Investigating Matureness of Probabilistic Graphical Models for Dry-Bulk Shipping

Nils Finke[1,2] FINKE@IFIS.UNI-LUEBECK.COM
Marcel Gehrke[1] GEHRKE@IFIS.UNI-LUEBECK.COM
Tanya Braun[1] BRAUN@IFIS.UNI-LUEBECK.COM
Tristan Potten[1] TRISTAN.POTTEN@STUDENT.UNI-LUEBECK.COM
Ralf Möller[1] MOELLER@IFIS.UNI-LUEBECK.COM

Institute of Information Systems, University of Lübeck, Lübeck, Germany[1]

Oldendorff Carriers GmbH & Co. KG., Lübeck, Germany[2]

Abstract

Dry-bulk shipping is crucial for a functioning global trade economy. Thus, additional research is highly relevant to further improve bulk shipping operations. Dry-bulk shipping involves many entities interacting with each other in an uncertain environment that changes over time. To assist dry-bulk vessel operators in how to position their vessels, efficient query answering and decision support is necessary. Therefore, we investigate existing modelling formalism and inference algorithms regarding which aspects of dry-bulk shipping are already realisable. Although not all challenges are already well-understood, we show that a lifted dynamic approach tackles most of the challenges involved in handling dry-bulk shipping.

Keywords: MEU; Relational Temporal Probabilistic Models; Lifting; Application

1. Introduction

With the growth of the world economy, the need for seaborne transportation of dry-bulk commodities has increased over the years. Dry-bulk commodities are non-liquid, unpacked goods that have undergone little to no processing, like iron ore, coal, or grains. Typically, these commodities are transported for further processing from their mining regions to different places on the globe. In contrast to liner shipping, where individual containers can be rented on a fixed route with fixed timetables, it is common in bulk shipping to transport bulk goods by order of one client on individual routes. Companies operating vessels and cargo suppliers negotiate contracts individually. The dry-bulk shipping industry is closely related to the commodity industry. Commodity trade flows emerge based on export and import as natural resources (commodities) are unevenly distributed over the earth. Vessels perform voyages, representing these trade flows, by transporting cargo from export nations to import nations. As such, dry-bulk shipping has a strong geospatial component: Vessels travel between ports, and ports are in countries, which are on continents. However, trade flows are not being evaluated based on landmass. Instead, they are analyzed from an ocean-centric perspective by splitting the world into different *regions*, e.g., West- and East-Australia or North-, Mid-, and South-Pacific. To give another example: The ports Vancouver and Seven Islands are both in Canada, but one is located on the west coast (Pacific) and the other on the east coast (Atlantic).

For the transportation of cargoes, a fee per ton, called freight rate, is charged. Freight rates are negotiated between shipping companies and cargo suppliers. Thus, freight rates are priced dynamically, influenced by conditions in regions. Operators try to position their vessels in regions with high export volume of bulk commodities to benefit from good freight rates (with good meaning high rates). It is less likely to find cargo-contracts for bulk vessels in typical import regions. In case cargo-contracts are available in import regions, lower freight rates are normally paid due to an oversupply of bulk vessels in these regions (which is common as high demand for goods equals many vessels discharging, thus, a lot of unladen vessels being on site with the need for further business). Hence, higher freight rates are paid to vessel owners for transporting goods to regions with high demand to compensate for lower freight rates on the way back to the origin. Another challenge for a vessel owner in positioning their vessels arises if export regions become supersaturated if too many vessels are on site. Therefore, operators on the one hand need to keep track of export and import behavior of countries and on the other hand of other vessels being operated.

The dry-bulk shipping market is a volatile environment characterized by uncertainty as decisions of participators in the industry, like cargo suppliers and owners of vessels, are not transparent to the market and are driven by each participator's own belief on how the market is evolving. From the perspective of a company operating vessels on the market, it is of high interest to know how other operators position their fleet, how trade flows will change over time, and how to position its ships to maximize own profits given all this uncertainty.

From a theoretical perspective, the dry-bulk shipping market contains various entities (shipping agents, vessels, regions) in relation to each other and requires reasoning under uncertainty, specifically for providing decision support. As such, we are looking at formalisms and algorithms from the field of statistical relational AI (StaRAI). Within StaRAI, we consider dynamic, probabilistic modeling techniques as a mechanism to enable vessel operators making more informed decisions w.r.t. to the constantly changing market environment. Probabilistic models allow for representing complex domains using probability distributions over a set of random variables (randvars).

