A Knowledge-Based Approach to Solving Hedge Design Problems

Michel Benaroch  Vasant Dhar

Information Systems Department
New York University
New York, NY 10006

Abstract

Problems such as the explosive number of hedging alternatives that is constantly growing, the quick decisions risk managers need to make in the face of the speed in which information flows, and the lack of appropriate computerized support in the early phases of the hedge design process, suggest that there is ample scope for risk managers to make suboptimal decisions. In this paper we formulate hedge design as a multi-objective optimization problem. This optimization problem involves several complexities with which existing quantitative solution techniques cannot deal. We also present a knowledge-based system called INTELLIGENT-HEDGER that we have developed to solve the hedge design problem. This system uses an object-centered representation that captures risk managers' deep domain knowledge, and facilitates emulation of first-principles reasoning processes risk managers use to make decisions and explain them.

1 Introduction

In recent years risk managers have been paying greater attention to the problem of hedging large portfolios against various types of events. The problem is inherently complex since various events can affect the many components of the portfolio in different ways. Risk managers have also learned that hedging vehicles for new markets (i.e., new countries, new underlying instruments) open new business opportunities. Because of the complexity involved, hedging has to be viewed not as a means to offset potential losses, but as an investment strategy, that is, to make money.

Two factors make the job of today's risk managers difficult. Firstly, the number of ways to hedge and their complexity is constantly increasing with the globalization of markets and the growing sophistication of financial instruments. As one successful trader explains, "we, as investors, are not as smart as the sum total of the instruments in the marketplace. Therefore, we must use every available instrument and every single exchange to try and capitalize on each opportunity" [4]. Secondly, risk managers need to act quickly as information circulates. Good risk managers decide on risk exposure only after they have answered the questions: Is there a way to hedge? and What is the best way to hedge? Answering these question rapidly is difficult since it requires an intelligent interpretation of the metrics characterizing each of the explosive number of hedging alternatives. Another successful trader comments, "What today's risk managers need is speed" [8].

Current day systems, such as Bloomberg's Financial Markets [5], do sophisticated quantitative analysis of instruments' prices in the course of designing a hedge. They provide no support in the early phases of the analysis process, in which a small number of alternative vehicles is selected from all the available hedging alternatives. This selection problem is combinatorial.

Typical risk managers specialize only in one or a small number of types of instruments, and are therefore prone to consider only a subset of the available hedging alternatives. Moreover, since risk managers need to operate quickly, they often do not have the time investment required to determine whether a better solution exists. In summary, there is ample scope for suboptimizing.

This paper describes a knowledge-based system called INTELLIGENT-HEDGER that we have developed to assist risk managers in hedge design. This system can design hedge vehicles based on a risk manager's specification of the situation to be hedged, the desired payoff-profile, the amount of cash available, etc. The current system exists as a prototype. It uses an object-centered representation to capture risk managers' deep domain knowledge. This knowledge allows the system to make recommendations and explain them intuitively. Moreover, the modularity and flexibility of this representation make it easier to cope with the constant need to revise and expand the knowledge base.

The remainder of this paper is organized as follows. Section 2 presents a hedge design scenario based on which we formulate hedge design as a multi-objective optimization problem. Section 3. Section 4 shows a knowledge-based solution to that optimization problem. Section 5 and Section 6 discuss the knowledge representation and the problem solving strategy used to implement the knowledge-based solution. Section 7 discusses the benefits of using a representation that is developed based on knowledge requirements of the application domain. Section 8 concludes with a number of potential extensions to our system.

2 A Hedging Scenario

Consider a firm that plans to raise capital by issuing bonds. That firm's treasurer believes there is a strong chance that interest rates will increase and a small chance that interest rates will decline prior to the issuance date. If interest rates will rise, the firm will have to offer a higher yield rate on its bond. The trea-
urer therefore wants to use a hedge vehicle that will cap the issuance rate (i.e., protect against an increase, while preserving the ability to benefit from a decline, in interest rates). The following is a possible dialogue between that firm's treasurer (T) and a risk manager (RM) who is trying to design the necessary vehicle.

