories by source, date range, keyword, named entity, or topic query. As exemplified in the upper left of Figure 4, after selecting news programs (e.g., CNN NewsNight, ABC World News Tonight) and indicating date ranges (e.g., February 6-9, 2003), a user types in keywords in the text box. If a user is unfamiliar with retrieval terms, they can select from an alphabetic listing of all the named entities extracted for the time period and from the programs of interest, such as shown in the person, organization, and location lists in the upper left of Figure 4 (note “Hans Blix” is highlighted). This retrieves 14 matching stories and displays them as a “Story Skim”, the source and date, the top 3 named entities in each story, and a representative key frame from each selected segment. Finally, by selecting a story, the user is presented with “Story Details”, including all named entities, a text summary, and access to video and transcript sources. In empirical studies this mix of extracted media and hypertext organization enables analysts to perform searches three times faster than with video source.



Figure 4. Text and Named Entity Search Menus, “Story Skim” Results and “Story Detail” Results

Figure 5 illustrates a user visualizing named entity frequencies and animating story occurrences associated with geospatial regions over time enabling information discovery.

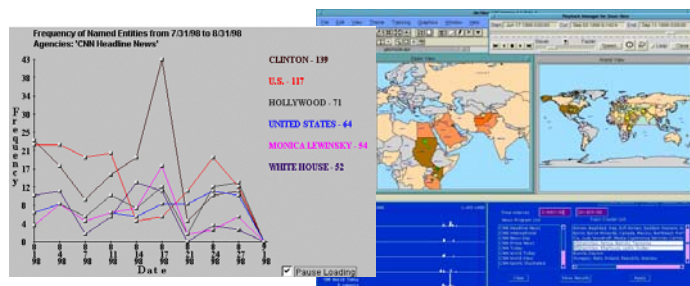


Figure 5. News Visualization

Acknowledgments

This research involved collaboration among Stanley Boykin, Andy Merlino, Warren Grieff, and Chad McHenry, Chris Clifton, John Aberdeen, John Burger, David Day, Lynette Hirschman, and Rod Holland.

References

- [Boykin and Merlino, 2000] Boykin, S. and Merlino, M. Feb. 2000. A. Machine learning of event segmentation for news on demand. *Communications of the ACM*. Vol 43(2): 35-41.
- [Maybury 2003] Maybury, M. 2003. Broadcast News Understanding and Navigation. Innovative Applications of AI. Acapulco, Mexico. 12-14 August, 2003.

Collaborative Web Search*

Barry Smyth, Evelyn Balfe, Peter Briggs, Maurice Coyle, Jill Freyne

Smart Media Institute, University College Dublin, Ireland

{firstname.lastname} @ucd.ie

Abstract

Web search engines struggle to satisfy the needs of Web users. Users are notoriously poor at representing their needs in the form of a query, and search engines are poor at responding to vague queries. However progress has been made by introducing context into the search process. In this paper we describe and evaluate a novel approach to using context in Web search that adapts a generic search engine for the needs of a specialist community of users. This collaborative search method enjoys significant performance benefits and avoids the privacy and security concerns that are commonly associated with related personalization research.

1 Introduction

It is a tragic irony of the information age that users are finding it increasingly difficult to benefit from access to unprecedented amounts of information. Even with the help of modern search engines we regularly fail to locate relevant information in a timely manner. Many factors contribute to this *information overload* problem. Certainly, the sheer quantity of information, and its growth rate, tax even the most advanced search engines. For example, various estimates indicate that even the largest search engines often cover only a fraction of the available information space [Lawrence and Giles, 1999].

However, search engine coverage is just the tip of the iceberg, and can be greatly enhanced by using meta-search methods [Dreilinger and Howe, 1997; Selberg and Etzioni, 1997]. A more pressing problem is the limited degree to which those pages that are covered can be accurately ranked with respect to a given query. Part of this problem lies with the users. Search engines work very well for properly formulated queries, but they come up short when presented with an average Web user's query, which typically contains about 2 query terms [Lawrence and Giles, 1998]. The inevitable outcome is long lists of apparently relevant results, with genuinely useful results (for the target user) few and far between. Moreover, these problems are exacerbated by the new generation of mobile computing devices (eg. WAP phones and

PDA's). Their restricted input capabilities and limited screen real-estate mean that mobile users are even less likely to provide well formulated queries or tolerate long result lists.

For the most part, recent search engine advances have focused on better ways to handle vague queries by improving existing page analysis, indexing and ranking methods. However a critical shortcoming still remains: a query might include terms that identify the primary information target, but rarely includes terms that usefully describe the search *context*. For example, a simple query for “*jaguar*” does not indicate whether the user is interested in cars or cats, and queries for “*Michael Jordan*” do not distinguish between the basketball star and the Berkeley professor. Consequently, researchers have recently focused on ways to exploit context during search, either by explicitly establishing context up-front or by implicitly inferring it. For example, the Inquirus 2 meta-search engine [Glover *et al.*, 2001] supplements keyword-based queries with a context category; users explicitly select from a set of categories such as “research paper”, “homepage” etc. Alternatively, implicit context can be automatically inferred. For example, systems such as Watson [Budzik and Hammond, 2000] take advantage of user activity prior to search to judge context; Watson monitors a user's word processing activity and uses document text as the basis for query terms. In contrast, relevance feedback techniques attempt to use actual search results to inform context. For example, [Mitra *et al.*, 1998] extract correlated terms from top-ranking search results to focus context on the most relevant search results as opposed to the entire set.

In this paper we describe a novel technique for exploiting context during search: *collaborative search* acts as a front-end for existing search engines and re-ranks results based on the learned preferences of a community of users. We describe and evaluate its implementation in the I-SPY system and highlight how I-SPY achieves personalization in an anonymous fashion, without storing individual user profiles.

2 Collaborative Search

Collaborative search is motivated by two key ideas. First, specialised search engines attract communities of like-minded users and naturally limited context variations. For example, a motoring search engine is likely to attract queries with a motoring theme; here “*jaguar*” queries are more likely to relate to cars than cats. Second, by capturing the selections of

*The support of the Informatics Research Initiative of Enterprise Ireland is gratefully acknowledged.

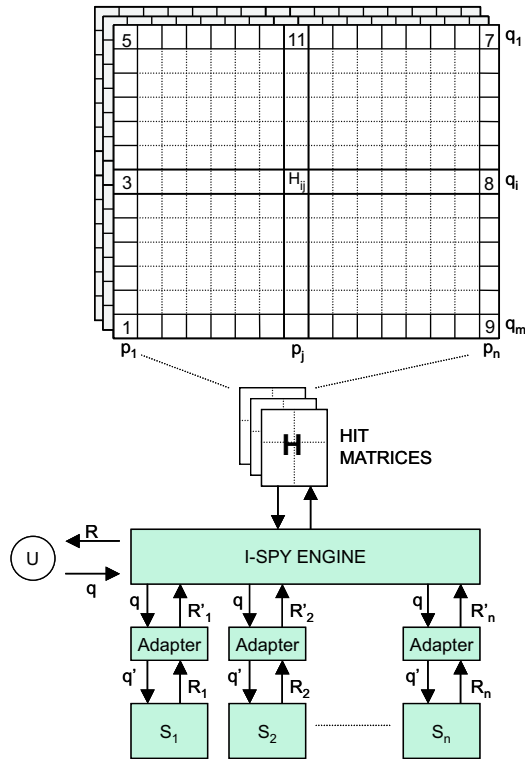


Figure 1: The I-SPY meta-search engine.

a community of users it is possible to estimate query-page relevance as the probability that a page p_j will be selected for query q_i . Collaborative search combines both ideas in a meta-search engine called I-SPY (Figure 1).

The unique feature of collaborative search is its ability to personalize search results for a community of users, but without relying on traditional context-analysis (eg. [Lawrence and Giles, 1998]) or personalization (eg. [Bradley *et al.*, 2000]) techniques. I-SPY achieves this by storing the usage patterns of users as a *hit matrix*, H . Each element, H_{ij} , equals the number of times that page p_j was selected for query q_i . This matrix is a powerful source of relevancy information; after all its data reflect query-page relevancy judgments by users. The relevancy of p_j to query q_i is estimated by the probability that p_j will be selected for query q_i ; see Equation 1.

$$Relevance(p_j, q_i) = \frac{H_{ij}}{\sum_{i=1}^n H_{ij}} \quad (1)$$

This relevancy metric is tuned to the preferences of a community of users, and the queries and pages that they tend to prefer. Deploy I-SPY on a motoring web site and its hit matrix will become populated with queries and pages that are relevant to car fans. For example, over time queries for “*jaguar*” will tend to result in the promotion of car sites because users submitting this query term will tend to select Jaguar car sites, ignoring the wild cat pages. The wild cat pages may still be returned but will be relegated to the bottom of the result list.

