3-1-1992

Progressive Horizon Planning - Planning Exploratory-Corrective Behavior

Ron Rymon
University of Pennsylvania

Bonnie L. Webber
University of Pennsylvania, bonnie@inf.ed.ac.uk

John R. Clarke
Medical College of Pennsylvania

University of Pennsylvania Institute for Research in Cognitive Science Technical Report No. IRCS-92-31.

Progressive Horizon Planning - Planning Exploratory-Corrective Behavior

Abstract
Much planning research assumes that the goals for which one plans are known in advance. That is not true of trauma management, which involves both a search for relevant goals and reasoning about how to achieve them.

TraumAID is a consultation system for the diagnosis and treatment of multiple trauma. It has been under development jointly at the University of Pennsylvania and the Medical College of Pennsylvania for the past eight years. TraumAID integrates diagnostic reasoning, planning and action. Its reasoner identifies diagnostic and therapeutic goals appropriate to the physician's knowledge of the patient's state, while its planner advises on beneficial actions to next perform. The physician's lack of complete knowledge of the situation and the time limitations of emergency medicine constrain the ability of any planner to identify what would be the best thing to do. Nevertheless, TraumAID's *Progressive Horizon Planner* has been designed to create a plan for patient care that is in keeping with the standards of managing trauma.

Comments
University of Pennsylvania Institute for Research in Cognitive Science Technical Report No. IRCS-92-31.

The Institute For Research In Cognitive Science

Progressive Horizon Planning - Planning Exploratory-Corrective Behavior

by

Ron Rymon
Bonnie Webber
John R. Clarke

University of Pennsylvania
3401 Walnut Street, Suite 400C
Philadelphia, PA 19104-6228

March 1992

Site of the NSF Science and Technology Center for

Research in Cognitive Science

University of Pennsylvania
Founded by Benjamin Franklin in 1740

IRCS Report 92-31

Progressive Horizon Planning – Planning Exploratory-Corrective Behavior

RON RYMON, BONNIE L. WEBBER AND JOHN R. CLARKE*

Abstract – **Much planning research assumes that the goals for which one plans are known in advance. That is not true of trauma management, which involves both a search for relevant goals and reasoning about how to achieve them.**

TraumAID **is a consultation system for the diagnosis and treatment of multiple trauma. It has been under development jointly at the University of Pennsylvania and the Medical College of Pennsylvania for the past eight years. TraumAID integrates diagnostic reasoning, planning and action. Its reasoner identifies diagnostic and therapeutic goals appropriate to the physician's knowledge of the patient's state, while its planner advises on beneficial actions to next perform. The physician's lack of complete knowledge of the situation and the time limitations of emergency medicine constrain the ability of any planner to identify what would be the best thing to do. Nevertheless, TraumAID's** *Progressive Horizon Planner* **has been designed to create a plan for patient care that is in keeping with the standards of managing trauma.**

I. Introduction

TraumAID is an implemented consultation system, designed to serve as an aid to residents and physicians during the *initial definitive management* phase of trauma care (i.e., after the patient has been stabilized, but the extent of his/her injuries not yet determined or addressed). TraumAID has been under development over the past eight years as a collaboration between the Computer and Information Science department in the University of Pennsylvania and the Department of Surgery in the Medical College of Pennsylvania (MCP). TraumAID currently offers management advice on multiple trauma involving penetrating injuries to the abdomen and chest.

TraumAID was first implemented (TraumAID 1.0) as a purely rule-based system. Following two tests of the quality of its recommendations [3, 4], a PC-based version of TraumAID 1.0 was set up in MCP's Emergency Room for an initial exploration of its clinical possibilities. In a twelve-month period, approximately 100 cases were collected involving penetrating injuries to the chest and/or abdomen. (These were entered into TraumAID 1.0 either in a timely fashion or after treatment was completed.) One thing we discovered was that, in those cases where TraumAID 1.0's advice was insufficiently in keeping with standards of trauma care, it was most often because the diagnostic and therapeutic actions it was recommending, while appropriate for isolated injuries, were inappropriate given the multiple injuries the patient had suffered.

At this point, we turned to Artificial Intelligence planning research for guidance on building a planner that would produce the desired results. We soon discovered, however, that major assumptions of then–current planner stood in conflict with some of trauma management's most salient features. This led us to develop our *Progressive Horizon* planning paradigm as a way of addressing those features.

To understand the differences between TraumAID's planner and the classical ones, it is important to remember that the latter (e.g. Strips, [8]) descend from general problem solvers (e.g. GPS [7]) that themselves originate from theorem provers and search programs. These planning programs inherited many of the assumptions of their predecessors: they viewed planning as an independent process that takes a goal and an initial world state as its input and produces a sequence of actions (plan), as its output. Within that framework, actions are viewed much like operators – well-defined transformations on a space of states. Even though subsequent state-space planners (e.g. Noah [17], Sipe [22]) have greatly enhanced capabilities and richer

*Manuscript received ....; revised .... Rymon and Webber are with the Department of Computer and Information Science, University of Pennsylvania, Philadelphia PA, 19104-6389. Clarke is with the Department of Surgery, Medical College of Pennsylvania, Philadelphia PA, 19129. This work has been supported by the Army Research Organization under grant DAAL03-89-C0031PRI, by the National Library of Medicine under grant 1-RO1-LM-05217-01 and by the Agency for Health Care Policy and Research under grant 1-R01-HS-06740-02. For correspondence please use electronic mail: rymon@linc.cis.upenn.edu, bonnie@cis.upenn.edu, and jclarke@grad1.cis.upenn.edu.

representations of actions ([23]), they still have a rather rigid view of the planning problem itself.

A central theme in recent research concerns a planner's ability to cope with a highly dynamic environment. Since planning is computationally costly [2], real-time planning runs the risk that by the time it is completed, some of its assumptions may no longer be valid. Furthermore, even if planning is instantaneous, unforeseen changes in the world, unpredictable effects of actions, and new information acquired *while* the plan is executed may also invalidate it. These have been the main arguments for reactive (or reaction) planning paradigms [1, 9, 18].

