Valuation and Optimization for Performance Based Logistics Using Continuous Time Bayesian Networks

Logan Perreault, Monica Thornton, and John W. Sheppard
Computer Science Department
Montana State University
Bozeman, MT 59717
{logan.perreault, monica.thornton}@msu.montana.edu
john.sheppard@montana.edu

***Abstract*—When awarding contracts in the private sector, there are a number of logistical concerns that agencies such as the Department of Defense (DoD) must address. In an effort to maximize the operational effectiveness of the resources provided by these contracts, the DoD and other government agencies have altered their approach to contracting through the adoption of a performance based logistics (PBL) strategy. PBL contracts allow the client to purchase specific levels of performance, rather than providing the contractor with the details of the desired solution in advance. For both parties, the difficulty in developing and adhering to a PBL contract lies in the quantification of performance, which is typically done using one or more easily evaluated objectives. In this work, we address the problem of evaluating PBL performance objectives through the use of continuous time Bayesian networks (CTBNs). The CTBN framework allows for the representation of complex performance objectives, which can be evaluated quickly using a mathematically sound approach. Additionally, the method introduced here can be used in conjunction with an optimization algorithm to aid in the process of selecting a design alternative that will best meet the needs of the contract, and the goals of the contracting agency. Finally, the CTBN models used to evaluate PBL objectives can also be used to predict likely system behavior, making this approach extremely useful for PHM as well.**

I. Introduction

Many modern systems are complex and change over time, making maintenance of these systems difficult. This is especially problematic when given the ubiquity of complex systems in a number of critical domains such as weapon system support. One way to reason about the expected performance of these systems is through the use of risk and prognostic models, which allow us to make predictions about the future states of the system, coupled with the impact of such failures on the ability of the systems to perform their functions to specification.

The continuous time Bayesian network (CTBN) is a probabilistic graphical model capable of describing discrete state systems that evolve in continuous time [1]. Conceptually, a CTBN is a continuous time Markov process factored using a directed graph, where nodes represent variables within the system, and edges represent dependence between these variables. This model has been applied successfully to a number of prognostic and health management (PHM) applications.

While the CTBN, as originally specified, provides information about how the states of a system evolve through time, situations may occur that require measuring and assessing the evolution of these states using more complex performance metrics. Performance functions, as defined by Sturlaugson and Sheppard, allow a user to specify cost and reward values over the behavior of a system [2]. In this way, performance functions provide information beyond the most probable state of the system at a given time. The CTBN model can be augmented with performance functions, allowing users to assign value to preferred system behavior. Although recent work has been done to show that performance functions can be used within the CTBN framework, this advancement has not yet been adopted by the PHM community.

We claim that by augmenting prognostic CTBN models with performance functions, we can support the tasks of prognostics and performance based logistics (PBL). CTBNs provide information about future states of the system, while performance functions are able to encode user-specified metrics. These metrics can be tailored to suit any problem, and it is a natural adaptation to convert from a PBL objective to a performance function. In the context of the DoD, PBL is intended to provide solutions that satisfy fighter requirements affordably, usually expressed in terms of availability [3]. Performance measures such as readiness and cost can be difficult to measure over the lifetime of a system, therefore motivating the use of evaluation methods with greater representational power. CTBN performance functions are able to represent a wide array of complex metrics, and offer a mathematically founded approach to evaluating the performance value of PBL objectives.

In our work, we demonstrate the applicability of performance function augmented CTBNs toward PBL tasks. By way of example, we built a CTBN for a simplified version of a vehicle system. We define a series of performance functions intended to capture different behavioral aspects of the system, measured in terms of cost, availability, and maintainability. Furthermore, we show how these performance metrics are affected by changes to the model, and how this can be used to find a vehicle design alternative that is optimal over the PBL objectives. Finally, we conclude by briefly describing how these models can be used in the context of PHM.

We believe this to be an innovative approach to evaluating PBL objectives that represents a significant contribution to the weapon-system support community as a whole. Through the incorporation of user-specified performance functions, domain knowledge pertaining to the system can be captured and utilized. This results in a model with greater representational power, which is valuable when planning and responding to logistics needs based on emerging performance of the weapon system.

II. Background

Prior to describing our proposed method for evaluating PBL contract objectives, we will first provide background on the key concepts required to explain our approach.

A. Performance Based Logistics

Performance based logistics (PBL) is a contracting strategy intended to enhance the operational effectiveness of large-scale, resource intensive systems that are generally in operation over long periods of time. Employed by a number of government agencies including the DoD and the various service branches of the armed forces, PBL has found success as a procurement and support strategy due to its ability to produce systems with improved performance. The strength of this approach lies in the fact that in contrast to other methods that contract for resources, PBL contracts for performance according to a variety of prespecified metrics [4]. Often, these metrics center around concepts that tie directly to a system's performance, such as availability, reliability, maintainability, and supportability [5].

This focus on performance is advantageous to both the contractor and the client. From the perspective of the contractor, they are afforded the freedom to develop innovative solutions that fulfill the objectives specified in the PBL contract. The contractor is provided with clear objectives that they must meet in order for their proposed solution to be deemed effective. Likewise, the client receives a solution that fulfills the performance and support objectives of the contract, allowing them to balance the operational effectiveness of their systems with the resources required to procure and maintain them.

Although PBL strategies have been adopted by a number of agencies in the public sector, there is great variation in PBL contracts because of differences in scope, needs, contract duration, and system application. The only guaranteed constants among PBL contracts are the specification of the performance goals required by the client and the schedule set to to meet those goals. To be successfully employed, PBL contracts must be explicit, well-defined, and contain objectives that can be evaluated according to a set of available metrics. This clarity provides contractors with the set of objectives they must consider when designing and implementing their solution, as well as the performance incentives and penalties used to ensure that they meet those objectives.

In the case where multiple contractors provide alternative designs for a system, the client needs to select one of the alternatives for implementation and deployment. This choice can be difficult, especially in those cases where there are a large number of alternatives or a variety of objectives that must be met to satisfy the contract. While PBL contracts provide a powerful framework for procuring and evaluating resources, employing this strategy requires methods to address issues of performance measurement and optimization. Work has been done to address these issues, and an overview of existing methods is provided in Section III of this work.

