Satisfying Multiple Rated-Constraints in a Knowledge Based Decision Aid
Michael Benjamin, Thomas Viana, Karen Corbett, Ann Silva
Naval Undersea Warfare Center, Division Newport
Building 1171-1, Newport, Rhode Island 02840
email: benjamin@ada.nusc.navy.mil

Abstract

The growing capabilities of information gathering systems in the submarine combat environment continues to generate interest in real-time decision support modules [N92], [P92], [S92], [V92]. The Maneuver Decision Aid (MDA) is a knowledge based expert system designed to simulate the command level human maneuver generation process. A key aspect of this model is that all identified situation goals and constraints are represented and combined according to their varying degrees of importance. This paper describes how constraint ratings are evaluated and combined within an object oriented blackboard framework.

AI topic: Constraint satisfaction, knowledge acquisition & representation.
Domain Area: Human decision aids, expert knowledge based systems
Language/Tools: Smalltalk-80, Blackboard Architecture for Control
Status: Functional prototype
Effort: 5-6 person years
Impact: Provides knowledge representation schema native to domain expert/submarine commander's reasoning process.

1 - Introduction

The Maneuver Decision Aid (MDA) is a knowledge-based planner currently being developed at the Naval Undersea Warfare Center. The goal of MDA is to generate a (theoretically) optimal next-maneuver recommendation, to support the submarine commander's decision making process, which includes absorbing an extensive amount of complex information in the short time interval between maneuvers.

The model of the Maneuver Generation Process (MGP) decomposes into three main parts: Situation Assessment, Goal Formulation, and Constraint Satisfaction. It is essential for an automated reasoning system in this domain to consider all factors and concerns in a given situation in order to put forth its best recommendation. Each of the goals identified in a situation directly leads to a set of constraints on the four motion variables (course, speed, depth, time), making it possible to consider the problem as a constraint satisfaction problem (CSP) [VH89], [K92]. In the submarine tactical domain however, it is often the case that no solution, or maneuver, exists that will satisfy all identified constraints. The typically over constrained nature of the problem is countered by propagating the priorities of maneuver goals to produced rated-constraints and choosing a feasible solution from a maximal subset of the original constraints.

The focus of this paper is on how rated-constraints are formed, represented, and combined to provide optimal solutions. The overall approach to implementing MDA was to adapt the Blackboard Architecture for Control [H79], [H85], [H87], [N86], [P89] as a framework for problem representation and solution derivation. In addition, the Smalltalk-80 object oriented environment was selected as the development environment [G89], [W90], [B91], [R90], [T89]. Although each paradigm has strongly influenced the overall implementation approach, the object oriented methodology is more central to the discussion here and, where possible, key portions of object definitions and behavior will be provided.

2.0 The formation of goals and constraints

Maneuvering in the submarine environment means choosing the best maneuver based on uncertain information and conflicting or competing goals. The approach described here is based on rating all goals and their sub-goals, and propagating these ratings down to maneuver constraints. The ability to recommend the (theoretically) optimal maneuver rests on a proper assignment of ratings to goals and constraints, and an efficient method for solving the resulting optimization problem.

2.1 The goal hierarchy

The MDA approach forms a group of maneuver goals based on the set of situation variables having been posted on the blackboard by situation assessment knowledge sources (KSs) during previous control cycles. A maneuver goal may describe objectives to be achieved or situations to be avoided. The goal generation KS traverses down the class hierarchy invoking each class' evaluation methods, continuing down as long as the
parent goal evaluates to a non-zero value.

The goals that are deemed to be relevant to the current situation have instances created and placed on the blackboard. Each goal instance (as in figure 1) has behavior defined to calculate its importance rating based on three factors: the importance rating of its parent goal, the extent to which the subgoal contributes to its parent, and the degree to which the goal in question may have been already met.

![Goal level of domain panel on MDA Blackboard](image)

Each branch of the goal hierarchy leads directly to a set of sub-goals and corresponds to a set of constraint areas. This is due to the fact that often there is more than one set of maneuvers that will achieve the leaf goal behavior. The positive contribution ratings are associated with states to be avoided.