A common thread in reactive planning research is the observation that an agent must be sensitive to its environment, i.e. that an agent's plans must be adaptable to changes in its environment. The same holds in trauma care – throughout the course of treatment, management plans have to adapt to changing patient condition, changing information about that condition, change in resources, etc. Plans in both thus evolve over time, through *cycles* of sensing, followed possibly by reasoning and planning, and then activity and further sensing.

However, most planning research (reactive as well as classical) still assumes that the goals for which one plans are *given* to the agent. This is not so in trauma management: goals must be determined through both reasoning and activity. In doing so, *diagnostic* activity – concerned with determining therapeutic goals – must often be interleaved with *therapeutic* activity – concerned with addressing therapeutic goals.

In trauma management, patients often present with multiple injuries or multiple problems arising from the same injury. Such problems often interact in a variety of ways (e.g. causally, physically etc.). This interaction also extends to the procedures used to address them. In trauma management, many problems can be diagnosed, or treated, through a variety of alternative procedures. While for an isolated problem, there often is a procedure that is preferable to others, the best management of multiple problems is often a function of the *combination* of problems and not a sum of the single treatments.

One other characteristic of trauma management is reflected in TraumAID's planner. Trauma management is cautious insofar as actions cannot be assumed to have their intended effects. After performing an action, others must be planned to determine its effects before conclusions can be drawn. For example, when a chest tube is inserted into the thoracic cavity to relieve pressure, first its output is visually monitored, then its proper placement is confirmed through a dedicated X-ray, and finally the achievement of its intended goal – the patient re-stabilizing – is verified. This need for verification thus leads to additional goals that must be satisfied in subsequent cycles of reasoning, planning and activity.

In Section II., we begin by arguing that in domains such as trauma management, diagnostic reasoning and planning must be integrated. Section III. describes the particular architecture of TraumAID.

Focusing on planning aspects of TraumAID, Section IV. discusses, in general terms, the *Progressive Horizon* planning paradigm which we argue to be an acceptable, computationally feasible compromise between classical and reactive planning. Sections V., and VI. describe the particular implementation of a progressive horizon planner in TraumAID.

II. The Interplay of Diagnosis and Therapy

In many domains it is common for decisions regarding the *treatment* of problems to reflect those concerned with their *diagnosis*. Artificial Intelligence research is clearly divided into separate *diagnosis* and (therapy) *planning* subfields: diagnostic research ignores the corrective actions that follow it, while planning research isolates itself from the reasoning and activity required to determine its goals and figure their achievement.

Trauma management requires both diagnosis and treatment, but its particular features suggest that the two need be strongly coupled:

1. **Diagnosis and treatment may have to be temporally interleaved.**

   Given a patient who suffers multiple injuries, at any given point during his/her management, some treatable conditions may have already been diagnosed, while other hypotheses await further attention. In many cases, therapy for the former cannot be delayed until resolution of all the latter.

   Facing multiple diagnostic and therapeutic goals, an agent has to choose which to address next. In trauma, a few rules of thumb are used to sort goals by their urgency and importance. Planning considerations are also important in determining the order in which goals are attempted (cf. [5]).

   Note that taking action before a case is completely characterized implies considerable uncertainty in planning and in execution.

2. **Diagnosis may require activity.**

   Diagnosing each injury may require information whose acquisition requires action. Such activity has associated costs (time, money, risk to the patient etc.).

Furthermore, it may sometimes result not only in enhanced knowledge, but also in actual change to the patient state (consider, for example, invasive tests). Most importantly maybe, it may interact with other needed activity, and thus has to be planned.

As a therapy-driven domain, trauma management emphasizes efficiency in its diagnostic efforts. Diagnostic activity should thus be limited to issues that can affect decisions concerned with treatment, even if that results in diagnostic "incompleteness" [15].

3. **Activity may require diagnosis.**

   Actions performed on a patient require diagnostic reasoning before, during, and after their execution. Before an action is planned, reasoning is required to determine its goals and check its validity (e.g. in terms of preconditions). When an action is taken, its unpredictability means that its execution must be monitored and that changing conditions and emerging goals must be identified. After an action is performed, its *actual* effects must be determined, as well as whether or not its goals have been satisfied.

   Note that, in turn, such diagnostic reasoning may require activity and so on and so forth.

III. INTEGRATING DIAGNOSIS AND ACTION

A. ARCHITECTURAL OVERVIEW

In light of the above observations, TraumAID's architecture integrates diagnostic reasoning, planning and activity. Its knowledge and reasoning are factored into two components:

- a goal-directed, *diagnostic reasoner* links clinical *evidence* through *conclusions* to two types of *goals*: (1) *diagnostic goals*, to help proving, or dismissing, suspected hypotheses through diagnostic activity, and (2) *therapeutic goals*, to address concluded diagnoses through corrective procedures;

- a *planner* constructs plans to address *combinations* of diagnostic and therapeutic goals proposed by the reasoner, recommending which of those goals to address at each stage of patient management and how best to address them.

For execution and reporting of evidence, TraumAID relies on the physician and nursing staff.

A typical management session runs through a number of cycles in which evidence is linked to goals to a plan of action, and again. Figure 1 depicts a basic cycle:

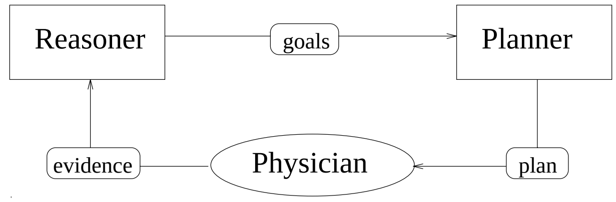

Figure 1: A Cycle of Reasoning Planning and Action

A cycle begins as initial information is provided by the physician to the reasoner. From this, the reasoner draws some initial conclusions and suggests preliminary goals of action. Initially, such goals are more likely to be diagnostic in nature, exploring potential problems.