The contributions of this paper are:

(i) A comparison of probabilistic formalisms for modeling the dry-bulk shipping market.
(ii) A parameterized probabilistic dynamic decision model (PDDecM) of the dry-bulk shipping market from the perspective of a vessel operator.
(iii) An analysis of the matureness of decision support in the dry-bulk shipping PDDecM using an algorithm called meuLDJT (Gehrke et al., 2019).

A short empirical evaluation looks at available implementations to answer queries on the PDDecM. In the course of the paper, we present which aspects of dry-bulk shipping are already captured with existing models and where challenges still exist. Specifically, we present three challenges, namely, non-stationary processes, liftability, and a more expressive query language.

The remainder of this paper has the following structure: We begin by highlighting the characteristics of dry-bulk shipping and derive requirements on probabilistic models, followed by an evaluation of commonly used probabilistic models which meet our requirements. We show that PDDecMs and meuLDJT meet most of our requirements. We conclude with some ongoing challenges that need to be addressed to bring formalisms and algorithms even closer to real-world applications.

2. Requirements

Efficient query answering and decision support depends on the expressiveness of the model and the efficiency of inference algorithms. We first motivate characteristics of our setting and investigate the fit of common probabilistic models.

2.1 Model Characteristics

We generalize the dynamics of the dry-bulk shipping market as follows: (i) Vessels are operated by agents, (ii) performing voyages between different zones on the earth, (iii) for the transportation of various commodities, (iv) depending on supply and demand, (v) both changing over time. Vessels, agents, zones, and commodities can be seen as entities, which stand in relation to each other, e.g., vessels are *operated by* agents. Generally speaking, we face a probabilistic, relational scenario with a temporal component. Additionally, a large domain size with already over ten thousand dry-bulk vessels as per year 2019 (UNCTAD, 2019) requires for models with compact representations.

Actions and decisions of players on the market are not fully transparent. Thus, every participant's decisions are driven by their own belief on how the market is evolving. A single operating agent, who only has a very limited view on the market, is confronted with various questions about market conditions: (i) For which commodities remains a need for transportation in a zone? (ii) How is demand for commodities affected given a set of vessels already being in a zone? (iii) Are freight rates high in a specific zone? (iv) Which results are expected w.r.t. to the payment of freight rates given actions on how to position vessels? The questions lead to queries given a probabilistic model regarding probabilities of events, probability distributions, and actions leading to a maximum expected utility, MEU for short. Thus, probabilistic inference with different types of queries is required such as computing prior and posterior marginals and solving MEU problems.

Further, being able to interpret decisions or calculations of inference algorithms is important, e.g., which actions or entities have the highest influence on an MEU. Explainable decision support is essential as human expertise still need to be put into consideration as not every aspect of the dry-bulk shipping market can be measured or specified in a model. To summarize, the following characteristics and requirements have to be considered when setting up an appropriate model: (a) *probabilistic, relational nature*, (b) *temporal behavior*, (c) *efficient marginal query answering*, (d) *sequential decision support*, and (e) *compact representation (large domain)*. Next, we benchmark existing models and inference algorithms w.r.t. these characteristics and requirements.

2.2 Benchmarking Existing Approaches

Probabilistic graphical models allow for representing relational models using probability distributions and exploiting independencies between randvars. More specifically, temporal behavior within an uncertain environment is commonly described in the propositional case by dynamic probabilistic graphical models, like dynamic Bayesian networks/factor graphs/Markov networks or by partially observable Markov decision processes (POMDP). In the relational case, there exist counterparts such as relational DBNs (Sanghai et al., 2005), parameterised factor graphs (Poole, 2003), Markov logic networks (Richardson and Domingos, 2006), or first-order POMDPs.