In fact I-SPY can deploy multiple I-SPY search agents, each with its own separate hit table. Thus the central I-SPY

engine can be used to service many different search services across a range of portals, for example, each one adapted for the needs of a particular user group through its associated hit matrix. Alternatively, different hit matrices could be associated with different regions of the same site to bias search with respect to different topics. For instance, the work of [Haveliwala, 2002] biases PageRank with respect to different topic groups in an Internet directory by generating category-biased PageRank vectors from the URLs contained in top-level directory categories. A similar strategy can be supported by I-SPY. Placing a search box on the *Programming Languages* directory page will naturally capture queries from this domain. And the behaviour of the users providing these queries, will gradually adjust I-SPY’s relevancy metric and ranking function in favour of Programming Languages pages.

3 Evaluation

For our evaluation we focus on a specific user community and search domain: computer science students and programming languages. A set of 60 queries is produced from the programming languages listed in Yahoo. I-SPY is configured to query two underlying search engines, Yahoo (which uses Google) and Splat!, and each of the 60 queries is submitted to obtain up to 30 results based on a standard meta-search ranking function. A group of 20 computer science students are asked to identify relevant results, based on the summary result descriptions returned by I-SPY.

A *leave-one-out* evaluation methodology is employed so that each user is designated to be a *test* user with the remaining 19 serving as *training* users. The relevancy results of the training users are used to populate I-SPY’s hit matrix and the results for each query are re-ranked using I-SPY’s relevancy metric. Next, we count the number of these results listed as relevant by the test user for various result-list sizes ($k = 5..30$). Finally, we make the equivalent relevancy measurements by analysing the results produced by the untrained version of I-SPY (*Standard*), which serves as a benchmark.

Figure 2 presents the results for I-SPY and the benchmark search engine as a graph of precision versus recall for each result-list size; these are really *bounded* versions of the standard precision and recall metrics and the measures for each engine converge once a complete result-list is returned. The results indicate a significant and consistent benefit for I-SPY over the standard meta-search benchmark. For example, for result-lists of 5 items, I-SPY achieves a precision of just over 96% compared to the standard meta-search precision of only 63%. Similarly, at the same result-list size, we find an average recall for I-SPY of 64% compared to just under 40% for the standard method. Indeed we see that I-SPY achieves 100% recall at just over 20 items whilst it takes the benchmark 30 items to achieve the same level of recall. The fact that larger relative benefits are available at smaller result-list sizes is important. Users rarely sift through large result-lists and so, the more relevant items that can be presented earlier on, the better. This means that I-SPY is likely to be especially valuable in situations where large result lists must be truncated for other reasons, such as the small screen sizes of mobile devices.

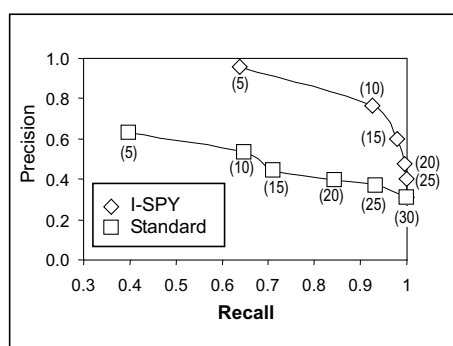


Figure 2: Precision vs. Recall Results.

4 Discussion

There are number of problems with collaborative search that need to be addressed to guarantee its applicability across a broad range of search tasks. Perhaps the most important problem is the so-called *cold-start problem*. This refers to the fact that newly indexed Web pages find it difficult to attract user attention since they will have a low relevancy score using I-SPY's metric and thus appear far down in result-lists, limiting their ability to attract the hits they deserve for a given query. Essentially there is an in-built bias towards older pages.

There are a number of ways that this problem might be dealt with. One is to look at ways to normalize the relevancy of pages with respect to their *age*. For example, we might measure the age of a page by the time (or number of queries) since its first hit and amplify the relevancy of young pages relative to older pages.

Indeed there is another side to this problem. Just as new pages find it difficult to attract hits, so too older pages may find it easy to attract hits. In the worst case scenario this could even bias I-SPY's result-lists towards pages that are likely to be out of date and thus less relevant to current users than they were to past users. Once again, biasing relevance towards new pages should help to cope with this problem.

Of course in general there are many factors that can, and probably should, be taken into account when ranking search results. We have focused primarily on I-SPY's relevancy factor, but other factors such as the age of a page and its meta-search ranking are also appropriate. As part of our future work we will explore how best to combine these factors to produce optimal result rankings. This may or may not involve a direct combination of the rankings. For example, one option is to present search results not as a single list of results, as is normally the case, but perhaps as two or more lists of results in order to emphasise the different qualities of the returned pages. For instance, in general only a subset of search results are likely to have non-zero I-SPY relevance scores; that is, a subset of results will have been selected in the past for the current query. Therefore, it is practical to present the I-SPY results with relevancy scores as special recommendations (ranked by their relevancy). The remaining results can be presented separately, ranked by their meta-score. In turn a third list of *new* pages, ranked by meta-search score or relevancy, can also be separately presented.

5 Conclusions

Improving the accuracy of Web search engines by introducing context into the search process is an important and challenging research problem. We have described a generic search engine that can be adapted or personalized to fit the context and needs of a community of users by using the collaborative search technique. The benefits include superior precision and recall characteristics when compared to a benchmark search engine. In addition, this level of personalization is achieved without the need to store individual user profiles, leading to superior security and privacy benefits when compared to alternative approaches.

In closing it is worth highlighting that collaborative search makes no strong assumptions about the form of the underlying search engines and is generally applicable across a range of content types including Web pages, graphics and photos, audio and video. Finally, its proposed ranking metric is computationally efficient ($O(k)$ in the number of search results) and requires no additional parsing of result pages.

References

- [Bradley *et al.*, 2000] K. Bradley, R. Rafter, and B. Smyth. Case-based User Profiling for Content Personalization. In O. Stock P. Brusilovsky and C. Strapparava, editors, *Proceedings of the International Conference on Adaptive Hypermedia and Adaptive Web-based Systems*, pages 62–72. Springer-Verlag, 2000.
- [Budzik and Hammond, 2000] J. Budzik and K. Hammond. User Interactions with Everyday Applications as Context for Just-In-Time Information Access. In *Proceedings International Conference on Intelligent User Interfaces.*, pages 44–51. ACM Press, 2000.
- [Dreilinger and Howe, 1997] D. Dreilinger and A. Howe. Experiences with Selecting Search Engines Using Meta Search. *ACM Transactions on Information Systems*, 15(3):195–222, 1997.
- [Glover *et al.*, 2001] E. Glover, S. Lawrence, M. D. Gordon, W. P. Birmingham, and C. Lee Giles. Web Search - Your Way. *Communications of the ACM*, 44(12):97–102, 2001.
- [Haveliwala, 2002] T. H. Haveliwala. Topic-Sensitive PageRank. In *Proceedings of the World-Wide Web Conference*. ACM Press, 2002.
- [Lawrence and Giles, 1998] S. Lawrence and C. Lee Giles. Context and Page Analysis for Improved Web Search. *IEEE Internet Computing*, July-August:38–46, 1998.
- [Lawrence and Giles, 1999] S. Lawrence and C. Lee Giles. Accessibility of Information on the Web. *Nature*, 400(6740):107–109, 1999.
- [Mitra *et al.*, 1998] M. Mitra, A. Singhal, and C. Buckley. Improving Automatic Query Expansion. In *Proceedings of ACM SIGIR*, pages 206–214. ACM Press, 1998.
- [Selberg and Etzioni, 1997] E. Selberg and O. Etzioni. The Meta-Crawler Architecture for Resource Aggregation on the Web. *IEEE Expert*, Jan-Feb:11–14, 1997.