RM: What is the situation of concern?
T: An increase in interest rates.
RM: Buy bonds.
T: But I won't be able to benefit from a decline in rates.
RM: What is your payoff-profile?
T:

\[
\begin{array}{c|c|c}
\text{Issuance Rate} & \text{unhedged} & \text{hedged} \\
\hline
\text{Interest Rate} & \text{y} & \text{y} \\
\end{array}
\]

RM: Consider buying a put option on bonds.
T: But this would require a large amount of cash for up-front fees. Right now the firm is short on cash.
RM: So buy a call on put on bonds (i.e., caput), which requires much less up-front fees.
T: How does it work?
RM: If interest rates rise, bonds' prices will decline, causing the put's value to increase, and in turn the caput to become more valuable. Thus, the gain on the caput will offset the extra cost of issuing at a higher yield rate, such that the actual yield rate is \( y \). If interest rates decline slightly below \( r \), bonds' prices will rise, causing the put and the caput to become valueless. Thus, the reduced cost of issuing at a yield rate slightly lower than \( y \) will offset the loss on the caput. If interest rates decline sharply, the firm will gain from issuing at a yield rate that is much lower than \( y \).
T: Sounds good. Yet, will it be easy to unwind the caput in case the firm decides not to go ahead with the project?
RM: Are fluctuations in your firm's credit spread small?
T: Yes.
RM: Use an exchange traded option on T-bond rates, not an over the counter option on corporate bond rates.
T: Why is that?
RM: An exchange traded instrument is easier to unwind. It is less customized, and therefore more liquid and less costly to unwind.
T: What is the level of protection the firm gets?
RM: Relatively high. An exchange traded caput on T-bond rates requires no credit exposure, no repo risk,...

3 Modeling Hedge Design

As the above example shows, hedge design can be conceptually formulated as a multi-objective optimization problem (see Figure 1). Problem solving is a two phase process. In the first phase, feasibility constraints such as the hedger's payoff-profile and the amount of cash usable for upfront fees, are used to identify a set of feasible solutions. In the second phase, optimality criteria such as liquidity and tax favorability, are used to select one or a small subset of best solutions from the feasible solutions identified in the first phase. Note that the number of constraints involved and their priorities can change from one case to another, across cases, depending on the specific hedger and the specific situation being hedged.

\[
\begin{align*}
\text{MAX} & \quad \text{Protection-Level} \\
\text{MIN} & \quad \text{Credit-Exposure} \\
\text{MIN} & \quad \text{Repo-Risk} \\
\text{MIN} & \quad \text{Hedge-Cost} \\
\text{MIN} & \quad \text{Setup-Cost} \\
\text{MIN} & \quad \text{Setup-Complexity} \\
\text{MIN} & \quad \text{Maintenance-Cost} \\
\text{MIN} & \quad \text{Maintenance-Complexity} \\
\text{MIN} & \quad \text{Unwind-Cost} \\
\text{MIN} & \quad \text{Unwind-Complexity} \\
\text{MAX} & \quad \text{Liquidity} \\
\text{MIN} & \quad \text{Customization} \\
\text{MAX} & \quad \text{Tax-Treatment-Favorability} \\
\text{MIN} & \quad \text{Accounting-Cost} \\
\text{MIN} & \quad \text{Industry-Regulations} \\
\text{MIN} & \quad \text{Hedge Maturity-Date} \\
\end{align*}
\]

S.T. 

\[
\text{Hedge's Payoff-Profile} \equiv \text{Instrument's Payoff-Profile} \\
\text{Hedge Maturity-Date} \leq \text{Instrument's Maturity-Date} \\
\text{Cash-Available} \geq \text{Upfront-Fees-Required} \\
\text{Collateral-Available} \geq \text{Collateral-Required} \\
\text{Federal & Industry Regulations} \\
\text{Hedger Policies} \\
\text{Position Limits} \\
\]

Figure 1: Hedge design as an optimization problem

Formulated as a multi-objective optimization problem, hedge design is a combinatorial NP-complete problem and hard to solve to optimality. A number of qualitative constraints, such as MIN setup-complexity that have to be modeled using discrete variables add to the complexity of the problem, making it difficult to solve using existing quantitative techniques. It is also difficult to keep the formulation up-to-date as the instruments change.