An instance of an MDA constraint is actually composed of a collection of objects called constraint areas. The boundary of a constraint area is defined by \( <X_{ic}, X_{is}, X_{id}, X_{it}> \), as shown in figure 2. Each constraint area also has a value called the contribution rating associated with it that indicates to what degree a maneuver in its region will contribute to the behavior described by the corresponding leaf goal. This number lies in the range \([-\infty, 1]\). With the contribution rating, a constraint area is a 5-tuple of the form,

\[
c(X_{ic}, X_{is}, X_{id}, X_{it}, CR_i) \text{ s.t. } CR_i \in [-\infty, 1].
\]

2.2 Forming constraints from low-level goals

The identification of a complete set of low level, or "leaf" goals is a description in terms of desired maneuver effects. The job of translating that desired behavior into motion constraints is a formidable task given to the domain experts. The expressive power of the knowledge engineer's choice of representing constraints must measure up to the language in which the domain expert expresses his expertise.

A domain expert can describe any one maneuver in terms of the four motion variables \( I = \{X_c, X_s, X_d, X_t\} \), which take on their values from the finite domains \( D_c, D_s, D_d, D_t \). The leaf goals implicitly refer to a set of maneuvers in the solution space corresponding to the desired behavior of the leaf goal. Therefore a 4-tuple, \( <X_{ic}, X_{is}, X_{id}, X_{it}> \), is a subset of the cross product, \( D_c \times D_s \times D_d \times D_t \), usually associated with a degree of compatibility for certain combinations of variables. This 4-tuple is generally an instance of the class region. The form of \( X_i \) is represented by an interval \( [\text{min}, \text{max}] \).

An instance of an MDA constraint is actually composed of a collection of objects called constraint areas. The boundary of a constraint area is defined by \( <X_{ic}, X_{is}, X_{id}, X_{it}> \) as shown in figure 2. Each constraint area also has a value called the contribution rating associated with it that indicates to what degree a maneuver in its region will contribute to the behavior described by the corresponding leaf goal. This number lies in the range \([-\infty, 1]\). With the contribution rating, a constraint area is a 5-tuple of the form,

\[
c(X_{ic}, X_{is}, X_{id}, X_{it}, CR_i) \text{ s.t. } CR_i \in [-\infty, 1].
\]

![Figure 2: Deriving variable constraints to realize a leaf goal](image)

One constraint frequently generates several constraint areas. This is due to the fact that often there is more than one set of maneuvers that will achieve the behavior described by the leaf goal. There are also sets of maneuvers that will achieve the leaf goal behavior, but to a lesser degree, resulting in a lesser rating association. All these constraint areas and their ratings are kept as an ordered collection in the MDAConstraint object, which henceforth will be referred to as a rated-constraint.

The variable combinations that are very favorable have values approaching 1.0, while those combinations approaching "unacceptable" are given values that grow toward \(-\infty\). The positive contribution ratings are generally derived from goals that describe objectives to be achieved, while the negative values come from goals that are associated with states to be avoided.

\[1 \quad X_c \leftrightarrow \text{course}, X_s \leftrightarrow \text{speed}, X_d \leftrightarrow \text{depth}, X_t \leftrightarrow \text{time}\]

\[2 \quad D_t \text{ ranges from a logical min to max, with an arbitrarily } \Delta = 1 \text{ unit for each. For example, } D_{ic} = [0, 359].\]
The ability to explicitly indicate maneuvers to be avoided as well as those that achieve a certain gain, is a significant tool that allows a domain expert to readily express his expertise. It can also be seen as a more expressive version of existing geometric constraint-based reasoning techniques developed to address maneuvering around obstacles in the underwater environment [Ty89], [S90]. The price is that the techniques described here are needed to combine these more expressive constraints.

The result of combining all MDAConstraints is an objective function that associates a rating for each element in the cross product Dc x Dg x Dd x Dt. The "solution" therefore is the variable assignment with the highest rating, corresponding to the maneuver that satisfies the most goals, or the highest priority goals, or some combination of the two. This solution will be an element of the space where some maximal subset of the entire set of constraints is defined.

2.3 The Constraint Generation KS

Once the knowledge is available for each constraint instance to create and rate its constraint regions, a simple knowledge source oversees the proper formulation of the constraint collection as a whole. As shown in figure 3, the KS retrieves the set of leaf goals posted in previous control cycles and uses this to create the corresponding constraints to be posted in the blackboard Constraint Level.

**Figure 3**: The Constraint Generation Knowledge Source

Once these initial constraints have been posted on the blackboard with their individual ratings, the task of combining them to reveal the optimal maneuver is undertaken.