A Statistical Model for Flexible String Similarity

Atsuhiko Takasu

National Institute of Informatics

2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan

takasu@nii.ac.jp

Abstract

This paper proposes a statistical model for defining string similarity. The proposed model is based on hidden Markov model and defines string similarity as the combination of similarities of substrings. The proposed model has an advantage that the similarity is flexibly defined and complex parameters for the similarity are learnable from training data.

1 Introduction

String matching is an important problem. Many studies have been done on this problem and developed methods are applied to various problems such as approximate information retrieval from OCR processed and spoken documents, pattern extraction from sensor data and so on.

In order to measure the similarity of strings, edit distance and DP matching have been frequently used. In the edit distance, similarity is defined as the minimum number of operations of insertion, deletion and replacement required to convert one string to another. The DP matching allows us to use weights of the operations which enable flexible definition of string similarity according to objective problems. However, weights should be determined depending on the problem and it is hard work to find appropriate weights manually.

This problem motivates us to develop a statistical model which has high expressive power of string similarity and whose parameters can be learned from training data. This paper proposes a statistical model called dual and variable length hidden Markov model (DVHMM) and applies it to structured string analysis.

2 Statistical Model for String Similarity

For a *base string* and a *compared string*, DVHMM defines their similarity by decomposing them into pairs of substrings and combining the similarities of the substrings.

A DVHMM is a form of Hidden Markov Model (HMM). Instead of producing a string, state transitions of DVHMM produce a pair of strings. Each state of the DVHMM defines the similarity of pairs of strings of fixed lengths in the form of output probability. A state is characterized by a pair of lengths. For example, a state characterized by a pair (2,1) of lengths defines similarities of two consecutive characters in

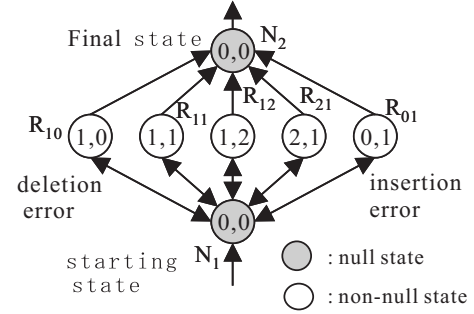


Figure 1: An example of DVHMM

a base string and one character in a compared string. A state (1,0) means that the length of the base (resp. compared) string is 1 (resp. 0), i.e., a delete operation, whereas a state (0,1) corresponds to an insert operation. A state (1,1) corresponds to both a replacement operation and equivalent mapping. Generally, the state corresponding to a pair (i, j) produces a pair of strings whose base string (resp. compared string) length is i (resp. j) with certain probability. States are categorized into *non-null* state that produces a pair of base and compared strings, and *null* state that produces null output and controls state transitions.

Figure 1 shows an example of DVHMM that consists of five non-null states which define the similarity of pairs of substrings of length (1,0), (0,1), (1,1), (2,1) and (1,2). In this figure, output symbols are omitted due to space restrictions. Suppose the alphabet is $\{a, b\}$. Then, the output symbols of the state (2, 1) are $\{(aa, a), (aa, b), (ab, a), (ab, b), (ba, a), (ba, b), (bb, a), (bb, b)\}$ and output probability is assigned to each output symbol.

DVHMM defines joint probability distribution of a pair of strings. Let us consider a base string “ab” and a compared string “a”. Then, the DVHMM in Figure 1 produces the pair of strings (ab, a) by one of the following six sequences of state transitions.

$$\begin{array}{ll} N_1 R_{10} N_1 R_{10} N_1 R_{01} & N_1 R_{10} N_1 R_{11} \\ N_1 R_{10} N_1 R_{01} N_1 R_{10} & N_1 R_{11} N_1 R_{10} \\ N_1 R_{01} N_1 R_{10} N_1 R_{10} & N_1 R_{21} \end{array}$$

Then, the joint probability of (ab, a) is obtained by summing

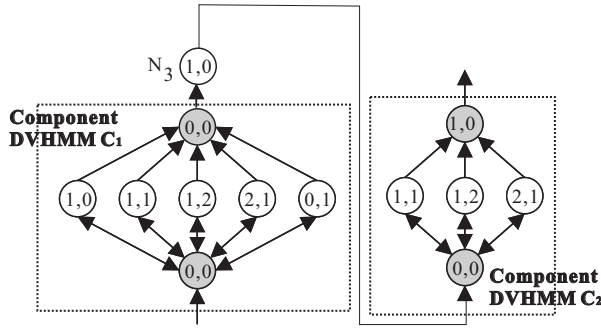


Figure 2: An example of DVHMM for strings consisting of two components

the probabilities of all the five sequences of transitions. Formally, for a base string $\mathbf{b}$ and a compared string $\mathbf{c}$, the joint probability of them is obtained by

$$\sum_{\mathbf{q} \in \mathbf{Q}} P(\mathbf{b}, \mathbf{c}, \mathbf{q}) \quad (1)$$

where $\mathbf{Q}$ is a set of state transitions producing $(\mathbf{b}, \mathbf{c})$ and $P(\mathbf{b}, \mathbf{c}, \mathbf{q})$ is the joint probability of $(\mathbf{b}, \mathbf{c})$ along the state transition $\mathbf{q}$. The joint probability is used as the similarity of a pair of strings.

DVHMM inherits HMM's ability to represent syntactical structure. This ability enables DVHMM to change similarity definition depending on the part of strings. Let us consider strings consisting of two components delimited by special characters. For example, base string consists of two parts delimited by colon and compared string is concatenation of two parts without delimiters. Then, the DVHMM depicted in Figure 2 defines the similarity of these strings, i.e., a component DVHMM C_1 defines similarity of prefix component whereas a component DVHMM C_2 defines the similarity of suffix component. State N_3 defines the delimiter, which produce a pair of colon and null string with probability 1.0. For a given pair ("a:ab", "ab") of string, the DVHMM may separate the pair into ("a", "a") as a prefix component and ("b", "b") as a suffix component, and similarities of these components are measured by different similarity measures, i.e., C_1 and C_2 , respectively. In this case, similarity can be calculated by the joint probability of the pair of string produced by the most likely state transition, i.e.,

$$\max_{\mathbf{q} \in \mathbf{Q}} P(\mathbf{b}, \mathbf{c}, \mathbf{q}) \quad (2)$$

where $\mathbf{Q}$ is the same set in expression (1).

Probabilities of DVHMM can be estimated from training data by the expectation-maximization technique efficiently [Takasu, 2002]. For a given pair $(\mathbf{b}, \mathbf{c})$ of strings, the similarity (2) can be calculated efficiently by a dynamic programming algorithm. We omit the details of the algorithms due to space restriction.

Compared with DP matching, DVHMM enables to

- define the weights in a more detailed manner, and
- calculate the weights systematically.

3 Preliminary Experiment

We applied the proposed method to bibliographic matching which matches references in academic articles with records in bibliographic databases. Reference information is effectively used in digital libraries such as CiteSeer [Lawrence, 1999]. In bibliographic references, bibliographic components such as author's name and article title are located in a specific order and separated by delimiters. The syntactical structure of bibliographic reference can be represented with graphical structure of finite automaton where each state represents a bibliographic component. A DVHMM for bibliographic references is obtained by replacing each state with a component DVHMM like Figure 2 where each component DVHMM defines the similarity of the corresponding bibliographic component. For example, because journal names are often abbreviated, higher probabilities are assigned to deletion errors in the DVHMM for journal name whereas low probabilities are assigned to the deletion errors for page because it is seldom abbreviated in references. In this way, the similarity is changed depending on the bibliographic components, and consequently, more accurate similarity model can be constructed for structured strings.

We carried out an experiment on approximate string matching. In this experiment, we prepared 1,575 references appeared in Japanese academic articles and corresponding records in a bibliographic database, then applied 5-fold cross-validation, i.e., divided them into 5 groups, trained DVHMM using 4 groups of them, and made bibliographic matching using remaining one group. Matching result is ranked records in the database according to the similarity. We used edit distance as baseline similarity and compared the performance of the proposed method with the edit distance. The following table shows the average accuracy that the correct record is ranked within top m records.

rank	1	5	10
ED	92.33	94.56	95.23
DVHMM	94.57	95.10	95.47

As shown in the table the proposed method achieved higher matching accuracy.

4 Conclusion

This paper proposed a statistical model for defining string similarity. The proposed model can represent similarity of strings with the similarities of substrings such as traditional edit distance and DP matching. However, the proposed model can define the similarity more flexible way. This paper shows the proposed model can be applied to the matching of structured strings.