Given what is already known, TraumAID's planner will construct a plan to address those goals – a plan that reflects both the importance ascribed to the different goals and standard practices of trauma care. The particular method it uses is the subject of the next few sections. TraumAID offers this plan as advice to the physician, and any new information subsequently received initiates a new cycle of reasoning, planning and action. (This information may or may not be related to any particular action in the plan, cf. Section C.)

In each cycle, conclusions regarding the patient's condition, achievement of goals by past actions, and the appropriateness of old and new goals are continuously reassessed. A need to adapt the system's *current* plan may be dictated by change in the current set of goals or by a change in the patient's condition (or in one's knowledge of that condition) that affects the choice of actions for those goals.

As further cycles are carried out, the system's understanding of the patient's injuries is refined and its plans become more therapeutic and less diagnostic.

B. DIAGNOSTIC REASONING

The system's diagnostic reasoning follows the goal-directed diagnostic (GDD) paradigm described in detail in [15]. Briefly, the paradigm uses two reasoning schemas in forward-chaining:

1. **Evidential rules** map evidence and lower level conclusions to new conclusions. For example, the following rule concludes that a patient's shock is due to abdominal bleeding:

   Shock ∧
   false(Single_Wound_to_Upper_Chest) ∧
   unless(Pericardial_Tamponade) ∧
   unless(Massive_Hemothorax) ∧
   unless(Tension_Pneumothorax)
   ⇒ *Shock_of_Abdominal_Origin*

2. **Goal-setting rules** map evidence and conclusions to goals (either diagnostic or therapeutic). For example, the following rule concludes that it is relevant to know whether or not the patient has hematuria:

   Wound(Type='Gunshot)) $\wedge$
   Bullet_in_Abdomen) $\triangleright$ *Hematuria*

The most salient property of GDD is that by focusing on appropriate *therapy*, it avoids instantiating diagnostic goals whose satisfaction cannot affect decisions concerning treatment.

C. The Physician

Physician and nursing staff cooperation is critical to TraumAID's success. To this end, a HyperCard-based interface to TraumAID is being developed in cooperation with emergency department nurses at the Medical College of Pennsylvania. This interface is meant to replace a paper-based system of data collection (the "trauma flow sheet") that has been in use at MCP for the past four years.

(A sample data entry card from the interface is shown in Figure 2.) Since the interface provides additional useful information (e.g., automatic updates of the patient's Glasgow coma score, constantly updated probability of survival, accumulated cost of care, etc.) and legible records, we believe that it will be an effective addition to the practice of emergency medical care. In addition to this interface development, we are also developing a critiquing facility for TraumAID [14, 20] as an alternative to actively proffered advice. A critiquing system operates less obtrusively, only offering comments when there are significant differences between its current view of appropriate management options and those expressed by the physician in his/her "orders".

From a planning perspective, it is very important that the system accommodates *whatever* action is taken by the physician and *whatever* information is reported. TraumAID's modularity allows it to quickly adapt itself to any change in the state of affairs, whether or not it has advocated, or even anticipated it. We have also determined that it is important that plans, presented to the physician throughout the management, be as complete as possible, addressing *all* known problems. Such a *global* view of the current situation is, in the opinion of our domain expert, important to gain physicians' cooperation.

D. Planning

TraumAID's planner is the main subject of this paper. In the next three sections, we focus on the *Progressive Horizon* planning paradigm which we have developed for TraumAID. Section IV. describes the paradigm in general terms, independently of its particular implementation in TraumAID. Sections V., and VI. detail TraumAID's progressive horizon planner.

IV. Progressive Horizon Planning

A. Overview

TraumAID's planning paradigm, which we call "Progressive Horizon Planning", belongs to a broader class of *partial* planning paradigms. Plans can be partial in many ways: plans that rely on unproven (default) assumptions are partial [6, 11], reaction plans which specify only the next action are partial [18], the Real-Time A^* algorithm [13] is a planning algorithm which performs a partial search, and plans that are verified only in a high, non-operational level of abstraction are also partial [16, 19]. The particular features of multiple trauma management call for somewhat different partial plans, which we call *partial-global* plans, and progressive horizon planning is a way of producing them. Trauma management plans must be global in that they must always acknowledge *all* known goals. As was pointed out in section C., this is necessary to gain a vital cooperation between the physician and the system. However, these plans may also have to be partial since not all goals may be known when a plan is constructed: in fact, part of the plan is dedicated to exploring potential goals.

Progressive horizon planning works within a cycling architecture, such as the one shown in Figure 1, where it is strongly coupled to reasoning and action. During a single consultation session, planning consists of a number of cycles in which its product, the management plan, is continuously reassessed and adapted to new information. Initial plans are typically exploratory in nature, acquiring diagnostic information. Such information will then point, through diagnostic reasoning, to therapeutic goals and will facilitate planning to address them. Later plans will thus become increasingly therapeutic.

Since planning is computationally intensive and since a progressive horizon planner is often presented with only a partial set of goals, and also since unpredictability may anyhow invalidate *any* plan, no matter how well its goals were specified and how much computation is put into it, a progressive horizon planner employs a *partial* planning scheme.

In each cycle, instead of seeking an optimal plan (which is a chimera, given any uncertainty in knowledge and unpredictability in the results of action), a progressive horizon planner proceeds in two steps:

- First, it constructs an *approximate* plan (also referred to as plan *sketch*, cf. Section V.);

Figure 2: Initial Data Entry Frame - HyperCard Interface to TraumAID

- Then, it optimizes this sketch using general, as well as domain specific optimizers. What is important about this optimization is that it is only applied to the first few actions in the sketch – those within the planner's horizon. Plans produced by a progressive horizon planner are thus partial in one more way.