POMDPs generalize Markov decision processes by maintaining a probability distribution over the set of all possible states. Solving a POMDP means calculating a policy that specifies an optimal action to be performed for each state an agent believes to be in. The optimal action maximizes the expected reward of the agent over a defined horizon. As such, POMDPs allow for sequential decision support (Item d) in probabilistic scenarios (Item a), with temporal behavior implicitly contained in POMDPs (Item b). But, providing a policy as solution does not allow for further marginal queries (Item c). Additionally, large domain sizes (Item e) are not addressed. The first-order variant of POMDPs builds on logic, allowing problems to be represented in a more compact manner (Wang and Schmolze, 2005; Sanner and Kersting, 2010), incorporating Item e. Still, algorithms to solve POMDPs are offline, meaning that best actions for all possible states are calculated prior to execution in the form of a policy (Ross et al., 2008), which does not allow for marginal queries.

Therefore, we turn to dynamic probabilistic graphical models. Dynamic models have one model (Bayesian net, factor graph, Markov net) copied for each timesteps with randvars of a timestep t connected to randvars of the previous timestep $t-1$. They represent the state space in a factored form, with Bayesian nets being a directed formalism and the other two undirected ones. Transferred to the bulk-shipping scenario, each randvar describes an instance, or more specifically, the behavior of an entity at any given timestep t such as a specific vessel being operated by a particular agent, the amount of cargo needed to be transported out of a specific zone, or the level of a freight rate in a zone. Such formalisms represent probabilistic, relational scenarios including temporal behavior (Items a and b). Variable elimination (VE) (Zhang and Poole, 1994) is an algorithm for exact inference enabling query answering (Item c) in them. Still, the modeling formalisms do not cater for sequential decision making (Item d) and do not handle large domain sizes very well (Item e).

Probability and utility theory make up decision theory, in which probabilistic models are extended with action and utility nodes. The MEU principle says that a rational agent should choose an action in the current state that maximizes its expected utility based on the probability of reaching an outcome state. A utility function is used to rate the desirability of states. An agent selects those actions over time, triggering transitions from a current state to the next state, which lead to the maximum expected utility. But these models still do not cater for large domains (Item e) and in general, inference in them is intractable (Koller and Friedman, 2009), which becomes especially apparent with large domains, yielding models with many randvars. Enumerating all entities using randvars is impractical with thousands of vessels operating in multiple zones. Lifted inference approaches reduce computational work as inference is performed using representatives for sets of indistinguishable randvars, which allows for tractable inference w.r.t. domain sizes (Niepert and Van den Broeck, 2014). PDDecMs provide a formalism based on factor graphs to compactly represent a probabilistic scenario with many entities in relation to each other and include actions and utilities to support decision making (Gehrke et al., 2019). Markov logic networks (MLNs) (Richardson and Domingos, 2006) represent another form of probabilistic relational model fulfilling Items a, c, and e. In contrast to PDDecMs, MLNs use weighted rules. There exist a dynamic variant (Kersting et al., 2009) and a decision-theoretic variant (Nath and Domingos, 2009) but no combination of both and therefore, no inference algorithm handling both aspects in a concerted effort. Additionally, for now, we do not want to impose any structure on the relationship between entities by using rules, where factor-based formalisms allow for keeping the relationship more vague. For PDDecMs, meuLDJT (Gehrke et al., 2019) efficiently solves lifted MEU problems, while also answering marginal queries. Together, PDDecMs and meuLDJT fulfill our requirements (Items a to e).

In the following sections, we set up a PDDecM for dry-bulk shipping and examine meuLDJT for answering questions as given above.

3. A Formal Model of Dry-bulk Shipping

In this section, we build a formal model of dry-bulk shipping step by step. The model itself addresses the core fundamentals of the dynamics of the dry-bulk shipping market, as motivated in Section 1, focusing on vessels and their movements (ignoring agents and different commodities for now). We start with a static model. Based on the static model, we specify a dynamic model. Taking the dynamic model, we introduce actions and utilities to finish modeling dry-bulk shipping.

3.1 Parameterized Probabilistic Model

The first step of setting up the dry-bulk shipping model is specifying a static model in the form of a parameterized probabilistic model (PM). The PM will describe dry-bulk shipping without the temporal and decision-theoretic components, encoding entities like vessels and zones and relations between them. The definition of a PM has its roots in work by Poole (2003) and Taghipour et al. (2013a) and is mainly based on the work by Braun and Möller (2016).