References

- [Lawrence, 1999] S. Lawrence, C. L. Giles, and K. D. Bollacker. Digital libraries and autonomous citation indexing. *IEEE Computer*, 32(6):67–71, June 1999.
- [Takasu, 2002] A. Takasu and K. Aihara. DVHMM: Variable Length Text Recognition Error Model. In *Proceedings of International Conference on Pattern Recognition 2002*.

Mining Video Associations for Efficient Database Management

Xingquan Zhu and Xindong Wu

Department of Computer Science, University of Vermont
Burlington, Vermont 05405, USA
{xqzhu, xwu}@cs.uvm.edu

Abstract

To support more efficient video database management, this paper explores the concept of video association mining, with which the association patterns are characterized by sequentially associated video shots and their cluster information. Given a continuous video sequence V , the video shot segmentation mechanism is first introduced to parse it into discrete shots. We then cluster shots into visually distinct groups and construct a shot cluster sequence by using the class label of each shot. An association mining scheme is designed to mine sequentially associated clusters from the sequence. Those detected associations will convey valuable knowledge for video database management. The experimental results demonstrate the effectiveness of our design.

1 Introduction

Since the 1990s, data mining has been a very active area for research and applications. Many successful techniques have been implemented through academic research and industrial applications [Agrawal and Srikant, 1994][Agrawal and Srikant, 1995][Wu, 1995] [Han and Kamber, 2000]. However, these approaches deal with various databases (like transaction datasets) in which the relationship between data items is explicitly given. Video and image databases are different from these databases. The most distinct feature of video and image databases is that the relationship between any two of their items cannot be explicitly (or precisely) figured out. This inherent complexity of the multimedia data has suggested that mining knowledge from multimedia materials is even harder than from general databases [Zhu *et al.*, 2003][Thuraisingham, 2001][Zaiane, *et al.*, 1998]. Generally, there are two types of video mining techniques: (1) special pattern detection [Zhu *et al.*, 2003], which detects some predefined special patterns; and (2) video clustering and classification [Oh *et al.*, 2002][Pan and Faloutsos, 2002], which clusters and classifies video units into different categories.

Different from these two types of video mining schemes, we address a new research area of video mining, video association mining, in this paper, where associations from video units are used to explore video knowledge. We will present a definition for video associations, and design a video association mining algorithm. As shown in Fig. 1, we first segment a video sequence into shots and cluster shots into groups. Then, we assemble the class information

of each shot to form a shot cluster sequence. We mine sequential associations from the sequence to find clusters with strong correlations and take strongly correlated clusters as video associations.

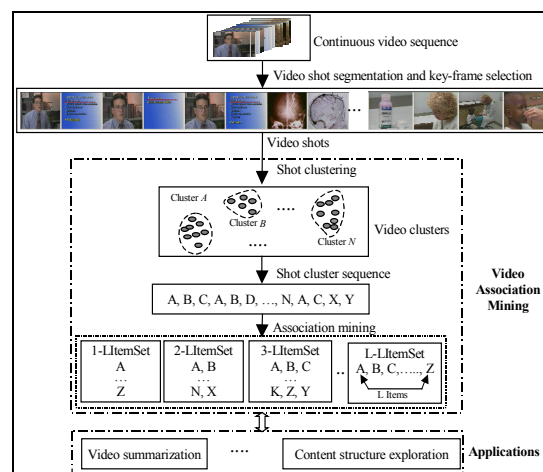


Fig. 1 Video association mining – System architecture

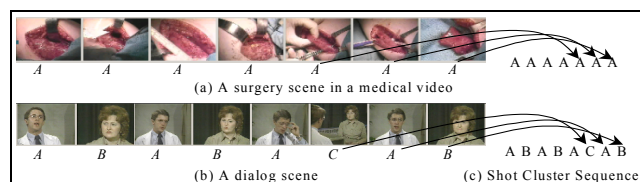


Fig. 2 Some typical video scenes and video data transformation

2 Video Association Mining

Generally, most videos from our daily life are edited by editors, where various kinds of shots are packed as scenes to convey video scenarios, as shown in Fig. 2. There are two typical video scenes: (1) scenes that consist of visually similar shots, as demonstrated in Fig. 2(a); and (2) scenes that consist of visually distinct shots, as shown in Fig. 2 (b). In the first type of scenes, most video shots are visually similar. Take Fig. 2(a) as an example, if we denote each of the shots by “A”, all shots form a sequence “AAAAAAA”, and the self-coherence of “A” indicates an association of itself. We name this type of association as an *intra-association*, i.e., all items in the association are the same. In the second type of scenes, sequential associations exist too. In Fig.2(b), if we denote the actor by “A”, the actress by “B” and the shot containing both of them by “C”, all

shots form a sequence “ABABACAB”. The co-occurrence of “A” and “B” implies an association. We name this type of association as an *inter-association*, i.e., items in the association are different.

Based on the above observations, we define a video association as a sequential pattern with $\{X_1..X_p..X_L; X_i' < X_j' \text{ for any } i < j\}$, where X_i is a video item (see definition below), L denotes the length of the association. X_i' denotes the temporal order of X_i , and $X_i' < X_j'$ indicates that X_i happens before X_j . For an *inter-association*, $X_1 \cap .. \cap X_L = \emptyset$; and for an *intra-association*, $X_1 \cap .. \cap X_L = X_k, k = \forall [1, L]$. For simplicity, we use $\{X\}$ as the abbreviation of an association.

Due to the fact that the temporal information in a video sequence plays an important role in conveying the video content, we integrate traditional association measures (*support* and *confidence*) and video temporal information to evaluate video associations. Some definitions are given as follows:

- An *item* is a basic unit, which denotes a shot cluster.
- Given a shot cluster sequence, the *temporal distance (TD)* between two items is the number of shots between them. E.g., given sequence “ABDEC”, the temporal distance of “AB” is $TD(AB)=0$, and for “AC” is $TD(AC)=3$.
- An *L-ItemAssociation* is a video association that consists of L sequential items. E.g., “ABC” is a 3-ItemAssociation.
- The *L-ItemSet* is an aggregation of all *L-ItemAssociations*, with each of its members being an *L-ItemAssociation*.
- Given a *temporal distance threshold (TDT)* $TDT=T$, the *temporal support (TS)* of an association $\{X_1..X_L\}$ is defined as the number of times that this association appears sequentially in the sequence. In addition, each time this association appears, the temporal distance between any two neighboring items of the association should be no more than T shots. That is, given any X_i and X_{i+1} ($i \geq 1, i+1 \leq L$), $TD(X_i, X_{i+1}) \leq T$.
- Given $TDT=T$, the *confidence* of an association $\{X_1..X_p..X_L\}$ is defined as the ratio between the temporal support of $\{X\}$ when $TDT=T$ and the number of maximal possible occurrences of the association $\{X\}$. For an inter-association, the maximal possible occurrence of the association is determined by the number of occurrences of the item with the minimal support. Its confidence is defined by Eq.(1). For an intra-association, all items are the same, the maximal possible occurrence of the association is determined by the support of the item and the association length (L), as defined by Eq.(2), where $\langle x \rangle$ indicates the maximal integer which is not larger than x .

$$Conf\{X\}_{TDT=T} = TS\{X\}_{TDT=T} / Min\{TS(X_1), .., TS(X_L)\} \quad (1)$$

$$Conf\{X\}_{TDT=T} = TS\{X\}_{TDT=T} / \langle TS(X_i) / L \rangle, i = \forall [1, L] \quad (2)$$

- The *L-ItemSet* is an aggregation of all *L-ItemAssociations* that each of their temporal support is no less than a given threshold.

2.1 Association Mining Algorithm

Our video mining algorithm consists of the following phases:

1. **Transform Phase.** Given video V , this step transforms V from continuous frames into a sequence dataset D .
2. **L-LItemSet Phase.** In this phase, we mine both intra-associations and inter-associations from D . We first find the L-ItemSet, and then use L-ItemSet and a user-specified threshold to find L-LItemSet. We will iteratively execute this phase until no more non-empty L-LItemSet can be found.

3. **Collection & Pruning Phase.** This phase prunes and selects valuable associations from all *L-LItemSets*.

Since Phase 3 is quite obvious, we focus on Phases 1 and 2. Meanwhile, due to the fact that all items in intra-associations are the same, mining this type of associations is relatively easy, we hereby introduce mechanisms on mining inter-associations only.