In what follows, we assume that a plan sketch can be constructed via a quick-and-dirty process, such as TraumAID's current algorithm (Cf. Section V.).

As suggested by its name, we do not expect the approximate planning stage to come up with an optimal plan. Since planning is intractable even in very restrictive forms [2], we do not expect a resource-bounded optimization to hit a global optimum either. Recall, however, that even if the computational resources were unbounded, uncertainty and unpredictability may anyhow obsolete *any* plan, during, or even prior to its execution.

To strike a balance between the competence of a plan, and the time it takes to produce it, we note that the chance for an action to be affected by some unpredicted event increases with its position in the plan. Similarly, knowledge acquired through earlier actions may obsolete the need for a later one, or affect decisions concerning the choice of means for addressing less urgent goals. For that reason, it seems plausible for a progressive horizon planner to focus its attention on the first few actions of the sketched plan.

B. Planning as Search in a Situation-Action Space

One way to view planning is as a type of search process. There are several ways to do this: Classical planners viewed it as searching a space of plans (e.g. [8], [17]). In a plan-space, states represent plan structures and transitions correspond to operations on those structures (e.g. add or remove action, impose or change order, etc.) Alternatively, planning can also be viewed as search through a space of world states. States in this space are world descriptors, connected to each other with directed edges, each corresponding to an operator that transforms one world state into another. In the context of planning, actions constitute a particularly interesting class of operators, but there are clearly other operators (e.g. *information-is-received*, or *blood-pressure-is-increasing*) in which the agent is *not* engaged in activity, but the state of the world and/or the agent's knowledge about that state do change.

Ideally, each state in this *Situation-Action* (SA) space would consist of a *complete* description of the corresponding world state, or at least all relevant information about it. A planner would then be given an *initial world state*, a sufficient description of what should hold in a *goal state* and a set of *operators*, with a complete description of their preconditions and effects. Its task would be to construct a sequence of actions (a path in the SA-space) that when followed from the initial state would reach one of possibly several goal states. In the context of this ideal SA-space, planning can be viewed as simply searching a (possibly infinite) directed tree, rooted at the initial situation. We call this tree the *Situation-Action tree* (SA-tree). Figure 3 depicts a simple SA-tree.

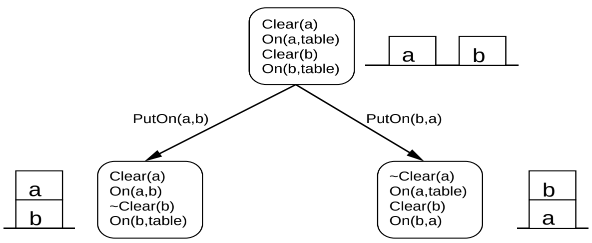

Figure 3: Situation-Action Tree.

Obviously, planning by simply searching some representation of the SA-space is not practical. Nevertheless, because the SA-space so closely resembles a *simulation* of reality, plans that were originally constructed using other representations may easily be transformed into the SA-space, making it attractive as common ground for evaluating artificial (and also human) planning techniques.

One way in which SA-space resembles a simulation of reality lies in how it captures *uncertainty* – the lack of knowledge – and *unpredictability* – the impossibility to predict the actual result of actions, or unforeseen changes in the patient state. (We find this a useful distinction in planning because of the different means by which the two can be addressed: uncertainty is amenable to exploratory activity embedded within a plan; unpredictability may only be monitored for, or detected *after* the fact.)

In SA-space, unpredictability means that a given action performed at a given state may result in one of a *number* of possible states, each corresponding to a different outcome. Search-wise the branching factor is increased. Uncertainty means that one may not know what state one is, so one may have to expand the plan to include activity of purely exploratory nature. (for example, figure 4 shows the effect of uncertainty on the example of figure 3. Essentially, if the agent does not *know* whether a is a block or a pyramid, it has one of two options: either go ahead with a plan that does not verify a's nature, or expand the plan with a preceding *diagnostic* action). Note that if not for the uncertainty, these actions could be eliminated without *any* effect on the plan's correctness. Uncertainty thus contributes to the *length* of the plan. In the context of a simple brute force search through the SA-space, the length of the plan corresponds to the *depth* of the search process.

C. Computational Analysis

The SA-tree concept can be used to understand progressive horizon planning. Consider Figure 5 where the optimal

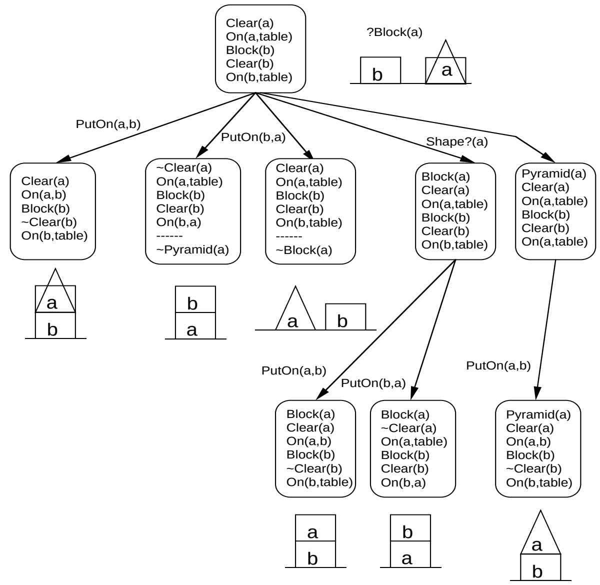

Figure 4: SA-tree with Uncertainty

plan (P) is boldfaced. In the first planning cycle, a progressive horizon planner searches the SA-tree that is rooted in the state which corresponds to the initial world setting. Then, each subsequent cycle searches a sub-tree rooted at the end of the path from that initial root and which edges are marked with the actual actions taken by the physician (and their results). Thus, the planning horizon, even if taken to be constant, progresses from one cycle to the next – hence the name *progressive horizon*.