4 A Knowledge-Based Solution

A knowledge-based solution entails modeling the knowledge required to solve a problem separately from the problem solving strategy. A knowledge-based system is normally comprised of two components - a knowledge-base (KB) that stores knowledge about the problem domain, and a general purpose or a specialized inference engine that use knowledge in a KB to make inferences in the course of problem solving.

Thought the rule-based approach has been used successfully in many knowledge-based applications, this approach is not adequate for hedging [2]. Hedging is a knowledge intensive domain involving volatile knowledge. The overwhelming number of unique market situations that a risk manager may encounter makes it excessively difficult and costly to set up a KB and keep it up-to-date. An alternative knowledge-based approach is therefore in order.

Risk managers' reasoning in hedging is often based on 'first-principles' rather than 'situation-action' rules. In such reasoning risk managers use deep knowledge about atomic domain entities and structural relationships between them, and specialized reasoning processes to make inferences based on which they reach decisions. Indeed, even experienced risk managers sometimes face unfamiliar situations for which they lack 'situation-action' rules, situations in which they must resort to reasoning from 'first-principles'. Moreover, risk man-
agents use such reasoning to explain many of the rules they have already compiled for recurring situations.

Developing a knowledge-based system capable of reasoning from first-principles entails identifying the deep knowledge risk managers use in hedging, and first-principles reasoning processes they employ with this deep knowledge. Let us explain the approach we use to do that.

Designing a hedge instrument for a given situation requires searching the space of hedging alternatives to find one instrument that satisfies all the constraints involved in that situation. One can conceptually look at hedging design as a series of subproblems, each involving a search for all instruments that satisfy one constraint using its own knowledge about instruments and its own problem solving strategy. Solving one subproblem for a given constraint requires the availability of specific items of deep knowledge about elements in the search space, and the use of a specific reasoning process that can determine which elements satisfy that constraint. Accordingly, for each constraint we first identify the type of compiled rules risk managers apply to select instruments that satisfy that constraint. We then analyze risk managers' explanations of such a rule to identify both the exact items of deep knowledge about instruments and the specific reasoning process that risk managers use to generate such an explanation.

Once the necessary items of deep knowledge and reasoning processes have been identified, it is required to develop a representation that captures the identified knowledge and facilitates emulation of the identified reasoning processes. Since hedging involves an extremely large space of hedging alternatives whose elements are constantly changing, such a representation should also provide a number of other features we shall discuss as we go along. We next describe the representation used in INTELLIGENT-HEDGER and explain the guidelines that led to its design.

5 Representing Deep Knowledge

Much of the deep knowledge risk managers use in hedging centers around various types of financial instruments and relationships between them. A financial instrument can be generic (e.g., stock) or compounded (e.g., ratio-spread). A generic instrument is either a basic one (e.g., bond) or a derivative one (e.g., bond option). A compounded instrument is created by combining two or more generic instruments. To design good hedge instruments a risk manager must appraise an extremely large number of instruments. These include many thousands of generic instruments that are traded daily in markets all over the world, and even more compounded instruments that can be created from these generic instruments [4].

There are two types of relationships between instruments. One is a specialization relationship between instruments (e.g., T-bills, T-notes, and T-bonds are all types of Treasury instruments). The other is a causal relationship between derivatives and their underlying instruments, to which we shall refer to as the ON relationship (e.g., a change in the value of stock s causes a change in the value of all options on s).

We store knowledge about instruments in an object-oriented semantic network to which we shall refer to as the ON-ISA network (see Figure 2). In this network, nodes are organized in an ISA hierarchy that reflects specialization relationships between instruments. Each node can represent a class of instruments or an individual instrument. A node representing a derivative instrument is linked via an ON link to the node representing its underlying instrument. Notice that in Figure 2 the class node called compounded has no instances. It is because we initially store in the ON-ISA network only knowledge about generic instruments. As we shall see later, we create compounded instruments from generic ones only when it is necessary.