In the transform phase, we adopt some video processing techniques to segment a video sequence into shots, and execute shot clustering to explore relationships among shots. To detect video shots, we use our existing algorithm in [Zhu *et al.*, 2003]. For the sake of simplicity, we select the 10th frame of each shot as its key-frame. After the shot segmentation, we adopt a modified split-and-merge clustering algorithm [Horowitz and Pavlidis, 1974] to cluster video shots into groups where visually similar shots are first merged into groups and the groups with large visual variances are split into two clusters. After shot clustering, each shot will receive a class label, we sequentially aggregate the class information of each shot by its original temporal order to form a shot cluster sequence D . As shown in Fig. 2, each icon image denotes one shot and the letter below it indicates its class label, and the acquired shot cluster sequences are given in Fig. 2 (c). Table 2 gives an example of the video association mining, where the first column presents the video shot cluster sequence.

In the L-LItemSet phase, we use the large ItemSet from the previous pass to generate the candidate ItemSet and then measure their temporal support by making a pass over the database D , as shown in Fig. 3. At the end of the pass, the support of each candidate is used to determine the L-LItemSet. The candidate generation is similar to the method in [Agrawal and Srikant, 1995]. It takes the set of all *k-1-ItemAssociations* in L_{k-1} and all their items as input, and works as shown in Fig. 4. Take the 3-LItemSet L_3 in the fourth column of Table 2 as an example. If L_3 is given as the input, we will get the ItemSet shown in the fifth column after the join. After pruning out sequences whose subsequences are not in L_3 , the sequences shown in the sixth column are left. E.g., {ABDC} is pruned out because its subsequence {BDC} is not in L_3 .

- (1) Given user specified TDT and TS thresholds.
- (2) $I_1 = \{1\text{-ItemSet}\}; L_1 = \{1\text{-LItemSet}\}; // \text{Find } \{1\text{-LItemSet}\} \text{ by using } \{1\text{-ItemSet}\} \text{ and the user-specified thresholds.}$
- (3) For ($k = 2; L_{k-1} \neq \emptyset; k++$)
Begin:
 - (4) $I_k = \text{New candidates generated from } L_{k-1} \text{ (see Fig. 5).}$
 - (5) For each *k-ItemAssociation* in I_k , we count its temporal support by considering the user's specification with TDT .
 - (6) $L_k = \text{Candidates in } I_k \text{ with the minimum temporal support.}$**End**

Fig. 3 The L-LItemSet Phase of the mining algorithm

- (1). Join the items of associations in L_{k-1} .
- (2). Insert the join results into I_k
select $\{p.Item_1, .., p.Item_{k-1}, q.Item_{k-1}\}$
from $\{L_{k-1}.p, L_{k-1}.q, p \neq q\}$
where $\{p.Item_1 = q.Item_1 \& .. \& p.Item_{k-2} = q.Item_{k-2}\}$
- (3). Delete any member $x \in I_k$ such that some $\{k-1\text{-ItemAssociation}\}$ of x is not in L_{k-1} .

Fig. 4 Candidate generation

In Fig. 3 and Fig. 4, L_k denotes the *k-LItemSet* and I_k the *k-ItemSet*.

3 Experimental Results and Discussion

Traditional video database systems use video shots as the units to index and manage video data where the visual similarities among shots are used to construct the index structure. Unfortunately, a single shot which is separated from its context has less capability to convey semantics; Moreover, the index considering only visual similarities ignores the temporal information among shots. Consequently, the constructed cluster nodes may contain shots that have considerable variances both in semantics and visual content and hereby do not make much sense to human perception. Accordingly, a semantic video database management framework has been presented [Zhu *et al.*, 2003] where video semantic units (events, scenes or other scenario information) are used to construct a database index. To facilitate this goal, one of the most important tasks is to detect the video semantic units. In this section, our experimental results will demonstrate that the proposed video association mining technique could be used to explore semantic units for the management of video database systems.

To evaluate the ability of video associations in addressing local event and scenario information and figure out the relationship between *TDT* and the mined associations, we set *TDT* with different values T ($T=1, 3, 5, 7, 9$ and ∞) and assess the associations. For each acquired association, we scan the dataset to check whether all items in the association belong to the same scene each time when the association appears. We define the *Scene Coverage* (*SC*) of an association as the ratio between the frequency of the association's items belonging to the same scene and the frequency of the association's appearance. The higher the *SC*, the better the association addresses the scene and event information. On the other side, with an adopted temporal support, the smaller the *TDT*, the less is the number of mined associations. This indicates that the *TDT* also acts as a factor for pruning associations. We hereby define the *Pruning Rate* (*PR*) as the ratio between the number of associations when *TDT* is T ($T=1, 3, 5, 7, 9$) and ∞ . We have performed our experiments with 5 news videos and 20 medical videos (about 700 minutes), and the results are given in Table 2.

Table 1 demonstrates that when the *TDT* increases, the mined associations become worse in addressing the local scenario and event information. One reason for this declination is that the clustering process may cluster semantically unrelated shots into one group, and consequently, when evaluating the *scene coverage*, the items of an association would come from different events. However, with relatively small *TDT* values, this type of errors can be reduced, because items with a small temporal distance would more likely belong to one semantic unit. On the other hand, Table 1 also indicates that the smaller the *TDT*, the less is the number of mined associations, but the better are the mined associations in addressing event and scenario information. Depending on the user's objectives of association mining, a balance between the number of associations and the *SC* is necessary to select a reasonable value for *TDT*.

Table 2. An example of video association mining, where $\{X\}_S^C$ indicates an association. X denotes the items of the association, S and C indicate the temporal support and confidence respectively. The sequential order in the first column is from left to right, top to bottom.

Shot cluster sequence of V	2-ItemSet		3-ItemSet ($TDT=\infty$)	Candidate 4-ItemSet (after join)	4-ItemSet (after pruning)	4-ItemSet ($TDT=\infty$)
	($TDT=1$)	($TDT=\infty$)				
A B C B C D C A C B C A B D A B C D B E	$\{AB\}_4^{1.0}, \{AC\}_3^{0.8}, \{AD\}_1^{0.3}, \{BA\}_2^{0.5},$ $\{BB\}_1^{0.3}, \{BC\}_4^{0.7}, \{BD\}_3^{1.0}, \{CB\}_4^{0.7},$ $\{CC\}_2^{0.7}, \{CD\}_2^{0.7}, \{DB\}_2^{0.7}$	$\{AB\}_4^{1.0}, \{AC\}_3^{0.8}, \{AD\}_3^{1.0}, \{BA\}_3^{0.8},$ $\{BB\}_3^{1.0}, \{BC\}_4^{0.7}, \{BD\}_3^{1.0}, \{CB\}_4^{0.7},$ $\{CC\}_3^{1.0}, \{CD\}_3^{1.0}, \{DB\}_3^{1.0}$	$\{ABC\}_3^{0.8}, \{ABD\}_3^{1.0},$ $\{ACB\}_3^{0.8}, \{ACD\}_3^{1.0},$ $\{BCD\}_3^{1.0}, \{CDB\}_3^{1.0}$	$\{ABCD\}$ $\{ABDC\}$ $\{ACBD\}$ $\{ACDB\}$	$\{ABCD\}$ $\{ACDB\}$	$\{ABCD\}_3^{1.0}$

Table 1. Video association mining results

<i>TDT</i>	1	3	5	7	9	∞
<i>SC</i>	0.893	0.868	0.755	0.623	0.553	0.342
<i>PR</i>	0.264	0.377	0.516	0.658	0.761	1.0

4 Conclusions

To facilitate video database management, we have explored a new research area of video data mining. A video association mining algorithm has been proposed. Given video V , we first transform it from sequential frames to a relational dataset by shot segmentation, clustering, and constructing a shot cluster sequence. The video mining scheme mines sequentially associated video items from this sequence. In addition to using the traditional association measures, we have integrated temporal features among video shots into the video association evaluation. The experimental results have demonstrated the ability of our mined associations in addressing semantic information for video database management.

Acknowledgments

This research has been supported by the U.S. Army Research Laboratory and the U.S. Army Research Office under grant number DAAD19-02-1-0178.