B. Continuous Time Bayesian Networks

A Markov process is a mathematical model capable of describing the temporal behavior of a discrete state system. The model is parameterized using an initial distribution $\mathbf{P}$ over the states of the system, as well as an intensity matrix $\mathbf{Q}$ that defines transition behavior between the states. Each entry q_{ij} in the i^{th} row, j^{th} column of the matrix $\mathbf{Q}$ represents the expected transition rate from state i to state j. Specifically, the time until state i transitions to state j is exponentially distributed with a rate of q_{ij}. Diagonal elements q_{ii} in the intensity matrix $\mathbf{Q}$ are equal to the negative sum of the remaining row, and its absolute value represents the rate at which state i will transition to any other state. The initial distribution $\mathbf{P}$, in combination with the intensity matrix $\mathbf{Q}$, are sufficient to describe any discrete state system as it changes over time.

Although a Markov process is a powerful model capable of describing complex systems, it has a major limitation. The concern is that the number of parameters required to specify $\mathbf{P}$ and $\mathbf{Q}$ are exponential in the number of variables. This makes storing and using Markov processes infeasible when modeling systems with even a moderate amount of variables. To mitigate this problem, continuous time Bayesian networks (CTBNs) have been introduced as a factored representation of Markov processes [1]. A CTBN works by representing each variable in the system as a node in a directed graph G. If a variable's behavior is directly dependent on another, then an edge is placed between the corresponding nodes in G. The model is parameterized in much the same way as a Markov process, except that an initial distribution $\mathbf{P}_X$ and a set of conditional intensity matrices $\mathbf{Q}_{X|Pa(X)}$ are used to describe the temporal behavior of each individual variable conditioned on its parents in the graph.

Depending on the structure of the graph, a CTBN can represent the temporal dynamics of a discrete state system using substantially fewer parameters than a single Markov process, making it possible to model more complex, real-world systems. To that end, CTBNs have been used in a diverse array of practical applications. In 2003 work focused on the challenge of reasoning about user activity over time, CTBNs were used to model a user's presence and availability with respect to a number of different computing applications. These models predicted user activity such as the time at which a user would be present in their home or office locations, or the time at which the user would be available for tasks like email response or videoconferencing, as determined by data such as the application currently in focus [6]. CTBNs have also found success when applied to the task of gene network reconstruction, which is a difficult task that has attracted the interest of the genetics community [7]. Finally, CTBNs have demonstrated their ability to accurately model failure and repair events in mechanical systems [8]. Our focus in this work is centered on this diagnostic and prognostic context for CTBNs.

The task of building CTBNs can be complex; fortunately, there exists several automated methods for constructing models using commonly available information. If data exists that details state transitions for the variables of a system, there are several algorithms available that will build and parameterize a CTBN that matches the observed data [9], [10]. When working with diagnostic models, it is possible to bypass these learning algorithms altogether and instead derive CTBN models directly from existing design information. A model relating tests and faults can be obtained by exploiting the information encoded by a D-matrix, much of which can be derived from a Failure Mode, Effects and Criticality Analysis (FMECA) [11]. Similarly, a CTBN relating faults to effects can be derived directly from fault-trees [12]. The models derived from D-matrices and fault-trees can be combined to obtain a single CTBN relating tests, faults and effects. While the algorithms themselves can be complex, these automated procedures make model construction relatively simple. Once built, the models can be used to query the probability of each variable at any point in time by making use of a variety of inference algorithms [13], [14].

C. Performance Functions

CTBN inference algorithms support basic queries that can return the probability of a variable being in a state over a specified time period. While this information can be extremely useful, it may be insufficient to answer more complex questions regarding the system. Instead, one or more performance functions can be defined for a CTBN, providing a method to capture more complex information. A performance function provides a mapping from the modeled system to a single value intended to represent a metric of interest [2]. This mapping still makes use of inference algorithms to compute expected states; however, the use of inference queries is implicit in this case rather than explicit. This is achieved by assigning positive or negative value to system states or transitions, defined in terms of a mathematical equation that potentially contains time as a variable. To maintain the advantages CTBNs hold over Markov processes, the performance functions themselves are also factored over the nodes in the network, meaning that value is assigned to each variable individually, with the final performance value calculated as an aggregate of the values computed for each node.

By way of example, consider a node X in a CTBN whose domain consists of three states x_{failed}, $x_{partial}$, and $x_{operational}$. A performance function can be defined that assigns value according to the observed states of X over a time period of interest. Let t_s and t_e be the starting and ending times for the period, measured in hours since the beginning of a mission, and let $\Delta t = t_e - t_s$. Then an example performance function for node X is as follows:

$$f_X(t_s, t_e, X(t_s)) = \begin{cases} -5 & \text{if } X(t_s) = x_{failed} \\ 10 & \text{if } X(t_s) = x_{partial} \\ 20 + 3\Delta t & \text{if } X(t_s) = x_{operational} \end{cases}.$$

Here, the function receives a fixed penalty of -5 for entering state x_{failed} and a fixed reward of 10 for entering state $x_{partial}$. State $x_{operational}$ is somewhat more complex in that it produces a fixed reward of 20, as well as an additional incentive of 3 per hour for the duration of Δt. For instance, if X is observed to be in state $x_{operational}$ from time $t_s = 100$ to time $t_e = 105$, then $\Delta t = 5$ and $f_X(100, 105, x_{operational}) = 20 + 3(5) = 35$. By aggregating the value returned by f_X with other variable specific performance functions, a global performance value can be obtained that assigns value to the model as a whole over the specified time period. Furthermore, multiple unique performance functions can be defined that each represent a different metric of interest, allowing the model to be queried for several performance values simultaneously.

D. Optimization using CTBNs

Optimization, in its most basic form, is the minimization or maximization of a function over a set of choices. Performance functions defined over a CTBN produce a value that is typically viewed as a penalty or reward. It is therefore a natural application of optimization to minimize or maximize the performance functions in the model.

In order to incorporate optimization into the CTBN framework, the model must first be adapted to allow for model alterations that will change the output of a performance function. While the graph structure or parameterization of the model could be altered directly, the process may be complicated and will change the original model. Instead, new nodes are introduced into the graph, which we refer to as "control nodes." These nodes are added as parents to other nodes in the network, and given that a variable's behavior is directly dependent on its parents' states in the graph, a control node is capable of altering system dynamics by simply changing its state. When used in the context of optimization, control nodes are manipulated to change their state, which in turn changes the performance values. This is done repeatedly in an attempt to find the optimal performance value.