A node in the network stores knowledge items in attributes that are specific to the class of instruments or to the individual instrument it represents. An attribute can store a value that does not change across situations (e.g., maturity-date), or knowledge about first-principles that permits derivation of an appropriate value for the specific situation being hedged (e.g., payoff profile). The knowledge items stored in a node are selected based on the kind of reasoning required with the class or individual represented. Let us describe few of these knowledge items, and the reasoning processes with which they are used.

6 Problem Solving Strategy

The general strategy we use in hedge design is one of a screen-and-rank process. Recall that hedge design can be conceptually viewed as a series of search subproblems, each involving its own knowledge about elements in the search space and its own problem solving strategy. Accordingly, the objective of the screen part is, given a set of candidate solutions, apply sequentially all feasibility constraints to screen out non-feasible candidate solutions. The objective of the rank part is, given a set of feasible solutions, apply some ranking scheme that is based on optimality constraints on that set to identify a small number of the highest ranking solutions.

The use of a screen-and-rank process for a problem involving an explosive search space and complex reasoning processes with a large number of constraints can require an overwhelming amount of computation. To carry out this strategy successfully, we use a number of heuristics that are based on characteristics of the domain. One heuristic uses inheritance properties of the ISA hierarchy. For example, when we screen out a class node because one of its properties violates a certain constraint, we also screen out every node that inherits that property from that screened out class node. A second heuristic is abstraction of knowledge items stored in the ISA hierarchy. This facilitates reasoning qualitatively about a whole class of nodes, rather than about individual nodes. Another heuristic is the sequencing of reasoning required with each constraint, and based on the number of solutions each constraint is likely to screen out.

6.1 Screening by Feasibility Constraints

The screen part of the process is done by applying one feasibility constraint at a time on nodes in the ON-ISA network (i.e., initial search space). The processing required for a feasibility constraint is relatively simple when a value for the attribute referred to by that constraint is pre-stored in each node in the network. For example, in the case of the maturity-match constraint, the
Figure 2: An ON-ISA object-oriented semantic network

value stored in attribute \textit{Maturity-Date} of every node is compared against the objective \textit{Hedge Maturity-Date}.

Additional processing is required when a value for an attribute referred to by a constraint is not pre-stored. A value for that attribute needs to be derived from other knowledge items using a proper first-principles reasoning process. For example, in the case of the upfront-fees-required constraint, the amount of upfront fees a certain type of instrument requires is not pre-stored for flexibility purposes. Rather, it is derived by analyzing the transactions required to set up a hedge instrument of that type. These transactions are one pre-stored deep knowledge item from which we also derive the setup-complexity, the repo-risk-exposure, and other attributes. Certain constraints require a more complex processing. Let us examine, for example, the case of the payoff-profile constraint.

6.1.1 Processing the Payoff-Profile Constraint

The payoff-profile constraint specifies, subject to the market situation a risk manager previews, the payoff s/he wants to derive using a hedge instrument. It is defined based on assessments of how the future behavior of certain economic variables is likely to affect the value of instruments. It is usually expressed qualitatively as a two-dimensional piecewise linear function.

To identify instruments that satisfy the payoff-profile constraint, one must know what payoff-profile each instrument provides under the market situation being hedged. An instrument may provide a different payoff-profile under each one of the explosive number of unique market situations. Therefore, it is not feasible to pre-store in the ON-ISA network all the payoff-profiles that every instrument may provide.

The hedging scenario in Section 2 demonstrates how risk managers derive the payoff-profile a certain type of instruments provides under a given situation by reasoning from first-principles, that is, using a qualitative causal analysis of structural relationships between fundamental economic variables and the value of instruments of that type (i.e., "If interest rates rise, bonds' prices will decline, causing the put's value...). To facilitate such an analysis one must know the causal relationships relevant for every type of instruments. We therefore store in every class node in the ON-ISA network the \textit{pricing model} which models the major causal relationships relevant for instruments in that class. Note that the pricing model of a certain class can be a specialization of the pricing model of its super class. For example, the pricing model of bond options is a specialization of the Black-Scholes model, which is used to derive the pricing model for various types of options [9].