In each cycle, a progressive horizon planner first constructs a plan sketch (denoted P' in Figure 5). It then optimizes that plan's first few actions, re-sketching a completion if necessary. In the SA-tree, the above process corresponds roughly to choosing among alternative plan *initiations*. Each such initiation is of a small size (determined by the *horizon*; C_{opt} in Figure 5) and is evaluated together with its own sketched completion.

The SA-space representation is also useful in comparing the progressive horizon idea to other planning paradigms: for example, classical planners carry out an exhaustive search in the plan-space. The analogous SA-space planner will thus plan by searching the complete SA-space. Such an exponential enterprise is very costly, particularly given uncertainty and unpredictability. At the other extreme, reactive planners [18, 9, 1] try to remedy this deficiency of classical planners by minimizing run time consideration of alternative plans. However, while these planners differ in the way in which situations are indexed and identified and in their plans format, all of them must represent, one way or another, *all* anticipated situations [10]. Their SA-space analog will have to represent a reaction for each of the

SA-tree nodes.

In search problems, the use of horizon is not new by itself. Virtually all game-playing programs traverse a search tree to a horizon, using an evaluation function to evaluate the quality of the hidden part of the tree. Korf's RTA* [13] applies that idea to planning: searching a structure which resembles the SA-tree, it cuts search at a predetermined level, replacing further search with an evaluation function. In terms of their SA-space analogs, the progressive horizon planner differs from its RTA* counterpart in that (a) its SA-space may not be deterministic (i.e. may allow multiple outcomes for a given action and state), and (b) although exhaustive search is limited by the horizon, a *complete* plan is nevertheless constructed.

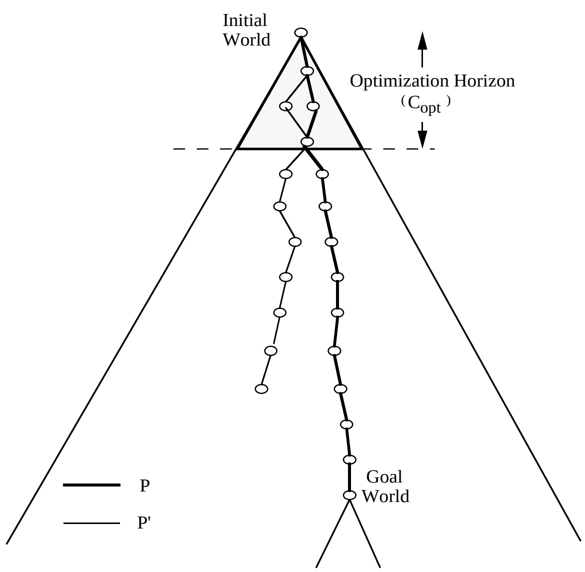

Figure 5: Optimizing to a Horizon in an SA-tree

To summarize, let a be the SA-tree branching factor, n be the size of an optimal plan, then

1. A classical, complete, run-time search requires $O(a^n)$ *time*.

2. Reactive planning techniques that address all anticipated situations may require *space* (and therefore preprocessing run time) of $O(a^n)$.

3. Under *RTA**, one iteratively searches to a given horizon, applies an evaluation function to the leaves, and propagates the results back up the tree. Let C_{opt} denote the search horizon and $f(a, n)$ denote the cost of evaluating a leaf at level n. Then the total cost of an *RTA** iteration is $O(a^{C_{opt}} \cdot f(a, n))$.

4. Under the progressive horizon paradigm, the complexity of each planning cycle is given by the sum of:

   (a) A term, $x(a, n)$, accounting for the cost of constructing an approximate plan (recall that we assume that this can be done fairly quickly).

   (b) A term, $O(y(a, n) \cdot a^{C_{opt}})$, accounting for the optimization phase. Here, the $y(a, n)$ term represents the time required to generate and evaluate a single plan with a given initiation and a corresponding sketched completion.

   Each planning cycle thus has a total run time of $O(x(a, n) + y(a, n) \cdot a^{C_{opt}})$. Assuming that $x(a, n)$, and $y(a, n)$ are both polynomial in a and n, then since C_{opt} is constantly fixed (or at least bounded by a constant) the total run time is also polynomial in a and n.

V. PLAN SKETCHING AS CHOICE AND ORDERING

Here we discuss the first step of progressive horizon planning, as embodied in TraumAID - the development of a plan sketch. We first provide a general analysis of the plan sketching task and then outline the actual algorithm used by the system.

In general, TraumAID's planner considers multiple diagnostic and therapeutic goals, deciding *how* to address them and *in what order*. Functionally, this task is divided into two sub-tasks:

1. Choosing an appropriate set of procedures that address all known goals;

2. Ordering that set in accordance with a given set of constraints.

This is an attractive division because each of the tasks can be mapped into a well-studied problem domain. However, since an effective choice of procedures often depends on possible orderings, one cannot simply run the two processes in sequence. TraumAID's planning algorithm interleaves the two. In general, the procedure choice subtask can be shown to be a generalized form of the known *Hitting-Set* problem. Section A. presents goal-procedure mappings used by TraumAID to represent its own procedure choice problem. Section B. characterizes the features of the scheduling sub-task in TraumAID. Finally, section C. depicts TraumAID's interleaved choice and ordering plan sketching algorithm.

A. Procedure Choice in TraumAID

In domains such as trauma management, problems can often be addressed by several alternative methods. Each method has its own positive and negative features which make it more or less preferable in a given situation. Procedures may also be applicable to more than one condition. For example, immobilizing the patient is part of an appropriate response to many different fractures.

When confronted with *multiple* goals, the choice of an appropriate set of procedures becomes more tricky. Since some procedures may address more than a single goal, one might take advantage of the situation by choosing procedures of wider coverage. Viewed this way, the choice of procedures instantiates the well-studied Hitting-Set problem. Unfortunately, that problem is NP-Hard [12].