When considering more than one performance function, there are two common approaches to handling the optimization problem. The first is to combine the optimization problems into a single performance function, which can then be optimized using standard maximization or minimization techniques. The concern is that the task of combining disparate performance metrics is difficult and may be inappropriate in some cases. Instead, multi-objective optimization may better be achieved by providing the Pareto frontier [15]. Formally, the Pareto frontier consists of a set of solutions where any improvement in one of the dimensions comes at the cost of degradation in another dimension. Any solution not in the frontier is considered a dominated solution, and could be improved with respect to all performance values by switching to a member of the Pareto frontier. The advantage to this approach is the identification of the dominated set, which is objectively a suboptimal solution to the optimization problem. This narrows the list of solutions to only those contained in the Pareto frontier, and if choosing a single solution is necessary, it is often done by a domain expert.

III. Related Work

A major motivating factor for our own research relates to the concerns raised by Doerr *et al.* in their 2005 work on measurement issues in PBL [16]. In this work, they highlighted the difficulties in defining consistent and representative measurements in the context of PBL. It was noted that known formulas based on time to failure distributions could be used under the assumption that any component failure causes the entire system to become non-mission capable. The concern is that this is a strong assumption that rarely holds in practice, and overly simplistic objectives may sidestep the intended goals of PBL. For instance, if maximizing operational availability was truly the only objective of relevance, then simply not using equipment would successfully accomplish this goal. When talking about the state of the art for the DoD, the authors state that "while PBL has been decreed as a preferred implementation strategy, real questions remain unanswered about objectives and measurability." The measurement issues discussed by Doerr *et al.* imply that more complex approaches to measurement will need to be employed; an issue we address in this work.

Another avenue that has strongly influenced our approach to improved PBL evaluation is the work done by Kumar *et al.* in 2007 [5]. Their work focused on three performance metrics to be used for PBL contracts: reliability (**R**), maintainability (**M**), and supportability (**S**). To choose a design that best meets the needs of a PBL contract, they make use of the notion of goal programming, which is a mathematical model capable of performing constrained optimization. By defining the relative importance between each of the **R**, **M** and **S** metrics, their goal programming algorithm was able to find a value that simultaneously optimizes over all three objectives. Although this is an interesting approach, it has several limitations. One concern is the consolidation of three distinct objectives into a single metric. Although this enables a single optimal solution to be produced by the algorithm, the task of multi-objective optimization is often better solved using the Pareto frontier approach, especially given that the correct choice can be very problem specific. Another facet of this work that we aim to improve is the simplicity of the **R**, **M** and **S** metrics. We contend that the measures used by Kumar *et al.* are overly simple and do not reflect real-world system dynamics, as they use fixed values that fail to capture more complex valuation as the system evolves in time.

Villanueva Jaquez extends the work of Kumar by employing a heuristic search during the optimization process, rather than enumerating every possible design alternative [17]. More specifically, the author uses a genetic algorithm (GA) to search iteratively for an optimal solution. As with many search heuristics, GAs work by evaluating a subset of the possible solutions, and based on the initially observed values, the algorithm chooses to explore solutions that are similar to the best values seen so far. Aside from relaxing the requirement that the alternatives are fully enumerated, this work improves upon Kumar *et al.* by using a Pareto frontier to describe the optimal values. Despite these advancements, the work by Villanueva Jaquez still suffers from the problem of using overly simplistic metrics, which as discussed by Doerr *et al.*, is a practical problem that needs to be addressed when using PBL.

Our work differs from prior work in this domain in that we are focusing on providing the PBL community with a framework for modeling complex contract objectives. This is a task whose importance has been acknowledged, but has yet to be addressed fully. Furthermore, although optimization of PBL objectives has been considered in the literature, we show how this can be achieved better using the CTBN framework. This novel method will allow more intricate objectives to be optimized, providing a method for identifying design alternatives that better meet the needs specified by a PBL contract.

IV. Common PBL Objectives

Consider a contract focused on obtaining a new military vehicle such as a High Mobility Multipurpose Wheeled Vehicle (HMMWV). Under the guidelines of PBL, rather than explicitly specifying how such a vehicle should be designed and produced, the client provides a set of required performance values. The details of how to design a vehicle that obtains these levels of performance are left to the contractor who is awarded the PBL contract. While this is a logistically effective approach, it relies heavily on the ability to evaluate the performance needs of the client. With many government agencies these needs are often highly complex, with multiple, potentially conflicting objectives. Consistently and fairly evaluating solutions provided by contractors will require a mathematically founded approach capable of evaluating and optimizing over these multiple objectives.

A CTBN is well-suited to model complex systems by defining the temporal interactions between the various subcomponents. Figure 1 shows an example of a CTBN that might be used to model a military vehicle such as a HMMWV. This model has been adapted from the one introduced in Cao's work, which used failure rates and deterministic gate functions to parameterize a CTBN [8]. Here, the vehicle is described by three major subsystems, each of which are represented using a subnetwork of nodes. The wheels/tires (WT), brakes (BR), axle (AX), and suspension system (SU) all feed into the chassis (CH) subsystem. Similarly, the engine (EG), transmission (TR), and cooling system (CO) make up the power train (PT) subsystem. Finally, the electrical (EL) subsystem is modeled by a single node. The Vehicle node represents the operational state of the system as a whole and is directly influenced by all three of the major subsystems. Dependencies between components are represented using directed edges. The bidirectional edges indicate that failure can propagate in either direction. For instance, damage to the wheels/tires can compromise the chassis, which in turn can cause damage to subcomponents like the axle. The failure rate and repair rate parameters for this network are again based on the values provided by Cao's experiments. The initial production cost[1] and estimated repair cost for each component are provided in Table I.

Approaches to fulfilling PBL contracts vary significantly across agencies and applications, meaning that there is not a "one-size-fits-all" method for performing PBL [20]. In this work, we incorporate three commonly used performance measures: cost, availability and maintainability. Cost, in this context, refers to the total cost of ownership (TCO) as measured in US dollars (USD), a measure which includes the initial production costs and any subsequent costs over the life of the solution. We draw our definitions of availability and maintainability from existing definitions provided by Ebeling [21]. Availability is the percentage of time a unit or system

[1]Production costs are loosely based on published material, but costs are ultimately hypothetical values intended for demonstration purposes only [18], [19].