3.0 Combining multiple rated-constraints

Two additional stages emerge after constraints have been posted by the Constraint Generation KS. The optimal maneuver recommendation surfaces as a direct result of the last stage. The first additional stage after constraint generation is marked by the creation of an ordered collection of regionFunctions directly from the posted constraints. One regionFunction is created for each constraint area posted. A regionFunction is a 5-tuple of the form:

\[ rF_i = < x_{iC}, x_{iS}, x_{id}, x_{it}, a_i > \]

where the first four elements are the boundaries of the constraint area, and the fifth element is called the alpha value. The alpha value is simply the product of the importance rating of the leaf goal and the constraint area's contribution rating. Each regionFunction associates a rating to one finite set of maneuvers, however two particular regionFunctions, \( rF_1 \) and \( rF_j \), may intersect and give ratings to a common set of maneuvers. When the regions of two regionFunctions do not intersect they are said to be mutually exclusive (ME).

\[ ME(rF_1, rF_j) \text{ iff } \neg (\exists < x_{c}, x_{s}, x_{d}, x_{t} > \text{ s.t. (}} \]

\[ (((x_c \in X_{ic}) \text{ AND } (x_c \in X_{jc})) \text{ AND } \]

\[ ((x_s \in X_{is}) \text{ AND } (x_s \in X_{js})) \text{ AND } \]

\[ ((x_d \in X_{id}) \text{ AND } (x_d \in X_{jd})) \text{ AND } \]

\[ ((x_t \in X_{it}) \text{ AND } (x_t \in X_{jt})) \text{ )} \]

\[ NME(rF_1, rF_j) \text{ iff } \neg ME(rF_1, rF_j) \]

An expected characteristic of the set of regionFunctions \( rFs \) derived from constraints is that they have numerous pairs of \( rFs \) that are not mutually exclusive. This is due to the supporting and conflicting interests of many pairs of goals.

The aim of the last stage is to combine all the regionFunctions derived from each of the rated-constraints to build a collective regionFunction (crF) of the form below, where an overall unique rating is assigned to each element of the solution space.

\[ crF = f(x_c, x_s, x_d, x_t) = \{ \]

\[ \alpha_1 \text{ if } < x_c, x_s, x_d, x_t > \in < X_{1c}, X_{1s}, X_{1d}, X_{1t}> \]

\[ \alpha_2 \text{ if } < x_c, x_s, x_d, x_t > \in < X_{2c}, X_{2s}, X_{2d}, X_{2t}> \]

\[ \alpha_3 \text{ if } < x_c, x_s, x_d, x_t > \in < X_{3c}, X_{3s}, X_{3d}, X_{3t}> \]

\[ \cdots \]

\[ \alpha_n \text{ if } < x_c, x_s, x_d, x_t > \in < X_{nc}, X_{ns}, X_{nd}, X_{nt}> \]

0 otherwise

The function crF can be thought of as an objective function. The default rating of a maneuver is zero as indicated, meaning that a maneuver that does not fall under the influence of any goals will have a neutral rating.

The final stage is complete once the crF consists of
a sorted collection of regionFunctions such that all rFs are pairwise mutually exclusive:
\(-\exists rF_i \exists rF_j (rF_j \neq rF_i) \text{ AND NME}(rF_i, rF_j).\)

These three stages are depicted in figure 4.

The collection of mutually exclusive regionFunctions is built by combining and splitting the non-mutually exclusive rFs, and then sorting them on \(\alpha\). Once this sorted collection is completed, the (theoretically) optimal maneuver \(<x_c, x_s, x_d, x_p>\) will lie in the region \(<x_c, x_s, x_d, x_p>\) of the region Function:
\[ rF_1 = <x_c, x_s, x_d, x_p> \text{ s.t. } \neg(\exists rF_j (\alpha_i < \alpha_j)). \]

3.1 Combining regionFunctions

To build the set of mutually exclusive regionFunctions, all non-mutually exclusive rFs must be combined and split individually. When two non-mutually exclusive regionFunctions (rF_i, rF_j) are combined, they form a non-empty set of mutually exclusive regionFunctions:
\[ \{rF_{ij}, (rF_{i1}, ..., rF_{in}), (rF_{j1}, ..., rF_{jn})\}, \]
where rF_{ij} is defined as:
\[ (X_{ij} \cap X_{j}), (X_{ij} \cap X_{j}), (X_{ij} \cap X_{j}), (X_{ij} \cap X_{j}), \alpha_i<\alpha_j> \]
and rF_{i1} ..., rF_{in} describes the remaining portion of rF_i that was not intersected by rF_j. The alpha values are combined to give a new rating to this area of solution space where two goals/constraints have exerted their influence. If \(X_y\) is in the form of \([\text{min}, \text{max}]\), then combining regions amounts to combining hypercubes; the two dimensional analogy is shown in figure 5.