References

- [Agrawal and Srikant, 1994] R. Agrawal and R. Srikant, Fast algorithm for mining association rules. *Proc. of VLDB, 1994*.
- [Agrawal and Srikant, 1995] R. Agrawal and R. Srikant, Mining sequential patterns. *Proc. of 11th ICDE Conference, pp.3-14, 1995*.
- [Han and Kamber, 2000] J. Han and M. Kamber, Data Mining: Concepts and Techniques, *Morgan Kaufmann, 2000*.
- [Horowitz and Pavlidis, 1974] S. Horowitz, T. Pavlidis, Picture segmentation by a directed split-and-merge procedure. *Proc. of Int. Joint Conf. on Pattern Recognition, pp. 424--433, 1974*.
- [Oh and Bandi, 2002] J. Oh and B. Bandi, Multimedia data mining framework for raw video sequence. *Proc. of MDM/KDD, 2002*.
- [Pan and Faloutsos, 2002] J. Pan and C. Faloutsos, GeoPlot: Spatial data mining on video libraries. *Proc. of CIKM, 2002*.
- [Thuraisingham, 2001] B. Thuraisingham, Managing and mining multimedia database. *CRC Press, 2001*.
- [Wu, 1995] Xindong Wu, Knowledge acquisition from databases. *Ablex Publishing Corp., USA, 1995*.
- [Zaiane *et al.*, 1998] O. Zaiane, J. Han, Z. Li, S. Chee and J. Chiang, MultimediaMiner: a system prototype for multimedia data mining. *Proc. of ACM SIGMOD, 1998*.
- [Zhu *et al.*, 2003] X. Zhu, W. Aref, J. Fan, A. Catlin, and A. Elmagarmid, Medical video mining for efficient database indexing, management and access. *Proc. of ICDE, Mar., 2003*.

POSTER PAPERS

MACHINE LEARNING

A Learning Algorithm for Localizing People Based on Wireless Signal Strength that Uses Labeled and Unlabeled Data

Mary Berna, Brennan Sellner, Brad Lisien, Sebastian Thrun, Geoffrey Gordon, Frank Pfenning

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Abstract

This paper summarizes a probabilistic approach for localizing people through the signal strengths of a wireless IEEE 802.11b network. Our approach uses data labeled by ground truth position to learn a probabilistic mapping from locations to wireless signals, represented by piecewise linear Gaussians. It then uses sequences of wireless signal data (without position labels) to acquire motion models of individual people, which further improves the localization accuracy. The approach has been implemented and evaluated in an office environment.

1 Introduction

This paper addresses the problem of localizing people using a wireless network. Various researchers have recently developed techniques using the signal strength of IEEE 802.11b access points for people localization [1; 3; 6; 9; 11]. In doing so, these techniques require measurement data labeled by the position at which the data was acquired.

Our approach builds on this idea, but augments it by a learned probabilistic motion model of people. The data used for training the motion model is acquired as people walk through the environment. We employ the EM algorithm to train the resulting hidden Markov model.

2 Signal Strength Maps From Labeled Data

At the most basic level, our approach acquires a map of the wireless signal strength at different locations in the environment. For that, it requires the availability of signal strength measurements labeled by the true position. Our implementation represents this map by a piecewise linear Gaussian, that follows the (mostly) one-dimensional manifold of the center of each corridor in the building. Figure 1 illustrates the signal strength measurements for one of the access points in our test building. The measurement noise is modeled as a Gaussian, whose covariance is estimated from the data using the maximum likelihood estimator.

3 Probabilistic Localization

The signal strength map is sufficient to coarsely localize people. Following [6], our approach implements a continuous-state hidden Markov model (HMM), in which the person's position is the internal state, and the signal strengths are the measurements. The signal strength map provides the measurement

probabilities of the HMM. The next state transition, or motion model, may be as simple as a probabilistic model of Brownian motion; a more sensible choice will be discussed below.

Our approach uses this HMM to track the location of the person, by calculating an approximate posterior. This is achieved using the well-known Monte Carlo localization algorithm [10], a variant of particle filters [5].

4 Motion Models From Unlabeled Data

The principal limitation of the approach thus far is its reliance on labeled data. Such data is difficult to collect, as it requires an independent means for generating correct position labels when carrying a wireless receiver through a building. Our approach specifically addresses the use of unlabeled data. Such data is acquired as people walk through the building.

This unlabeled data consists of signal measurements *without* a pose estimate. We use this data to learn a motion model of a person. This motion model is realized by mixtures of Gaussians that characterize the relative change of position within a fixed time interval. Each linear sub-piece of the signal strength map possesses its own Gaussian mixture; in this way, the approach can learn location-specific predictions of people's motion. The motivation to use a Gaussian mixture model arises from the fact that people often engage in one out of a small number of different motions (e.g., turning left or right at an intersection; see Figure 2). In our approach, the number of Gaussians is fixed, as is their variance. Our approach learns the means of the Gaussians and the mixture weights as a function of people's location.

Learning from unlabeled data is achieved using the EM algorithm (see [2; 7; 8] for related work). Beginning with an initial motion model, position probabilities are computed based on the signal strength, using the particle filter described above. These position probabilities are then used to calculate a distribution over the mixture components of the Gaussian, and subsequently to calculate the mean and new mixture weights. This methodology is a straightforward application of the EM algorithm, assuming that both a person's pose and the index of the Gaussian mixture are latent variables [4]. In our implementation, we bias the initial motion model (before learning) towards typical motions people might take, such as going forward, turning left or right, and standing in place.

5 Experimental Results

Systematic experimental results were conducted in an office environment equipped with approximately 50 802.11b wireless access points, distributed over an area of size 120 by 22.5

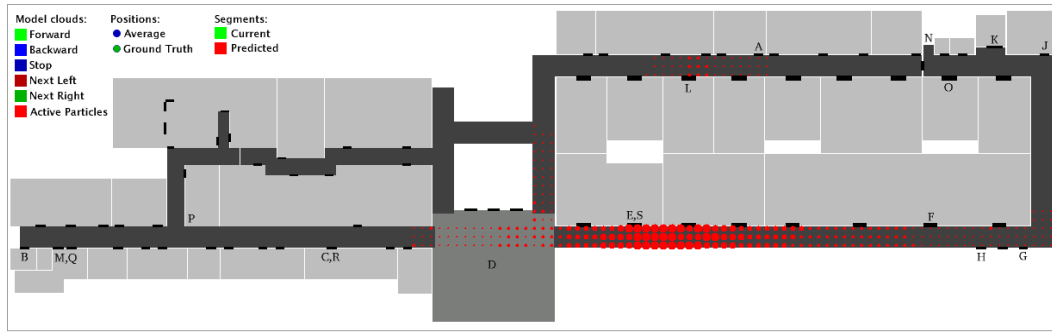


Figure 1: Access point strength as a function of measured location.

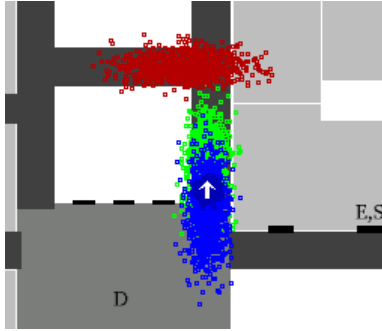


Figure 2: An example motion model. Clearly visible are some of the modes induced by the different actions a person may take.

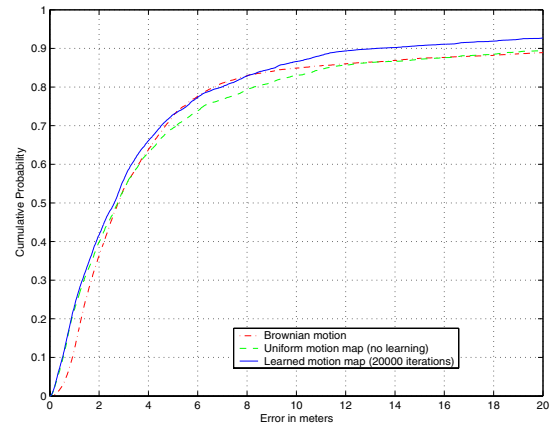


Figure 3: Cumulative probability versus error in meters.

meters. On average, five access points are within communication range of each location in the environment.

All labeled data was acquired by using a laser range finder for localization, similar to work in mobile robotics [10]. A set of labeled data was collected by simply walking about the environment, traversing each corridor at least three times in each direction (along the center, left, and right of the hallway). We also collected a large set of unlabeled training data by traversing paths characteristic of the movements of an individual (all paths originated or terminated at a specific office). Finally, we collected an independent test set with labels.