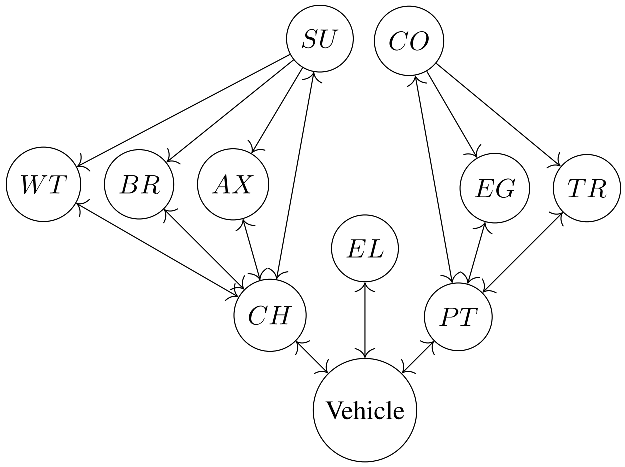

Fig. 1. Example CTBN for the Vehicle Model

TABLE I. COSTS FOR THE BASE VEHICLE MODEL IN FIGURE 1, AS MEASURED IN USD

Component	Production Cost	Avg Repair Cost
EL	8000	1200
BR	3500	950
WT	4000	700
AX	5000	2000
SU	8000	850
EG	14000	450
TR	7500	6500
CO	3500	200
$Body$	25000	N/A
$Assembly$	9000	N/A

is able to perform its intended function. Availability can be computed as follows:

$$A = \frac{t_o}{t_o + t_f} \tag{1}$$

where t_o represents the amount of time the component spends in the operational state and t_f represents the amount of time the component is not operational. Similarly, maintainability is the conditional probability that a component will be repaired to an operational condition within some prespecified amount of time, given that the component has failed. Note that this definition of maintainability is an approximation suggested by Doerr, and we recognize that this does not conform to the standard definition. Maintainability can be computed using the following equation:

$$M = \frac{n_r}{n_r + n_f}. \tag{2}$$

Here n_r corresponds to the number of observed instances where the time required to repair the component did not exceed a prespecified threshold τ. Conversely, n_f corresponds to the number of observed instances where the time required to repair the component exceeded τ. The described measures for cost, availability and maintainability are in this case simplistic; however, the approach to evaluation described in this work can be applied to substantially more complex objectives as well.

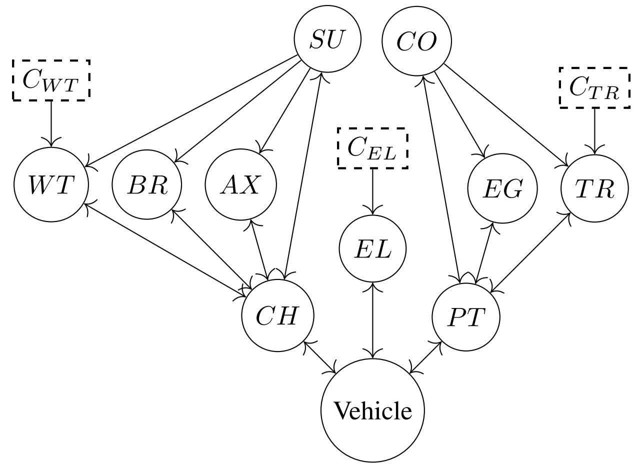

Fig. 2. Example CTBN for the Vehicle Model, augmented to include control nodes

V. DEMONSTRATION OF PBL VALUATION

Given the previously defined cost, availability and maintainability metrics, the client must specify desired values for each objective. By way of example, a contract for a military vehicle may require a maximum cost of $T_C = \$950000$, a minimum availability of $T_A = 0.7$, and a minimum maintainability of $T_M = 0.65$. Contractors will present a number of different designs that address each of these objectives to varying extents, and a valid solution must achieve all of the goals simultaneously, while still meeting additional specifications that are not based on performance measures. That said, there is the potential for a number of valid solutions, which may make the process of deciding between competing options very difficult for the client. To this end, we require a method capable of evaluating multiple design options easily, while also identifying those designs that are objectively superior.

To show how this process would work in practice, refer back to the CTBN model depicted in Figure 1. Consider the case where a contractor has alternative design options for several of the components. Specifically, assume there are two alternatives for the TR component, corresponding to a 3-speed or 4-speed transmission style. Next, assume there are three options for WT: a standard tire option, one with more aggressive tread, and one that is equipped with a central tire inflation system. Finally, let the electrical system EL have four design alternatives. One alternative is can be considered the default case, while the remaining three provide an alternative choice for either the alternator, starter, or electronic control unit. To model these alternatives, and their impact on the behavior of the system, the base CTBN structure is augmented to include three additional nodes. Referred to as control nodes, these three nodes are represented as the dashed rectangles shown in Figure 2 and the number of states for the node equates to the number of alternatives for that component. Control nodes are added as a parent to their corresponding component, and represent possible alterations that can be made to the network. These alterations are achieved by assigning a state to the control nodes over the time period of interest. Given that nodes are directly dependent on the state of their parents, the children of control nodes change according to the state assignment. Note that each state assignment to the control

TABLE II. DESIGN ALTERNATIVE MODEL ALTERATIONS

	Alternative	Failure Rate	Repair Rate	Cost
TR	A_0	1.00	1.00	1.00
	A_1	2.30	0.10	1.80
WT	A_0	1.00	1.00	1.00
	A_1	3.00	0.60	0.40
	A_2	0.30	1.60	2.20
EL	A_0	1.00	1.00	1.00
	A_1	2.10	0.45	0.35
	A_2	0.50	1.90	2.20
	A_3	1.80	3.10	1.30

nodes in Figure 2 corresponds to a unique set of vehicle design alternatives.

Recall that there are two alternatives for the TR component, three for WT, and four for EL. In this work, we consider the baseline model to be the case where TR, WT and EL have all been chosen to be alternative A_0. For conciseness, we refer to this design decision using the string `(WT:0;EL:0;TR:0)`. This baseline model uses the failure and repair rates from Cao's original model to parameterize the CTBN and also makes use of the costs from Table I. An alternate selection for TR, WT or EL is considered a deviation from the baseline, and will result in changes to the system performance including failure rate, repair rate and cost. The factors by which these values change from the baseline for each component are shown in Table II.

Note that for each of the three components, assigning an alternative of A_0 results in a factor change of 1.0 for failure rate, repair rate and cost. This is because we have chosen A_0 as the baseline, and therefore expect no change in the values associated with these components. Recall that cost is broken down by production cost and repair cost, as shown in Table I. In this work, the factors provided in the cost column simultaneously change both cost categories. These rates are intended to demonstrate one possible scenario, and more complex interactions can be defined using this framework if necessary. To illustrate, an assignment of `(WT:2;EL:0;TR:0)` means that the failure and repair rates for node WT are multiplied by a factor of 0.30 and 1.60, respectively. Furthermore, the production and repair costs associated with node WT are increased by a factor of 2.20. Neither TR nor EL are altered, because they correspond to the baseline.