TraumAID's knowledge of the relationship between goals and the actions that it plans is specified in two sets of mappings: *goal-procedure* mappings which map goals (diagnostic and therapeutic) into a set of one or more alternative procedures, and *procedure-action* mappings that map each procedure into a sequence of actions that satisfies it – i.e., its sub-actions. For example, the following goal-procedure mapping expresses the fact that the diagnostic goal of ruling out abdominal bleeding can be satisfied by either a *peritoneal lavage* or a *CT scan* of the abdomen.

RO_Abdominal_Bleeding :
  Get_Peritoneal_Lavage,
  Get_CT_Scan_Abdomen.

Other goal-procedure mappings show that both peritoneal lavage and the abdominal CT scan can be used for other diagnostic goals such as ruling out a *suspicious abdominal wall injury* (lavage) or ruling out *renal injury* (CT scan).

The following example of a procedure-action mapping specifies the steps in a standard treatment for simple pneumothorax:

Std_Care_Simple_Pneumothorax(Side=S) :
  Primary_Tube_Thoracostomy(Side=S),
  Primary_Tube_Thoracostomy_Report(Side=S),
  Post_Primary_Chest_Tube_X-Ray.

The use of procedure-action mappings is discussed further in the next section.

While a goal-procedure mapping may identify several procedures as being applicable to satisfying a given goal, they may not be equally preferable in all cases. Procedures are ordered by their respective preference, assuming that the goal at hand is the *only one* posed by the reasoner. The presence of other goals may alter this preferential order. So, for example, given both the goal of diagnosing abdominal bleeding *and* the goal of ruling out a renal injury, it may be worth adopting the second-best option, the procedure *abdominal CT scan* which appears second in both goal-procedure mappings might better be chosen, since it addresses both goals.

RO_Renal_Injury :
  Get_CT_Scan_Abdomen,
  Get_IVP.

In determining whether one combination of procedures should be preferred over another, TraumAID uses an additional cost function. Currently, this is a single measure, vaguely tagged *cost*, which combines factors such as risk to the patient, likelihood of success, time, and dollar cost. Co-author Clarke is now developing an alternative measure based on *disutilities* – costs (in the above terms) mapped onto the complete space of possible outcomes.

In practice, procedure choice is affected by the resource configuration of the particular hospital and by subjective preferences of the attending physician. Some of these constraints can be supported by the current paradigm. For example, physician preference with respect to procedures can be encoded by their order in a given goal-procedure mapping. Similarly, procedures that require equipment that is not available can be so marked. More complicated variations of resource availability are the subject of research within our group.

B. Procedure Scheduling in TraumAID

At least in theory, once a set of procedures has been chosen, the planning problem becomes one of recommending an appropriate execution order. In trauma management, principles of trauma care and logistic considerations provide a guideline as to what procedures should precede others. TraumAID's plan sketching algorithm uses the following:

1. *Urgency*. Patients may arrive in an unstable condition or become unstable while under care. Instability and its sources must be addressed immediately, or more precisely within a time frame determined by its particular nature.

2. *Logistics*. Areas within a hospital differ in their resources and thus in their ability to support different procedures. Thus a patient may have to be transferred to another site in order to undergo some procedure, such as an X-ray study or an operation. However, since a transition also has costs associated with it (mostly in terms of time), it is common to delay procedures of higher priority, in order to minimize such transfers.

3. *Standardized priorities*. The ABCs of trauma care (i.e. Airway, Breathing, Circulation) specify standard priorities in the treatment of multiple problems. They call for the treatment of airway problems first, then those concerned with breathing, then circulation etc.

Urgency and priority are associated with goals, while logistics are associated with procedures. The result is that chosen procedures can be partitioned into disjoint equivalence classes, each of which composed of procedures with same urgency, site of execution, and priority (see Figure 6). Obviously, such partitioning reduces significantly the number of possible orderings of procedures. It results in a partial (also layered) order on procedures in which procedures of same equivalence class (partition) are unordered with respect to one another.

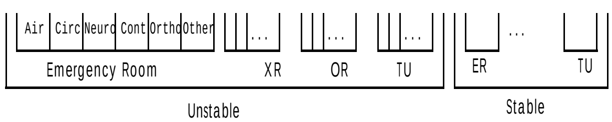

Figure 6: Sorting Procedures

TraumAID 2.0 creates this partition by first ordering goals by urgency and priority and then associating a procedure with the highest ordered goal. Since procedures can often be scheduled in more than one site, TraumAID 2.0 makes use of a preference order on sites (i.e. Emergency Room (ER), X-Ray Unit (XR), Operating Room (OR), Trauma Unit (TU) - cf. Figure 6) and chooses the first site compatible with other constraints.

Urgent goals must be addressed within a certain time frame. TraumAID 2.0's ordering, as so far specified, cannot guarantee that. Although within a given site, procedures are ordered by the urgency of their respective goals, it is often possible that an urgent procedure must take place in the operating room (OR), and so will be scheduled after all emergency room (ER) procedures. This is fine if the performance of those procedures does not exceed the time limitation of the urgent procedure, but may subject the patient to severe risk otherwise. To avoid that, TraumAID 2.0 considers the time factor of each procedure, rejecting lengthy procedures if they would greatly delay urgent ones.

Now it is not procedures themselves that are scheduled but rather the actions that comprise them, as specified in their *procedure-action* mappings. When a procedure is chosen for scheduling, its actions will keep their respective order in the plan, but other actions may be interleaved between them. Scheduling actions, rather than procedures, also gives rise to new opportunities in the *choice* of procedures. It often happens, for example, that procedures share sub-actions. In such cases, it is sometimes possible to reduce the overall cost of a plan by *merging* the procedures' actions, rather than simply carrying out one procedure after the other. For example, if a patient suffers a pneumothorax on *both* sides, it is possible to wait with the post-tube X-ray until *both* chest tubes are inserted and their clinical results reported.