We produce the payoff-profile of each class of instruments under the market situation being hedged using an algorithm called QSIM (Kuipers, 1986), and store it in that class node. This algorithm uses an abstracted version of the pricing model stored in a class node to emulate the qualitative causal analysis risk managers use to determine how the price of instruments in that repre-
6.1.2 Synthesizing Compounded Instruments

We must first create compounded instruments from knowledge about generic instruments. The number of compounded instruments one can create is as large as the number of different permutations of generic instruments. It is therefore not sensible to create all possible permutations of generic instruments and check which of them satisfy the payoff-profile constraint.

Risk managers synthesize the goal payoff-profile from payoff-profiles of generic instruments (hereafter, generic payoff-profiles), and trace back the generic payoff-profiles used in the synthesis to the specific types of generic instruments that should be used to create good compounded instruments. Searching the space of all permutations of generic payoff-profile against the goal payoff-profile is a generate-and-test problem that is combinatorial and can be shown to be NP-complete. We solve this problem using a technique called qualitative synthesis that we have developed [3]. This technique emulates the process risk managers use to synthesize good compounded instruments.

Qualitative synthesis uses a goal directed process to search the space of permutations of the generic payoff-profiles that QSIM produced for every type of generic instruments. It generates one permutation at a time by algebraically adding and/or superimposing two generic payoff-profiles, and tests it against the goal payoff-profile. Qualitative synthesis constrains its generator using two heuristic synthesis operators. These operators use knowledge about stretching and steepening operations on a two-dimensional piecewise linear function over one of its definitional region to prune permutations of payoff-profiles that do not match the goal payoff-profile.

The use of qualitative synthesis guarantees the creation of all compounded instruments that satisfy the payoff-profile constraint. For every permutation of generic payoff-profiles that matches the goal payoff-profile we create a compounded instrument and represent it as an instance of node compounded in the ON-ISA network. Each such node is also linked via CONFIGURATION links to nodes representing the generic instruments used to create that compounded instrument. A newly created instrument can be examined using knowledge derived via its CONFIGURATION links to determine if it violates other feasibility constraints.

6.2 Ranking by Optimality Constraints

Once all feasible solutions are identified, INTELLIGENT-HEDGER ranks these solutions based on optimality constraints. Ranking solutions by one optimality constraint, say MAX Liquidity, means ranking instruments along the attribute referred to by that constraint.

The values based on which instruments are ranked along certain attributes cannot be pre-stored in the ON-ISA network since they change across market situations. For example, an instrument’s liquidity can be high in one situation and low in another. Consequently, one needs to derive rankings of instruments along such attributes.

The attributes of instruments that appear in optimality constraints can be organized in a hierarchy based on dependency relationships between them. For example, the less customized an instrument the less costly it is to unwind. We call the lowest level attributes in that hierarchy atomic attributes (e.g., price-volatility).

Rankings of instruments along atomic attributes are constructed from knowledge about instruments that is observable in the market, knowledge that we store in each node in the network. Such rankings often form only partial orders because at every moment the information available about observable values of attributes can be imprecise and/or incomplete.

Rankings along non-atomic attributes can be derived by propagation of rankings along atomic attributes up in the hierarchy of attributes. That is, a comparison of the values of a certain non-atomic attribute of two instruments can be done by a comparison of the values of certain other lower level attributes for the two instruments. For example, consider four instruments \( a, b, c, d \) that need to be ranked along customization and liquidity. Assume that customization is equally determined by the atomic attributes — strike-flexibility, number-of-dealers, and trading-volume — and that liquidity is equally determined by the atomic attribute time-to-maturity and the non-atomic attribute customization (see Figure 3). A propagation of known rankings along atomic attributes up in the hierarchy in Figure 3 determines the rankings along customization and liquidity.