Figure 3 shows a key result, obtained for 200 particles. This figure shows the cumulative probability function over the average localization error from our labeled test set for different techniques. The top blue curve corresponds to the combined labeled-unlabeled data approach described above, utilizing the learned mixture models. The green dashed curve reflects the performance of the same system with no learning; all mixture weights are uniform. The red dot-dashed curve represents the performance of a pure Brownian motion model, rather than a mixture of Gaussians.

A localization error of 2.25 meters is achieved 70% of the time when only using the labeled data. This number appears to be well in tune with results reported in [6] while requiring a factor of 50 less labeled training data. This error is reduced by 20% as the motion model is trained with the unlabeled data. This illustrates the relative improvement that can be achieved by adding unlabeled measurements to the pool of training data.

References

- [1] P. Bahl and V. N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *INFOCOM* (2), pages 775–784, 2000.
- [2] M. Bennewitz, W. Burgard, and S. Thrun. Using EM to learn motion behaviors of persons with mobile robots. *IROS-02*.
- [3] P.K. Chrysanthis. On indoor position location with wireless LANs. *ISPIRC-02*.
- [4] A.P. Dempster, A.N. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1), 1977.
- [5] A. Doucet, J.F.G. de Freitas, and N.J. Gordon, editors. *Sequential Monte Carlo Methods In Practice*. Springer, 2001.
- [6] A. M. Ladd, K.E. Bekris, G. Marceau, A. Rudys, D. Wallach, and L. Kavraki. Using wireless ethernet for localization. *ISPIRC-02*.
- [7] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Learning to classify text from labeled and unlabeled documents. *Machine Learning*, 39(2/3), 1998.
- [8] B. Shahshahani and D. Landgrebe. The effect of unlabeled samples in reducing the small sample size problem and mitigating the Hughes phenomenon. *IEEE Trans. on Geoscience and Remote Sensing*, 32(5), 1994.
- [9] J. Small, A. Smailagic, and D. Siewiorek. Determining user location for context aware computing through the use of a wireless LAN infrastructure. <http://www.cs.cmu.edu/~aura/docdir/small00.pdf>, 2003.
- [10] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust monte carlo localization for mobile robots. *Artificial Intelligence*, 128(1-2), 2000.
- [11] M. Wallbaum. Wheremops: An indoor geolocation system. *ISPIRC-02*.

Learning to Compete in Heterogeneous Web Search Environments*

Rinat Khoussainov and Nicholas Kushmerick

Department of Computer Science, University College Dublin
Belfield, Dublin 4, Ireland
{rinat, nick}@ucd.ie

1 Introduction

Previous research in Web search has mainly targeted performance of search engines from the user's point of view. Parameters such as precision, recall, and freshness of returned results were optimised. On the other hand, a provider of search services is rather interested in parameters like the number of queries processed versus the amount of resources used to process them. We focus on performance optimisation of search engines from the service provider's point of view.

An important factor that affects the search engine performance is competition with other independently controlled search engines. When there are many engines available, users want to send queries to those that provide the best possible results. Thus, the service offered by one search engine influences queries received by others.

Competition is even more important in *heterogeneous search environments* consisting of many specialised search engines (which provide access to the so-called “deep” or “invisible” Web). Each specialised engine in such environment indexes only a subset of all documents (e.g. on a particular topic). Each user query is only sent to a small number of the engines which can return the best results. Selection between specialised search engines is done (semi)automatically by meta-searchers, which rank engines for each user query. Thus, parameters of specialised engines automatically affect their ranking and, hence, the queries they receive, and ultimately their profit.

We are examining the problem of performance-maximising behaviour for non-cooperative specialised search engines in heterogeneous search environments. In particular, we analyse how specialised search engines can select on which topic(s) to specialise and how many documents to index on that topic. We provide a game-theoretic analysis of a simplified version of the problem and motivate the use of the concept of “*bounded rationality*”. Bounded rationality assumes that decision makers are unable to act optimally in the game-theoretic sense due to incomplete information about the environment and/or limited computational resources. We then cast our problem as a reinforcement learning task, where the goal of a specialised search engine is to exploit sub-optimal behaviour of its competitors to improve own performance.

*This research was funded by Science Foundation Ireland and the US Office of Naval Research.

2 Problem Formalisation

Performance metric. We adopt an economic view on search engine performance. Performance is a difference between the value of the search service provided (income) and the cost of the resources used to provide the service. In our simplified version of the problem we only take into account the cost of resources involved in processing search queries. Under these assumptions, engine performance can be expressed as follows:

$$P = \alpha Q - \beta QD,$$

where Q is the number of queries received in a given time interval, D is the number of documents in the search engine index, α and β are constants.

αQ represents the service value: if the price of processing one search request for a user is α , then αQ would be the total income from service provisioning. βQD represents the cost of processing search requests. If x amount of resources is sufficient to process Q queries, then we need $2x$ to process twice as many queries in the same time. Similarly, twice as many resources are needed to search twice as many documents in the same time. Thus, the amount of resources (and, hence, the cost) is proportional to both Q and D , and so can be expressed as βQD , where β reflects the resource costs. The example of the FAST search engine confirms that our cost function is not that far from reality [Risvik and Michelsen, 2002].

Environment model. Let us assume that users issue queries on just a single topic, e.g. “cars”. We will see later how this extends to multiple topics. It is reasonable to assume that users would like to send queries to search engines that contain the most relevant documents on the topic, and the more of them, the better. Suppose that search engines have “ideal” Web robots which for a given D can find the D most relevant documents on the topic. In this case, the user would simply have to send queries to the engine containing the largest number of documents on the topic.

This model can be extended to multiple topics, if the state of a search engine is represented by the number of documents D_i^t that engine i indexes for each topic t (a query on topic t would be sent to the engine i with the largest D_i^t). Of course, such extension ignores possible overlaps between topics in both queries and documents. On the other hand, if we associate each topic with a search term, the whole engine selection model would closely reflect how some real-life engine selection algorithms work [Callan *et al.*, 1995].

Decision making process. For each time interval, the search engines simultaneously and independently decide on how many documents to index and allocate resources to process expected search queries (thus incurring the corresponding costs). As search engines cannot have unlimited crawling resources, we presume that they can only do incremental adjustments to their index contents that require the same time for all engines. The user queries are allocated to the engines based on their index parameters (D_i^t) as described above.

Since a search engine cannot know the number of queries that users will submit, it may allocate less query processing resources than the number of queries it will receive. We assume that excess queries are discarded and the search engine does not benefit from them. Then, performance of engine i over a given time interval can be represented as

$$P_i = \alpha \min(Q_i, \hat{Q}_i) - \beta \hat{Q}_i D_i,$$

where $\hat{Q}_i = \sum_{t: D_i^t > 0} \hat{Q}_i^t$ is the expected number of queries across all topics for which the search engine is trying to compete (i.e. index at least one document), $D_i = \sum_t D_i^t$ is the total number of indexed documents, $Q_i = \sum_t Q_i^t$ is the total number of queries actually received by engine i , and

$$Q_i^t = \begin{cases} 0 & : \exists j, D_j^t < D_i^t \\ \frac{Q^t}{|K|} & : i \in K, K = \{k : D_k^t = \max_j D_j^t\} \end{cases}$$

is the share of queries in topic t received by engine i , and Q^t is the number of queries users submitted on topic t .

3 Analysis and Solution Approach

The decision-making process can be modelled as a multi-stage game. At each stage, a matrix game is played, where players are search engines, actions are values of (D_i^t), and player i receives payoff P_i (as above). If player i knew the actions of its opponents at a future stage k , it could calculate the optimal response as the one maximising its payoff $P_i(k)$ at that stage. For example, in case of a single topic and a constant query stream it should play $D_i(k) = \max_{j \neq i} D_j(k) + 1$, if $\max_{j \neq i} D_j(k) + 1 < \alpha/\beta$, and $D_i(k) = 0$ otherwise (simply put, outperform opponents by 1 document if profitable, and do not incur any costs otherwise).

In reality, the players do not know the future. One possible way around this would be to agree on (supposedly, mutually beneficial) actions in advance. To avoid deception, players would have to agree on playing a Nash equilibrium of the game, since only then there will be no incentive for them to not follow the agreement. Agreeing to play a Nash equilibrium, however, becomes problematic when the game has multiple such equilibria. Players would be willing to agree on a Nash equilibrium yielding to them the highest (expected) payoffs, but the task of characterising all Nash equilibria of a game is NP-hard even given complete information about the game (as follows from [Conitzer and Sandholm, 2002]).