On the other hand, shared actions also represent a difficulty when trying to satisfy the domain constraints. If an action belongs to more than a single procedure, for example, it is not always easy to determine what role it plays, or how much of the characteristics of which procedure it inherits. Moreover, if an action is part of two procedures that disagree on any of the three characteristics, it may be impossible to schedule the action so as to keep the intra-procedural order.

Consider, for instance, a patient suffering both a hemothorax and a pneumothorax. A single chest tube can often serve to address both problems in such patients. Note however that the standard treatment of the hemothorax condition:

Std_Care_Simple_Hemothorax(Side=S) :
Antibiotics,
Setup_Auto_Tranfusion(Side=S),
Primary_Tube_Thoracostomy(Side=S),
Primary_Tube_Thoracostomy_Report(Side=S),
Post_Primary_Chest_Tube_X-Ray.

requires the administration of antibiotics and the setup of an auto tranfusion device prior to the chest tube insertion. Considering the two procedures, one must be able to equate the respective chest tubes, or otherwise *two* tubes will be called for. On the other hand, one cannot respect both the internal order within the hemothorax procedure and the rule by which a pneumothorax must be attended prior to the hemothorax.

TraumAID partially solves these conflicts by (a) allowing actions to carry their own characteristics that distinguish them From the procedures they participate in, and (b) allowing special scheduling information to be associated with an action. For example, we may only require an action to satisfy a subset of the characteristic-based constraints. In the above hemopneumothorax example, special scheduling information conveys that only intra-procedural order is important for the auto tranfusion setup action, while the antibiotics administration can be ordered anytime. TraumAID will thus recommend following the hemothorax procedure, with the last three steps aimed at both the hemothorax and the pneumothorax. Similarly, for patients presenting with a combination of pneumothoraces and hemothoraces on both sides, a single post-tube X-ray will be recommended.

C. TraumAID's Planning Algorithm

In TraumAID's planning algorithm, a plan is incrementally constructed by interleaved choice and ordering:

1. The complete set of diagnostic and therapeutic goals Γ is first sorted based on goal urgency and priority. Recall that urgency is related to shock or the cause of shock, and indicates the need to address this goal promptly. Priority reflects standard practices in trauma care that call for addressing airway problem first, then those concerned with blood circulation, etc.

2. A plan Π is constructed through the following iterated steps, stopping when Γ is exhausted:

   (a) Pick the next goal γ on Γ. If γ is not already addressed by a procedure included in Π (recall that procedures can satisfy more than one goal), identify the most preferred procedure π for addressing γ that does not require unavailable equipment – i.e., identify the procedure that would have been chosen for that patient, were γ the only problem to be addressed.

   (b) Add π's actions to Π, imposing the following ordering constraints:

      i. Actions must keep their respective order within π.

      ii. Unless indicated otherwise in their scheduling information, actions to be performed in a prior site are ordered prior to actions of a subsequent one.

      iii. Within the same site, unless indicated otherwise, order actions addressing goals of higher urgency prior to actions of lesser urgency.

      iv. Within the same site, unless indicated otherwise, order actions of same urgency so that actions of higher priority are ordered prior to actions of lesser priority.

      v. If an action requires more time than is allowed by the urgency of a given goal, order those actions addressing the urgent goal prior to the time-consuming one.

   (c) Check that π does not violate any procedure already in the plan (this is detected by checking for cycles in the partial order resulting from the above constraints). If it does, try to locate π's actions in their next best site. If that does not work, choose choose the next best procedure that addresses γ and repeat.

   (d) If there is no valid way of addressing γ in the current plan, leave it unaddressed and inform the physician. Note that having ordered goals in Step 1 by urgency and priority, any goals left unaddressed will be less urgent and less important (vis-a-vis standard practices of trauma care) than any goal already addressed by the plan.

VI. Plan Optimization in TraumAID

The planning algorithm just described is clearly a greedy one. As such it cannot guarantee an optimal plan. To improve the quality of its plans, TraumAID takes a progressive horizon approach. In each cycle, before a plan is recommended, a 1-depth optimization is carried out in which the potential replacement of the first procedure on the plan with an alternative option is checked out. The effect of each such modification on the entire plan is evaluated.

To date, we have identified two general situations in which the approximate algorithm can be led to wrong coverage decisions. Consider Figure 7 in which goals are indicated by circles and procedures by rectangles. The numbers on the arcs represent order of preference. For simplicity, assume that all procedures are equal in terms of their risk, cost etc. In both situations, the approximate planning stage of TraumAID will select procedure 1 for addressing goal **A** and procedure **2** for goal **B**. However, a quick glance at the first example discovers that procedures **1** and **2** can be both replaced with procedure **3**. Similarly, in the second example, procedure **2** can serve both purposes. To address these shortcomings, we currently employ a general optimizer that considers all possible replacements for the optimized procedure.

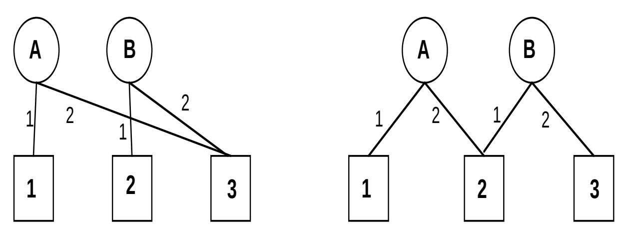

Figure 7: Weaknesses in Approximate Algorithm

We are also looking at several domain specific optimizers that will encode some domain knowledge concerned with the advantages of using certain procedures in conjunction with others. For example, one may choose to replace a usually preferred lavage with an alternative CT scan to rule our liver injury so as to take advantage of a intravascular contrast solution that has already been administered for an arteriogram.