The factored nature of CTBNs enables the system behavior to be specified on a component level. This can be extremely advantageous, especially when there are a large number of design alternatives. Let D be the set of components that can be modified when designing a system, and let A_c be the set of alternatives associated with some component C. The number of possible design alternatives n for a system is as follows:

$$n = \prod_{c \in D} A_c.$$

In the case of the vehicle model with two, three and four alternatives for the TR, WT and EL components, $n = 24$. This means that we must specify 23 different modifications to the baseline, as compared to the 6 modifications to the baseline provided in Table II. As models become more complex, the exponential growth of n makes the factored representation even more advantageous.

With the CTBN vehicle model and the PBL objectives defined, we must specify the PBL objectives in terms of CTBN performance functions. The `Cost` performance function is computed by summing the fixed production costs for each component, along with the repair costs associated with a component transitioning from a failing to operational state. These costs are specified in Table I but may be modified according to the design alternative. For instance, if the baseline alternative `(WT:0;EL:0;TR:0)` is chosen and the CTBN dynamics indicate that the WT component is expected to fail twice within the period of interest, then the `Cost` performance value will be the sum of the production costs (\$87500) along with the cost for repairs (\$700 $\times$ 2), for a total of \$88900. The `Availability` performance function can be computed by querying the CTBN model to determine the expected portion of time that the Vehicle node spends in the operational state. Finally, the `Maintainability` performance function can be obtained by simulating repairs using the CTBN model and counting the number of instances where repair time did not exceed the specified threshold τ. This count is divided by the total number of simulated repairs, providing an estimate of the vehicle's maintainability. Although these performance functions are capable of capturing complex performance objectives, the factored representation greatly simplifies the specification of these objectives.

To determine which vehicle designs are valid candidates according to the PBL contract, the `Cost`, `Availability`, and `Maintainability` performance functions were evaluated for each of the 24 design alternatives. This was achieved by first assigning the appropriate alternatives to WT, EL and TR and then querying the CTBN model for each of the three performance values, as described previously. These performance values are presented in the third, fourth and fifth columns of Table III. If we assume the PBL contract thresholds require a maximum cost of $T_C = \$950000$, a minimum availability of $T_A = 0.7$, and a minimum maintainability of $T_M = 0.65$, then a subset of ineligible design alternatives can be eliminated. The set of remaining alternatives $R_{ALT} = \{A, C, E, F, G, H, I, O, Q, R, U, W, X\}$ appear in boldfaced type in Table III.

Although each of the alternatives in R_{ALT} meet the technical requirements of the contract, there are still 13 remaining options, as shown in Table IV. Faced with this decision, it may be tempting to select one of the valid design alternatives arbitrarily; however, this may result in the selection of a sub-optimal solution. To aid in the decision process, design alternatives can be optimized over the performance values obtained by the CTBN model. For instance, in a scenario where `Cost` is the motivating factor, finding the optimal vehicle design alternative is as simple as identifying the minimum cost value. In that case, the set of desirable design alternatives is reduced to a single element, $P_C = \{C\}$, as illustrated by the shading in the P_C column of Table IV. Note that prior to the elimination of the ineligible design alternatives, K had the lowest cost overall. However, K failed to meet the specified thresholds for `Availability` and `Maintainability`, likely due to the fact that minimizing cost often conflicts with maximizing availability and maintainability. While C is the remaining design alternative in R_{ALT} with the lowest cost, given that our objectives include factors beyond `Cost` alone, it is not sufficient to simply choose the least expensive

TABLE III. PERFORMANCE FUNCTION EVALUATIONS FOR THE VEHICLE MODEL IN FIGURE 2. DESIGN ALTERNATIVES IN BOLDFACED TYPE ARE MEMBERS OF THE SET R_{ALT}, WHICH ARE THE ELIGIBLE ALTERNATIVES UNDER THE GIVEN PBL CONTRACT.

ID	Design Alternative	Cost (in USD)	Availability	Maintainability
A	**(WT:0;EL:0;TR:0)**	799670	0.7041	0.7180
B	(WT:0;EL:0;TR:1)	862435	0.6988	0.6687
C	**(WT:0;EL:1;TR:0)**	752623	0.7018	0.6757
D	(WT:0;EL:1;TR:1)	815444	0.6938	0.6277
E	**(WT:0;EL:2;TR:0)**	886081	0.7087	0.7387
F	**(WT:0;EL:2;TR:1)**	949198	0.7044	0.6857
G	**(WT:0;EL:3;TR:0)**	822286	0.7002	0.7462
H	**(WT:0;EL:3;TR:1)**	885229	0.7055	0.6995
I	**(WT:1;EL:0;TR:0)**	778105	0.7040	0.6686
J	(WT:1;EL:0;TR:1)	841201	0.7008	0.6332
K	(WT:1;EL:1;TR:0)	730955	0.6926	0.6311
L	(WT:1;EL:1;TR:1)	794057	0.7034	0.6021
M	(WT:1;EL:2;TR:0)	864635	0.6989	0.6796
N	(WT:1;EL:2;TR:1)	927874	0.7076	0.6411
O	**(WT:1;EL:3;TR:0)**	800818	0.7000	0.6938
P	(WT:1;EL:3;TR:1)	863833	0.6981	0.6529
Q	**(WT:2;EL:0;TR:0)**	842648	0.7054	0.7322
R	**(WT:2;EL:0;TR:1)**	905856	0.7051	0.6751
S	(WT:2;EL:1;TR:0)	795560	0.6999	0.6805
T	(WT:2;EL:1;TR:1)	858806	0.6991	0.6313
U	**(WT:2;EL:2;TR:0)**	929063	0.7006	0.7435
V	(WT:2;EL:2;TR:1)	992094	0.7049	0.6890
W	**(WT:2;EL:3;TR:0)**	865252	0.7008	0.7543
X	**(WT:2;EL:3;TR:1)**	928041	0.7001	0.7022

alternative.

A superior method would incorporate an additional performance goal, such as `Availability`. To optimize over both `Cost` and `Availability` simultaneously, a Pareto frontier is employed. The Pareto frontier is constructed by first selecting C, the alternative that minimizes `Cost`. Alternatives are then added to the frontier that maximize `Availability`, without being strictly dominated by an element already contained in the set. In this context, an alternative is dominated if another alternative has a lower cost and higher availability. For this scenario, the resulting Pareto frontier is the set of optimal design alternatives $P_{CA} = \{A, C, E, I, Q\}$. The design alternatives belonging to set P_{CA} are indicated by the shading in the respective column of Table IV. Figure 3 shows each of the 13 vehicle design alternatives, plotted as a function of `Cost` and `Availability`. The members of the Pareto frontier are denoted with red diamond markers, labeled with their respective ID values. The remaining design alternatives are depicted using blue circular markers. Note that no datapoint occurs above and to the left of a point in the Pareto frontier, which is in contrast to all elements in the dominated set. Any member of the set P_{CA} is a valid solution when considering the objectives `Cost` and `Availability`.