Derived rankings along non-atomic attributes are determined by two factors. One is rankings along atomic attributes which change whenever observable values of instruments’ attributes change (e.g., price-volatility). The other is the weights assigned to relationships in the hierarchy of attributes. These weights can change across situations. They depend on how the individual risk manager prioritizes optimality constraints based on the market situations being hedged. Thus, derived rankings are also determined by how risk managers parameterize the hierarchy of attributes.

7 Discussion

INTELLIGENT-HEDGER is a prototype that we have implemented in C++, and that is currently under testing. So far, our experience suggests that a model-based development approach, such as the one de-
scribed in this paper, can be extremely beneficial from a knowledge engineering point of view. It can reduce the amount of effort required for knowledge acquisition by focusing attention only to those knowledge items and relationships that are vital in problem solving, that is, to the real core of domain expertise. It can also provide guidelines to the design of a representation to accommodate special knowledge requirements of the application domain.

We feel that our system's ability to cope with much of the complexity involved in hedge design is greatly due to the object-oriented representation we use. This customized representation has three major advantages over representations such as the rule-based one.

One advantage is its ability to capture much of the deep domain knowledge risk managers use in hedge design, and to support direct reference to the basic structural entities of the domain to allow reasoning from first-principles. This ability facilitates the emulation of reasoning processes risk managers use to make hedging decisions, and thus derivation of many of the situation-action rules one might elicit from a risk manager if one were trying to develop a rule-based system. It also facilitates the generation of intuitive explanations of decisions, something that is important in domains where the role of an intelligent system is to assist domain experts rather than replace them.

The second advantage is the use of object-oriented concepts to provide increased modularity and flexibility. These features promote the notion of an open KB architecture by allowing to relate knowledge about new domain entities to knowledge about entities it already knows about with relatively little effort. It thus can reduce the amount of effort required to revise knowledge in a KB without effecting its usability. This feature also makes the knowledge transferable across related problem areas, something that is important for the integration of INTELLIGENT-HEDGER with systems dealing with other aspects of risk management.

A third advantage is the use of inheritance properties of the ISA hierarchy reflecting specialization relationships between domain entities. This feature can be used to minimize the amount of knowledge stored in the KB, and thus to reduce the amount of effort required to set up the KB and keep it current.

8 Future Research

A preliminary evaluation of INTELLIGENT-HEDGER indicates that it will encounter three major problems in the real world. These problems are currently on our future research agenda.

One problem is the inability to use surface knowledge. Risk managers use certain situation-action rules that they cannot always explain intuitively (i.e., rules that cannot be derived by reasoning from first-principles). It is likely that the inability to use surface knowledge will sometimes lead the system to make decisions that are inferior to ones made by a risk manager. Any attempt to account for such knowledge would require integrating it into our representation without effecting its usability and flexibility. It would also require identifying the relevant surface knowledge as well as understanding when and how to use it in the hedge design process.

Another problem is the amount of time and effort required to parameterize the hierarchy of attributes used to rank feasible solutions by optimality constraints. That hierarchy represents relationships between more than 45 attributes of instruments, of which 30 attributes are atomic. Currently, the hierarchy is parameterized by a risk manager based on the market situation s/he previews, and accordingly the way s/he prioritizes the optimality constraints involved in the situation. Theoretically, this prioritization is greatly determined by the projected behavior of the many economic variables used to characterize the state of certain capital markets and of the economy as a whole. However, the time and cost involved in gathering and analyzing data for the development of such projections for all variables can frequently overrun the potential benefits from hedging.

It seems that risk managers actually use surface knowledge to identify a small set of key variables in a given situation. They then gather and analyze data to develop projections only for variables in that set, based on which they can make assumptions about the remaining variables. We feel that a study of how risk managers identify a different set of key variables in each situation can reveal what kind of default reasoning scheme might enable our system to do much of the parameterization of the hierarchy of attributes with less input from the user.

The third problem is interaction with related systems. For example, one such system can be used to identify underpriced instruments. Risk managers will always prefer to hedge with an underpriced instrument even if it ranks low by optimality constraints. This interaction problem involves communication and control issues that must be first identified and resolved on a conceptual level.

References