Currently, both the general and the domain specific optimizers we are looking at are essentially subsumption operators. They consider the replacement of the first procedure, or a set of procedures that includes it, with an alternative set of procedures. Since in TraumAID, procedures are originally ordered based on urgency and importance, it follows that under a 1-depth optimization, actions with higher urgency-importance rating at a given time are those that are optimized at that time.

VII. Results

When TraumAID 1.0 was developed, its output was verified by challenging it with 220 different patient care scenarios to test all the rules in the knowledge base. The rules were modified as necessary until TraumAID's management of each scenario was considered acceptable by the domain expert. For TraumAID 2.0, an extended set of 266 cases were used to verify that the management plans produced by the new planning module were likewise considered acceptable by the domain expert. In three cases, the domain expert volunteered that the plans produced by TraumAID 2.0 were, in fact, superior. Subsequent to this verification of the acceptability of the management plan produced by the planning module, a side by side blinded comparison of TraumAID l.0 and TraumAID 2.0 will be done by a panel of experts to test the possible superiority of a flexible planner over the fixed plan of the original TraumAID system.

VIII. Summary

In this paper, we identify some features of trauma management that challenge some classical assumptions of planning research. To cope with these features, TraumAID's architecture integrates diagnostic reasoning and planning capabilities. In that architecture, planning is part of a continuous process in which knowledge is acquired and refined, goals are formed and revised, and plans are constructed, carried out, and verified. Inherent to that architecture is the fact that action is often taken under considerable uncertainty and unpredictability.

The authors believe that TraumAID demonstrates a flexible approach to the diagnosis and therapy of multiple injuries in an uncertain and unpredictable domain, and that this flexibility comes in part from the type of Progressive Horizon Planner that we have described here.

Acknowledgements

We thank other members of the TraumAID project (Jim Farrelley, Abigail Gertner, Jonathan Kaye, Kaiti Moore, Michael Niv and Charles Ortiz Jr.) and anonymous reviewers for their help with this paper.

References

[1] P. Agre, and D. Chapman, Pengi: An Implementation of a Theory of Activity. *AAAI-87*, Seattle WA, pp. 268-272, 1987.

[2] D. Chapman, *Planning for Conjunctive Goals*. Masters Thesis, MIT-AI-TR-802, MIT Laboratory for Artificial Intelligence, Cambridge MA, 1985.

[3] J. R. Clarke, D. Cebula, and B. L. Webber, A Computerized Decision Aid for Trauma, *Journal of Trauma* 28, pp. 1250-1254, 1988.

[4] J. R. Clarke, M. Niv, and B. L. Webber, Computerized Patient-specific Protocols for Managing Penetrating Chest Injuries (abstract). *Theoretical Surgery* 5(3), p.148-149, 1990.

[5] M. Drummond, and K. Curie, Goal Ordering in Partially Ordered Plans. *Proc. IJCAI-89*, Detroit MI, pp. 960-965, 1989.

[6] C. Elkan, Incremental, Approximate Planning. KRR-TR-89-12, Technical Report, University of Toronto, 1989.

[7] G. Ernst, and A. Newell, *GPS: A Case Study in Generality and Problem Solving*. Academic Press, New York, 1969.

[8] R. E. Fikes, and N. J. Nilsson, STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. *Artificial Intelligence* 2, pp. 189-208, 1971.

[9] M. P. Georgeff, and A. Lansky, Reactive Reasoning and Planning. *AAAI-87*, Seattle WA, pp. 677-682, 1987.

[10] M. L. Ginsberg, Universal Planning: An (Almost) Universally Bad Idea. *AI Magazine*, 10(4), pp. 40-44, 1989.

[11] M. L. Ginsberg, Defaults and Hierarchical Problem Solving. Technical Report, Stanford University, Stanford CA, 1990.

[12] R. M. Karp, Reducibility Among Combinatorial Problems. In Miller and Thatcher eds., *Complexity of Computer Computations*, Plenum Press, New York, pp. 85-103, 1972.

[13] R. E. Korf, Real-time Search for Dynamic Planning. *Proceedings AAAI Spring Symposium on Planning in Uncertain, Unpredictable or Changing Environments*, Stanford CA, 1990.

[14] P. L. Miller, *A Critiquing Approach to Expert Computer Advice: ATTENDING*. Pitman Publishing, London, 1984.

[15] R. Rymon, B. L. Webber, and J. R. Clarke, Towards Goal-directed Diagnosis (Preliminary Report). *Proc. 2nd International Workshop on Principles of Diagnosis*, Milan, Italy, pp. 23-39, 1991.

[16] E. D. Sacerdoti, Planning in a Hierarchy of Abstraction Spaces. *Artificial Intelligence*, 5, pp. 115-135, 1974.

[17] E. D. Sacerdoti, *A Structure of Plans and Behavior*. New York: American Elsevier, 1977.

[18] M. J. Schoppers, *Representation and Automatic Synthesis of Reaction Plans*. Ph. D. Thesis. Computer Science Department, University of Illinois, 1989.

[19] J. D. Tenenberg, *Abstraction in Planning*. Ph. D. Thesis. Department of Computer Science, University of Rochester, 1988.

[20] J. van der Lei, *Critiquing Based on Computer-Stored Medical Records*. Ph. D. Thesis, Dept. of Medical Informatics, Erasmus University, 1991.

[21] B. L. Webber, R. Rymon, and J. R. Clarke, Flexible Support for Trauma Management through Goal-Directed Reasoning and Planning. *Artificial Intelligence in Medicine*, 4(2), pp. 145-163, 1992.

[22] D. E. Wilkins, Domain Independent Planning: Representation and Plan Generation. *Artificial Intelligence*, 22, pp. 269-301, 1984.

[23] D. E. Wilkins, Can AI Planners Solve Practical Problems? *Computational Intelligence*, 6(4), pp. 232-246, 1990.