The benefit of identifying the Pareto frontier is the elimination of the dominated design alternatives, which represent the objectively inferior choices in terms of performance values. For instance, when compared to A, design alternative B is both more expensive and less available. Selection of an appropriate design alternative will depend on the relative importance of each objective, as determined by the client or contractor, as well as other system specifications. The design alternatives contained within the Pareto frontier can be further refined by considering additional specifications that are not based strictly on performance measures. By way of example, alternatives A and Q are both members of the Pareto frontier and therefore neither one can be considered strictly better than the other when considering `Cost` and `Availability` performance objectives alone. To make an informed choice, the client or contractor must determine whether the improved availability obtained by selecting Q is worth the increase in cost. Furthermore, there may be substantive differences between alternatives A and Q that are not captured by the specified performance objectives, such as the presence of underbody armor, which may be a requirement of the contract. In the event that all members of the Pareto frontier fail to meet the additional requirements specified by the PBL contract, a viable design alternative can be selected from the dominated solutions such that the distance to the frontier is minimized.

TABLE IV. THE ELIGIBLE ALTERNATIVES R_{ALT} UNDER THE GIVEN PBL CONTRACT. SHADING IN THE FINAL THREE COLUMNS IS USED TO INDICATE THAT THE DESIGN ALTERNATIVE IS A MEMBER OF THAT SET. P_C IS THE SET OF ALTERNATIVES WHERE COST IS THE SOLE OBJECTIVE BEING OPTIMIZED, P_{CA} IS THE SET WHERE COST AND AVAILABILITY ARE BEING OPTIMIZED SIMULTANEOUSLY, AND P_{CAM} OPTIMIZES OVER ALL THREE OBJECTIVES.

ID	Design Alternative	$\mathbf{P_C}$	$\mathbf{P_{CA}}$	$\mathbf{P_{CAM}}$
A	**(WT:0;EL:0;TR:0)**			
C	**(WT:0;EL:1;TR:0)**			
E	**(WT:0;EL:2;TR:0)**			
F	**(WT:0;EL:2;TR:1)**			
G	**(WT:0;EL:3;TR:0)**			
H	**(WT:0;EL:3;TR:1)**			
I	**(WT:1;EL:0;TR:0)**			
O	**(WT:1;EL:3;TR:0)**			
Q	**(WT:2;EL:0;TR:0)**			
R	**(WT:2;EL:0;TR:1)**			
U	**(WT:2;EL:2;TR:0)**			
W	**(WT:2;EL:3;TR:0)**			
X	**(WT:2;EL:3;TR:1)**			

To conclude the Vehicle Model performance evaluation, a third performance objective, `Maintainability`, is now considered during the optimization process in addition to `Cost` and `Availability`. Again the Pareto frontier is utilized, although in this case the design alternatives are now evaluated along a third axis corresponding to `Maintainability`. Due to the difficulties in plotting a three-dimensional graph in two-dimensional space, we omit the graph, however, the set $P_{CAM} = \{A, C, E, G, H, I, Q, W\}$ making up the Pareto frontier is shown via the shading in the final column of Table IV. Again, note that B is dominated by design alternative A with respect to `Cost`, `Availability` and `Maintainability`. The set P_{CAM} represents the optimal choices of vehicle design alternatives. Compare this to the 13 design alternatives prior to optimization, and it becomes apparent that five of the choices would have been objectively inferior. Without performing optimization explicitly, it is difficult to identify which alternatives are superior, a task which becomes even more difficult as additional objectives and alternatives are considered.

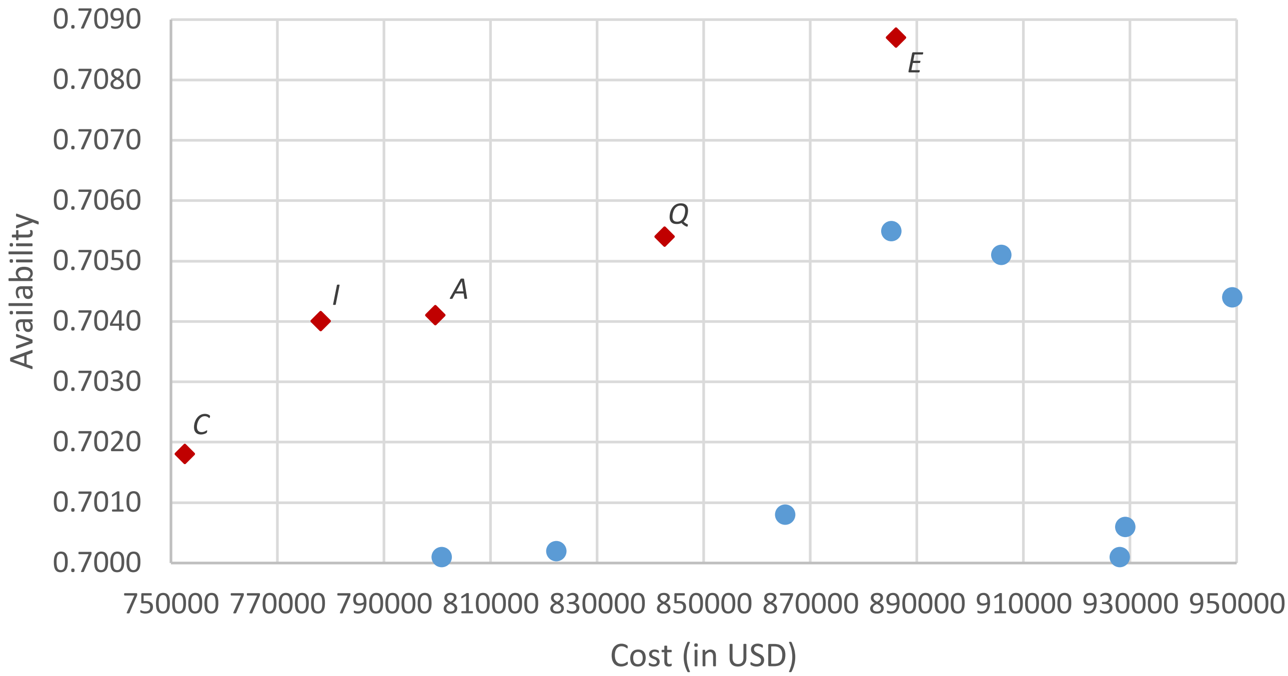

Fig. 3. Pareto front for the vehicle optimization problem considering cost and reliability objectives