PMs are based on factor graphs, using logical variables (logvars) as parameters for randvars (parameterized randvar, PRV for short). PRVs compactly represent sets of randvars that are considered indistinguishable without further evidence. Logvars have domains that contain constants. For certain scenarios, one might wish to restrict logvars to certain constants, which is why PRVs are often accompanied by a constraint indicating admissible constants. Constraints act as an abstraction for, e.g., instances stored in a database such as vessels. PRVs are combined using parametric factors (parfactors) to represent relations. A parfactor describes a factor, mapping argument values to real values (*potentials*), of which at least one is non-zero. In the following, we formally define PMs.

Definition 1 *Let $\mathbf{R}$ be a set of randvar names, $\mathbf{L}$ a set of logvar names, Φ a set of factor names, and $\mathbf{D}$ a set of constants. All sets are finite. Each logvar L has a domain $\mathcal{D}(L) \subseteq \mathbf{D}$. A* constraint *is a tuple $(\mathcal{X}, C_{\mathcal{X}})$ of a sequence of logvars $\mathcal{X} = (X^1, \dots, X^n)$ and a set $C_{\mathcal{X}} \subseteq \times_{i=1}^{n} \mathcal{D}(X^i)$. The symbol $\top$ for C marks that no restrictions apply, i.e., $C_{\mathcal{X}} = \times_{i=1}^{n} \mathcal{D}(X^i)$. A* PRV *$R(L^1, \dots, L^n), n \geq 0$ is a construct of a randvar $R \in \mathbf{R}$ possibly combined with logvars $L^1, \dots, L^n \in \mathbf{L}$. If $n = 0$, the PRV is parameterless and forms a propositional randvar. The term $\mathcal{R}(A)$ denotes the possible values (range) of a PRV A. An* event *$A = a$ denotes the occurrence of PRV A with range value $a \in \mathcal{R}(A)$. We denote a* parfactor *g by $\phi(\mathcal{A})_{|C}$ with $\mathcal{A} = (A^1, \dots, A^n)$ a sequence of PRVs, $\phi : \times_{i=1}^{n} \mathcal{R}(A^i) \mapsto \mathbb{R}^+$ a function with name $\phi \in \Phi$, and C a constraint on the logvars of $\mathcal{A}$. A PRV A or logvar L under constraint C is given by $A_{|C}$ or $L_{|C}$, respectively. We may omit $|\top$ in $A_{|\top}$, $L_{|\top}$, or $\phi(\mathcal{A})_{|\top}$. The term $gr(P)$ denotes the set of all instances of P w.r.t. given constraints. An instance is an instantiation (grounding) of P, substituting the logvars in P with a set of constants from given constraints. A* PM *G is a set of parfactors $\{g^i\}_{i=1}^{n}$, representing the full joint distribution $P_G = \frac{1}{Z} \prod_{f \in gr(G)} f$ with Z as normalizing constant.*

Given a parfactor $\phi(\mathcal{A})_{|C}$, ϕ is identical for the propositional randvars in $gr(\mathcal{A}_{|C})$, compactly encoding a model with recurring structures. Figure 1 shows a graphical depiction of a PM. Variable nodes (ellipses) correspond to PRVs, factor nodes (boxes) to parfactors. Edges occur between factor and variable nodes whenever the parfactor of the factor node contains the PRV behind the variable node. The PM depicted describes the dynamics of supply and demand in different zones Z on the globe using the PRVs $Supply(Z)$ and $Demand(Z)$. Vessels V move between zones, captured by $InArea(Z, V)$, which represent trade flows: Vessels can be found in zones with high supply (to load cargo), in zones with high demand (to discharge cargo), and in between while traveling. To infer in which zone new, and, more importantly, cost-covering cargo contracts can be closed, the level of freight rates $Rate(Z)$ are a key figure.