Figure 5: 2-D analogy for combining regions

The constraint combination knowledge SOurce creates the collective regionFunction by building the sorted collection of mutually exclusive regionFunctions. It does this by fetching the first element, rF_1, of the non-mutually exclusive rFs and comparing it to each of the other rFs in the collection. If all other regionFunctions are mutually exclusive with rF_1,
\[ \neg(\exists rF_j (rF_j \in \{rF_2, ..., rF_n\}) \text{ AND NME}(rF_1, rF_j)), \]
then rF_1 is placed in the sorted collection of mutually exclusive rFs on the blackboard Constraint Level. If the above condition does not hold for rF_1,
\[ \exists rF_j (rF_j \in \{rF_2, ..., rF_n\}) \text{ AND NME}(rF_1, rF_j)), \]
then rF_1 and rF_j are combined to form the set
\[ \{rF_{ij}, (rF_{i1}, ..., rF_{in}), (rF_{j1}, ..., rF_{jn})\}, \]
described above, and these new rFs replace rF_1 and rF_j in the set of non-mutually exclusive rFs. The process is repeated by the KS until the original set is empty.

Figure 6: The final two knowledge sources

Once the Constraint Combination KS has completed its work, the Maneuver Generation KS is triggered, as indicate in figure 6. Upon its execution a maneuver, \(<x_c, x_s, x_d, x_p>, \) is generated from the highest rated mutually exclusive regionFunction on the constraint level.
4.0 An Example

An example is difficult to show given the average number of constraints and their constraint areas in a given situation. However, four non-tactical leaf goals and their constraints have been chosen to illustrate the stages of the process. The four goals that have been chosen are shown in figure 7, along with arbitrarily assigned importance ratings for a particular scenario.

Figure 7: Four goals and importance ratings

As described earlier, each of these leaf goals will cause a constraint to be formed by the Constraint-Generation knowledge source. Once each of the constraint instances has been created, messages are sent to each to initiate the building of the constraint areas. The constraint areas are of the form shown in figure 4, and the values for this example are shown in figure 8. The alpha values for the region functions that will follow are also given.

Figure 8: Constraint areas for four goals

When a particular constraint does not constrain a variable, the entire logical range is given, i.e., c[000,359] indicates that the course variable is unconstrained as in the speed-depth constraints in figure 8.

A set of region functions is built from these constraint areas (figure 4) and combined into the set of disjoint or "mutually exclusive" region functions shown in figure 9. The alpha value for each region function is obtained by multiplying the contribution rating of the constraint area by the importance rating of the corresponding goal.

Figure 9: Result of combining constraints

From the highest rated region function, an appropriate maneuver recommendation can be derived. One method of doing so would be to choose a point central to each interval of the region. Thus, the maneuver: <x_v(22), x_v(10), x_v(200), x_v(47)> would be recommended in this example. Another method is to choose the maneuver in this region closest to the ship's current motion parameters.

The two highest rated regions in this example "border" each other and also have equal ratings, perhaps causing one to wonder why the two do not exist as one region. The answer lies in the way that each region has accumulated its rating with respect to individual constraint satisfaction, as described below.

5.0 Justification by constraint satisfaction

Once a maneuver has been recommended through the above process, the requirements of the decision aid have not been exhausted. Given the high consequences of any maneuver, the likelihood of a recommendation having a strong influence on the commander's decision-making process rests on the ability to justify a recommendation. A major portion of justification is in the form of displaying constraint and goal satisfaction. The degree to which a particular maneuver satisfies a constraint can be given by:

\[
\text{Constraint_Sat(aManeuver, aConstraint)} = \frac{\text{CRpoint} - \text{CRmin}}{\text{CRmax} - \text{CRmin}}
\]
where CRpoint is the contribution rating of a particular maneuver to a constraint. CRmax and CRmin are the maximum and minimum CR values of all constraint areas within a constraint instance. For example (figure 8) Constraint #001 has a CRmin = -1.0, and a CRmax = 0.1. For the maneuver

< x_c(100), x_a(12), x_d(185), x_t(30) >

the value of CRpoint is -0.2. Therefore the above maneuver would satisfy Constraint #001 by 73%.