VI. PHM Applications

In addition to providing a method for evaluating and optimizing PBL objectives, the CTBN models introduced in this work have additional benefits to clients beyond the design stage. These models can also be used to determine the optimal maintenance crew assignment, or other decisions related to the use and maintenance of a system [22]. Aside from aiding in the decision process, these models have demonstrated utility in diagnostic and prognostic contexts, allowing for queries that identify the probability of failures through time. CTBNs have been applied successfully to diagnostic and prognostic tasks in the past [8]. This is in part due to their natural ability to represent transition behavior in terms of common time to failure distributions [23]. By predicting failures before they occur, preparations can be made or preventative maintenance can be employed to mitigate the impact of these failures. Such strategies can also be incorporated into the analysis when evaluating maintenance and diagnostic alternatives.

By way of example, refer back to the vehicle model shown in Figure 2. After the completion of the PBL contract, a design is chosen that provides a fixed value for the control nodes C_{WT}, C_{EL} and C_{TR}. At this point, the model can be simplified by removing these control nodes and allowing their children in the graph to transition according to the chosen design specifications. This results in a network structure that once again conforms to the one shown in Figure 1. Using existing design information, tests and effects can be added to the model automatically. This model is now suitable for performing diagnostics and prognostics. When tests are performed, their outcomes are observed and recorded. With this information, the CTBN is able to determine the expected probability of each subsystem failing at any point in time. For instance, assume there are three tests added to the vehicle model, and test T_1 passes, T_2 fails, and T_3 has not yet been run. Querying this model might indicate that the brakes subsystem has a 4% chance of failing after another 100 hours of use, or that the expected time to failure for the wheels and tires is 230 hours. This type of information can be extremely valuable, and may allow time to order parts, perform preventative maintenance, or otherwise make arrangements to prepare for likely events.

VII. Conclusion

In this work, we have presented a novel method for evaluating PBL contract objectives using CTBN performance functions. This is achieved by using a CTBN that models the system of interest, augmented with control nodes, to compactly represent competing design decisions. In CTBNs, a performance function provides an application-specific method for assigning a reward or penalty value to expected system dynamics. A design alternative can be evaluated by querying the model over the performance functions specified for each PBL objective. The resulting values are then compared to the requirements defined by the PBL contract, and design alternatives that do not meet the technical objectives are eliminated from the pool of possible choices. The remaining design alternatives represent the eligible solutions according to the contract requirements.

We also demonstrated that the set of eligible solutions can be refined further by optimizing over the performance functions. In this context, optimization minimizes or maximizes performance values as a function of design alternatives. To model these design alternatives in the CTBN framework, we make use of control nodes, which represent design alternatives at the component level. Optimization is then achieved by enumerating the possible design alternatives, evaluating the performance functions using the CTBN model, and choosing the alternative that best meets the needs of the contract. In the event that multiple performance objectives are under consideration, the optimization process returns a set of non-dominated solutions, referred to as the Pareto frontier. The elimination of dominated solutions prevents the selection of an objectively inferior design that may meet the technical requirements of the PBL contract, but is a sub-optimal choice with respect to at least one of the other design alternatives. Having further narrowed the list of viable candidates using optimization, the task of making a final selection is reduced to identifying the combination of performance values that best capture the needs laid out by the contract. The relative importance of each objective is largely application-specific, and therefore the decision is best made by a domain expert.

The vehicle design problem presented in this work is intended to be an illustrative example, and although it demonstrates the feasibility of our proposed approach, it does not fully exploit the representational power afforded by the CTBN framework. Real-world systems are often highly complex, consisting of many interdependent sub-systems. For example, the Vehicle Model used throughout this work represents the entire electrical sub-system using a single node. A more accurate representation could further decompose the EL node into components such as the battery and alternator. The level of detail required to model a system of interest is domain-dependent, and should be considered on a case-by-case basis. As a reminder, there exist automated techniques for developing these models using existing design information such as fault-trees and D-matrices. The method presented in this work is agnostic with respect to the details of the model, and we expect this approach to work for any practical application of PBL.

In addition to more complex models, our approach also allows for the specification of performance functions of greater complexity. The example vehicle used in this work defined the TCO using initial production cost and estimated costs for repairs. However, a client may also find it beneficial to include the disposal costs associated with the design alternative at the end of its useful life, or the costs associated with wages for maintenance technicians. The design alternatives presented here were assumed to influence both production and repair costs equally. Instead, it may be more accurate to define a mapping such that the design alternative could uniquely influence each of the cost sub-categories. With respect to availability, we made use of a definition that is well-established in the PHM literature, but may prove overly simplistic for some applications. Specifically, availability is defined as the portion of time that a system is expected to remain in the operational state, which is the case when the system meets all functional specifications. Note that a system is considered inoperable if any of its required functions cannot be performed, however there may be value in the ability to perform even a subset of the required functions. A military vehicle with a non-functional weapons system can still perform tasks related to transport, which still has mission value. For this reason, it may be useful to consider the case where a system is partially operational, as distinguished from fully operational or failing. Partial performance can be awarded to the partially operational state to account for the degradation from fully operational while still considering the improvement over total failure. Finally, performance functions could be added or removed to match the PBL goals for each application.

One of the primary strengths of the CTBN framework is its ability to model system dynamics over time. In this work, we defined performance functions that computed expected values using the simulations performed by a CTBN model. Although this does capture important information about the system, it does not fully take advantage of the available temporal information. One way to exploit the temporal nature of the model is to define performance functions that increase or decrease in value the longer a variable remains in a state. For example, a vehicle may be assigned negative performance for entering a failing state, along with an additional penalty that increases every hour until it is repaired. In addition, given that degradation is expected to occur over time, performance functions could be designed to account for the age of a system. When dealing with functions like availability, this would prevent the underestimation of values at earlier stages and overestimation near the end of the lifecycle.

Finally, more complex models and performance functions can be constructed that take into account multiple, potentially distinct systems. For instance, an agency may have a variety of vehicles and equipment at its disposal. The system dynamics for each of these resources may vary, but by combining their distinct models, it is possible to construct performance functions that span these resources. This would provide a more accurate value for large-scale mission operations, which may require a diverse set of resources.