NP-hardness results and the possibility that players may not have complete information about the game lead to the idea of “bounded rationality”, when players may not use the optimal strategies in the game-theoretic sense. Our proposal is to cast the problem of optimal behaviour in the game as a

learning task, where the player would have to learn a strategy that performs well against its sub-optimal opponents.

Learning in repeated games has been studied extensively in game theory and machine learning. Examples include fictitious play and opponent modelling [Robinson, 1951; Carmel and Markovitch, 1996]. We apply a more recent algorithm from reinforcement learning called GAPS [Peshkin *et al.*, 2000]. In GAPS, the learner plays a parameterised strategy represented by a finite state automaton, where the parameters are the probabilities of actions and state transitions. GAPS implements stochastic gradient ascent in the space of policy parameters. After each learning trial, parameters of the policy are updated by following the payoff gradient.

GAPS has a number of advantages important for our domain. It works in partially observable games. It also scales well to multiple topics by modelling decision-making as a game with factored actions (where action components correspond to topics). The action space in such games is the product of factor spaces for each action component. GAPS, however, allows us to reduce the learning complexity: rather than learning in the product action space, separate GAPS learners can be used for each action component. It has been shown that such distributed learning is equivalent to learning in the product action space. We call a search engine that uses the proposed approach COUGAR, which stands for **C**OMPETITOR **U**SING **G**APS **A**gainst **R**ivals.

4 Preliminary Results

We evaluated our approach in a number of simulation experiments, which demonstrated COUGAR’s ability to compete successfully with different opponents. For the sake of brevity, we present here results from one such experiment with two competing search engines. One search engine was using a fixed strategy, called “Bubble”, the other one was COUGAR. The search engines could increase, decrease (by one), or keep the number of documents indexed on each topic. They could also observe the size of own index, the relative sizes of opponents’ indices, and the number of user queries Q^t submitted for each topic (the last two observations can be obtained from the meta-searcher). The expected number of queries on topic t at future stage k was calculated as $\hat{Q}_i^t(k) = Q^t(k - 1)$.

The experimental setup consisted of three components: the generator of search queries, the metasearcher, and the search engines. The state of a search engine’s document index is represented by a vector (D_i^t) of the numbers of documents indexed by the search engine for each topic. This is the information used by the metasearcher to select search engines.

To simulate user search queries, we used HTTP logs obtained from a Web proxy of a large ISP. We developed extraction rules individually for 47 well-known search engines. The total number of queries extracted was 657,861 collected over a period of 190 days. We associated topics with search terms in the logs. To simulate queries for T topics, we extracted the T most popular terms from the logs. The number of queries generated on topic t for a given day was equal to the number of queries with term t in the logs belonging to this day.

The “Bubble” strategy tries to index as many documents as possible without any regard to what competitors are do-

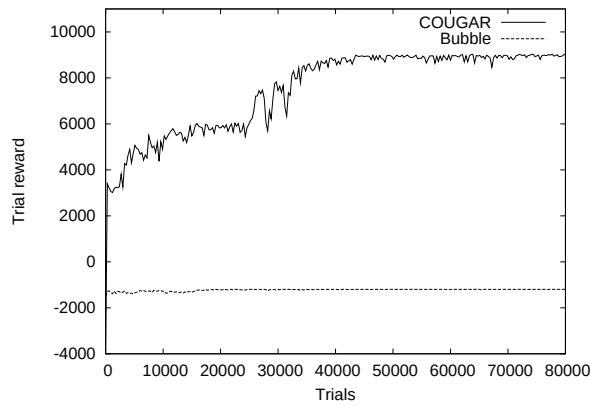


Figure 1: “Bubble” vs COUGAR: learning curve

ing. As follows from our performance formula (Section 2), such unconstrained growing leads eventually to negative performance. Once the total reward falls below a certain threshold, the “Bubble” search engine goes bankrupt (it shrinks its index to 0 documents and retires until the end of the trial). This process imitates the situation, in which a search provider expands its business without paying attention to costs, and eventually runs out of money (or an analogy with the “.com bubble”). An intuitively sensible response to the “Bubble” strategy would be to wait until the bubble “bursts” and then come into the game alone. That is, a competitor should not index anything while the “Bubble” grows and should start indexing a minimal number of documents once the “Bubble” search engine goes bankrupt.

Our simulation had two topics, and “Bubble” was increasing (and decreasing) the number of documents indexed for each topic simultaneously. First, we trained COUGAR in a number of learning trials. Once it reached a steady performance level, the resulting strategy was evaluated in a series of testing trials. Each simulation trial consists of 100 days, where each day corresponds to one stage of the multi-stage game played. The resulting performance in the whole trial is calculated as a sum of discounted rewards from each day.

Figure 1 shows how COUGAR’s performance improving during learning. Figure 2 visualises a testing trial between the “Bubble” and the COUGAR engines by showing the number of documents indexed by the engines on each day of the trial (the top half of Y axis shows the number of documents for topic 1, the bottom half shows the number of documents for topic 2). Note that COUGAR has learned to wait until “Bubble” goes bankrupt, and then to win all queries for both topics.

We also investigated effectiveness of our approach against evolving opponents. In particular, we evaluated performance of a COUGAR strategy learned in self-play against other learners (other COUGARs in our case). We observed that while COUGAR’s performance can be quite stable against equally complex learners, it may still be sub-optimal, in the sense that a “cleverer” learner (e.g. a COUGAR with more policy states) can outperform it.

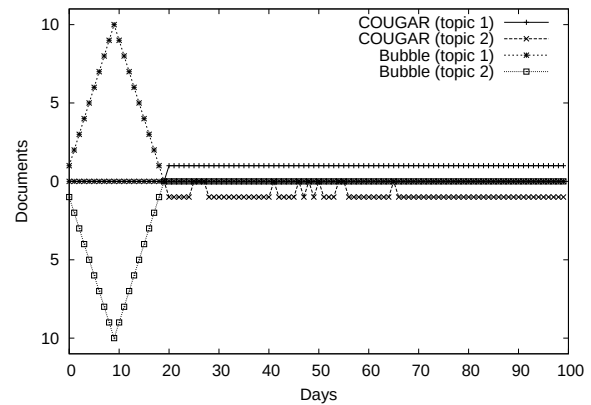


Figure 2: “Bubble” vs COUGAR: sample trial

5 Future Work

We do not claim to provide a complete solution for the problem of performance management in heterogeneous Web search, but COUGAR is a promising first step. Clearly, we have made many strong assumptions in our models. One future direction will be to relax these assumptions to make our simulations more realistic. Another important direction would be to further study performance of the learning algorithm in self-play, as well as against other learners.

While we are motivated by the optimal behaviour for search services over document collections, our approach is applicable in more general scenarios involving services that must weigh the cost of their inventory of objects against the expected inventories of their competitors and the anticipated needs of their customers. For example, it would be interesting to apply our ideas to large retail e-commerce sites which must decide what products to stock.

References

- [Callan *et al.*, 1995] J. P. Callan, Z. Lu, and W. B. Croft. Searching distributed collections with inference networks. In *Proc. of the 18th Annual Intl. ACM SIGIR Conf.*, pages 21–28. ACM Press, July 1995.
- [Carmel and Markovitch, 1996] D. Carmel and S. Markovitch. Learning models of intelligent agents. In *Proc. of the 13th National Conf. on Artificial Intelligence*, pages 62–67, Menlo Park, CA, 1996.
- [Conitzer and Sandholm, 2002] V. Conitzer and T. Sandholm. Complexity results about Nash equilibria. Technical Report CMU-CS-02-135, Carnegie Mellon University, 2002.
- [Peshkin *et al.*, 2000] L. Peshkin, N. Meuleau, K.-E. Kim, and L. Kaelbling. Learning to cooperate via policy search. In *Proc. of the 16th Conf. on Uncertainty in Artificial Intelligence*, pages 489–496. Morgan Kaufmann, 2000.
- [Risvik and Michelsen, 2002] K. Risvik and R. Michelsen. Search engines and web dynamics. *Computer Networks*, 39:289–302, June 2002.
- [Robinson, 1951] J. Robinson. An iterative method of solving a game. *Annals of Mathematics*, 54:296–301, 1951.