The following four maneuvers lie within the four regionFunctions denoted by (8) in figure 9:

Man1 = x_c(025), x_d(10), x_d(200), x_t(20).
Man2 = x_c(025), x_d(08), x_d(200), x_t(20).
Man3 = x_c(100), x_d(12), x_d(185), x_t(30).
Man4 = x_c(200), x_d(15), x_d(300), x_t(40).

The individual constraint satisfaction values are given in Table 1 below. Note that Man1 and Man2 lie within the top two rated regions respectively. By examining the degrees to which they satisfy the four constraints (and thus the corresponding goals), one may notice that the top two regions do in fact distinguish themselves with respect to what goals are satisfied and to what degree.

<table>
<thead>
<tr>
<th></th>
<th>#001</th>
<th>#002</th>
<th>#003</th>
<th>#004</th>
</tr>
</thead>
<tbody>
<tr>
<td>Man1</td>
<td>100%</td>
<td>10%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>Man2</td>
<td>100%</td>
<td>30%</td>
<td>80%</td>
<td>100%</td>
</tr>
<tr>
<td>Man3</td>
<td>73%</td>
<td>0%</td>
<td>80%</td>
<td>70%</td>
</tr>
<tr>
<td>Man4</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
</tbody>
</table>

Table 1: Constraint / leafGoal Satisfaction

Further justification may be in the form of displaying other stages of the solution derivation process. The nature of the blackboard domain panel makes it easy to access any portion of the process. For example a user may be interested in knowing how MDA interpreted the environmental situation by examining the objects on the status level of the Domain panel. Key premises of the solution and their certainties can then be altered to examine their effects on the solution.

The importance ratings of certain goals can also be changed to see the effects of the maneuver recommendation. It is interesting to note that when goal importance ratings are altered, the form of the combined regionFunctions (figure 9) is not altered, although its ordering is. By changing the importance rating of goal 002 from 10 to 12, the region-Function:

rf = [c(000,045), a(1.0,3.0), d(190,210), t(5.90), α=32.0 changes its α-value from 30 to 32, now the highest rated region. The form of the constraintAreas directly determines the form of the combined regionFunctions, although not their ordering.

Finally, the maneuver recommendation is displayed to the user on a display similar to the one shown in figure 10. The full maneuver recommendation is given, but the regions shown pictorially are in terms of the speed and time parameters only.

Figure 10: The Maneuver Recommendation display

The user interacts with this display to determine what goals have been satisfied and to conduct "what-if" queries.

6.0 Conclusions

Overall three implementation choices were made to actualize the human decision making model constructed through expert knowledge acquisition sessions. The rated-constraint combination technique allows the implementors to build knowledge structures in a format similar to the ones naturally generated by the experts. By employing the combination technique described, an exhaustive search of the solution space (roughly 100 million points) is avoided. More traditional approaches for solving the optimization problem were explored such as integer programming techniques. The task of recasting the problem into a form where these techniques could be applied was found to be non-intuitive and difficult at best, due in part to the need to introduce many new variables [VH89], and the inability to find a feasible solution in general.

The blackboard architecture has provided the ability to represent and develop the solution process in a modular and incremental manner. The organization of knowledge

---

3 Actually CRmin = min(0, lowest CRval), and CRmax = max(0, highest CRval) i.e. CRmin of constraint #003 = 0.0, not 0.4. See Man1 and constraint #002 in Table 1.
sources allows for the incorporation of existing expert systems (to aid in the difficult tasks of situation assessment, target motion analysis, etc.). The domain panel histories are also instrumental in the solution justification task described above.

Finally the use of object-oriented structures to implement the described constraint concepts, as well as the entire blackboard and control architectures, has provided the tools to build a working prototype with mature engineering displays in a remarkably efficient manner.

The main drawback of the approach described here is the need to define new behavior to the constraint classes when constraints of different types emerge in the domain. One example encountered so far in our domain is the need to express distance constraints (i.e. speed-time [min, max]). The extensions however do not go beyond the implementation of the constraint class behavior.

References