VIII. Future Work

In the vehicle design problem presented in this work, there were three components with two, three and four design choices. This results in a total of 24 possible vehicle design alternatives, each of which were evaluated using the CTBN model and compared to one another for optimization purposes. Although this approach provides a method for identifying the optimal design alternatives, the process of completely enumerating every possible alternative may become infeasible as the complexity of the model grows. For instance, consider the case where a system has ten components, each of which have three possible design alternatives. In this case, there are $3^{10} = 59049$ possible design alternatives for the system, each of which will generally have a unique performance evaluation.

In cases where the number of alternatives is too large to enumerate explicitly, we instead must rely on heuristics to evaluate only a subset of the possible choices. In future work, we intend to apply methods such as hill climbing, genetic algorithms (GAs), or particle swarm optimization (PSO), which are all algorithms that attempt to find optimal values without completely covering the search space. This is achieved by using previous evaluations to make an informed choice about which alternative to consider next. In this way, we could approximate the Pareto frontier, and upon completion of the algorithm, any alternatives that were not evaluated explicitly would be considered a member of the dominated set.

Another interesting research avenue that we hope to pursue is the notion of hierarchical performance functions. Recall that a single `Cost` performance function was constructed that incorporated two unique sub-categories: initial production cost and repair costs. Furthermore, it was noted that other cost categories may be useful for other applications as well. Rather than creating a single performance function that directly captures these sub-categories, it may make more sense to treat each one as an independent performance function that can be computed directly. If desired, a single performance function could then be defined in terms of these other performance functions, thus creating a hierarchy of functions. In this way, it would be possible to evaluate the TCO for a system, or instead evaluate on the repair cost, depending on the needs of the problem. These performance functions could be nested many levels deep, which could provide a manageable way of modeling complex goals.

Acknowledgments

This paper was developed in part due to work that was performed under the NASA STTR T13.01-9887 contract. Without the support and guidance from our contacts at NASA and the people at Qualtech Systems, Inc, this work would not have been possible. We would also like to thank the members of the Numerical Intelligent Systems Laboratory (NISL) at Montana State University for their input during the development stages of this paper.

References

[1] U. Nodelman, C. R. Shelton, and D. Koller, "Continuous time Bayesian networks," in *Proceedings of the Eighteenth conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc., 2002, pp. 378–387.

[2] L. Sturlaugson and J. Sheppard, "Factored performance functions with structural representation in continuous time Bayesian networks," in *Proceedings of the Florida Artificial Intelligence Symposium*, May 2014, pp. 512–517.

[3] ACC, *Performance Based Logistics Community of Practice*. Defense Acquisition University, 2016. [Online]. Available: https://acc.dau.mil/CommunityBrowser.aspx?id=527126

[4] A. Sols, D. Nowick, and D. Verma, "Defining the fundamental framework of an effective performance-based logistics (PBL) contract," *Engineering Management Journal*, vol. 19, no. 2, pp. 40–50, 2007.

[5] U. D. Kumar, D. Nowicki, J. E. Ramirez-Marquez, and D. Verma, "A goal programming model for optimizing reliability, maintainability and supportability under performance based logistics," *International Journal of Reliability, Quality and Safety Engineering*, vol. 14, no. 03, pp. 251–261, 2007.

[6] U. Nodelman and E. Horvitz, "Continuous time Bayesian networks for inferring users' presence and activities with extensions for modeling and evaluation," *Microsoft Research Technical Report*, 2003, MSR-TR-2003-97.

[7] E. Acerbi and F. Stella, "Continuous time Bayesian networks for gene network reconstruction: a comparative study on time course data," in *International Symposium on Bioinformatics Research and Applications*. Springer, 2014, pp. 176–187.

[8] D. Cao, "Novel models and algorithms for systems reliability modeling and optimization," Ph.D. dissertation, Wayne State University, 2011.

[9] U. Nodelman, C. R. Shelton, and D. Koller, "Learning continuous time Bayesian networks," in *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc., 2002, pp. 451–458.

[10] U. Nodelman, C. R. Shelton, and D. Koller, "Expectation maximization and complex duration distributions for continuous time Bayesian networks," in *Proceedings of the Twenty-First International Conference on Uncertainty in Artificial Intelligence*, 2005, pp. 421–430.

[11] L. Perreault, M. Thornton, S. Strasser, and J. Sheppard, "Deriving prognostic continuous time Bayesian networks from D-matrices," in *IEEE AUTOTESTCON Conference Record*, 2015.

[12] L. Perreault, M. Thornton, and J. Sheppard, "Deriving prognostic continuous time Bayesian networks from fault trees," *Prognostics and Health Management Society*, 2016, (Submitted).

[13] Y. Fan, J. Xu, and C. R. Shelton, "Importance sampling for continuous time Bayesian networks," *Journal of Machine Learning Research*, vol. 11, no. Aug, pp. 2115–2140, 2010.

[14] I. Cohn, T. El-Hay, N. Friedman, and R. Kupferman, "Mean field variational approximation for continuous-time Bayesian networks," *Reasoning about Structured Stochastic Systems in Continuous-Time*, vol. 11, p. 2744, 2011.

[15] R. T. Marler and J. S. Arora, "Survey of multi-objective optimization methods for engineering," *Structural and Multidisciplinary Optimization*, vol. 26, no. 6, pp. 369–395, 2004.

[16] K. Doerr, I. Lewis, and D. R. Eaton, "Measurement issues in performance-based logistics," *Journal of Public Procurement*, vol. 5, no. 2, pp. 164–186, 2005.

[17] D. Villanueva Jaquez, "Multiple objective optimization of performance based logistics," Ph.D. dissertation, University of Texas at El Paso, 2009.

[18] "High mobility multipurpose wheeled vehicle (hmmwv)," http://www.army-guide.com/eng/product2759.html, accessed: 2016-07-02.

[19] "Steep cost of military vehicles outlined in army report," http://www.cnn.com/2011/US/01/27/army.vehicle.costs/, accessed: 2016-07-02.

[20] Department of Defense [DOD], "Product support: A program managers guide to buying performance." November 2001.

[21] C. E. Ebeling, *An Introduction to Reliability and Maintainability Engineering*. McGraw-Hill, 1997.

[22] L. Sturlaugson, L. Perreault, and J. Sheppard, "Factored performance functions and decision making in continuous time Bayesian networks," *International Journal of Applied Logic*, 2016, (To appear).

[23] L. Perreault, M. Thornton, R. Goodman, and J. W. Sheppard, "A swarm-based approach to learning phase-type distributions for continuous time Bayesian networks," in *Proceedings of the IEEE Swarm Intelligence Symposium (SIS)*, December 2015, pp. 1860–1867.