Need for transportation in the dry-bulk industry is mainly driven by supply and demand. Additionally, the availability of other vessels have an impact on finding cargo contracts. Owners, who have vessels in regions with supply, can charge lower freight rates for the transportation, as they do not need to be compensated for expenses for traveling to these regions. The potentials in the parfactors need to reflect this behavior. Parfactor g^1 encodes that supply of goods in a specific zone, given vessels being in the same zone, has a direct effect on the freight rate in this zone, e.g., having

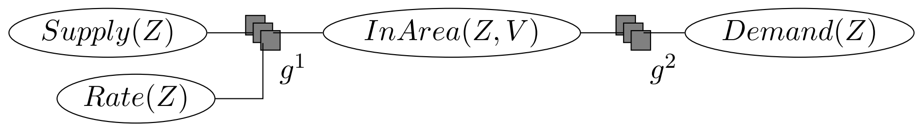

Figure 1: Parameterized probabilistic model G (and G_0)

high supply plus only a few vessels most probably results in high freight rates. Parfactor g^2 encodes that demand for goods in a zone additionally affects the number of vessels in the zone as a high demand usually yields many vessels on site for discharging.

Formally, we have the randvar names $\mathbf{R} = \{Supply, Rate, InArea, Demand\}$ and the logvar names $\mathbf{L} = \{Z, V\}$ with logvar Z denoting zones with domain $\mathcal{D}(Z) = \{z_1, \ldots, z_{n_z}\}$, and logvar V denoting vessels with domain $\mathcal{D}(V) = \{v_1, \ldots, v_{n_v}\}$. Using $\mathbf{R}$ and $\mathbf{L}$, we build the PRVs $Supply(Z), Rate(Z), InArea(Z, V), Demand(Z)$. The PRV $InArea(Z, V)$ has a boolean range. For sake of simplicity, the other PRVs have three range values: $\forall R \in \{Supply(Z), Rate(Z), Demand(Z)\} : \mathcal{R}(R) = \{high, medium, low\}$ The PM is given by $G = \{g^i\}_{i=1}^2$ with

$$g^1 = \phi^1(Supply(Z), Rate(Z), InArea(Z, V)), \quad g^2 = \phi^2(InArea(Z, V), Demand(Z)).$$

The semantics of a PM is given by grounding and building a full joint probability distribution. Querying the model corresponds to asking for probability distributions of a randvar using a model's full joint distribution and fixed events as evidence. Query answering algorithms seek to avoid grounding and building a full joint distribution when answering a query.

Supply and demand of goods change over time and, in consequence, vessel positions change accordingly. Therefore, we look at parameterized dynamic probabilistic models (PDMs) next.

3.2 Parameterized Probabilistic Dynamic Model

Based on static PMs, we define PDMs (Gehrke et al., 2018). A PDM is given by a pair of PMs, one representing an initial time step and the other how the model transitions from one time step to the next. PDMs are based on the first-order Markov assumption, i.e., randvars from time slice t depend only on randvars from the preceding time slice $t-1$. PDMs model a stationary process, i.e., changes from one time step to the next follow the same distribution. A formal definition of PDMs follows.

Definition 2 *A PDM is a pair of PMs $(G_0, G_\rightarrow)$ where G_0 is a PM representing the first time step with PRVs $\mathbf{A}_0$ and $G_\rightarrow$ is a two-slice PM using PRVs $\mathbf{A}_{t-1}$ and $\mathbf{A}_t$ where $\mathbf{A}_\pi$ is a set of PRVs from time slice π. $G_\rightarrow$ contains inter-slide parfactors $\phi(\mathcal{A})_{|C}$, which have arguments $\mathcal{A}$ under constraint C containing PRVs from both $\mathbf{A}_{t-1}$ and $\mathbf{A}_t$, encoding transitioning from time step $t-1$ to t.* Unrolling *$(G_0, G_\rightarrow)$ for T timesteps, i.e., instantiating $(G_0, G_\rightarrow)$ for a given number of time steps using G_0 for the initial time step and appending $G_\rightarrow$ to the corresponding PRVs for the other time steps, forms a PM, representing the full joint distribution $P_{(G_0, G_\rightarrow), T}$.*

Figure 2 shows the PM as described before with temporal information added. It forms $G_\rightarrow$ with the PM in Fig. 1 forming G_0. The parfactors g^S, g^V, and g^D are inter-slice parfactors. The submodel to the left and the submodel to right of these inter-slice parfactors are duplicates of each other, with the left referencing time step $t-1$ and the right referencing time step t.

The semantics of a PDM is given by unrolling the PDM (forming a PM), followed by grounding and building a full joint distribution (as with PMs).

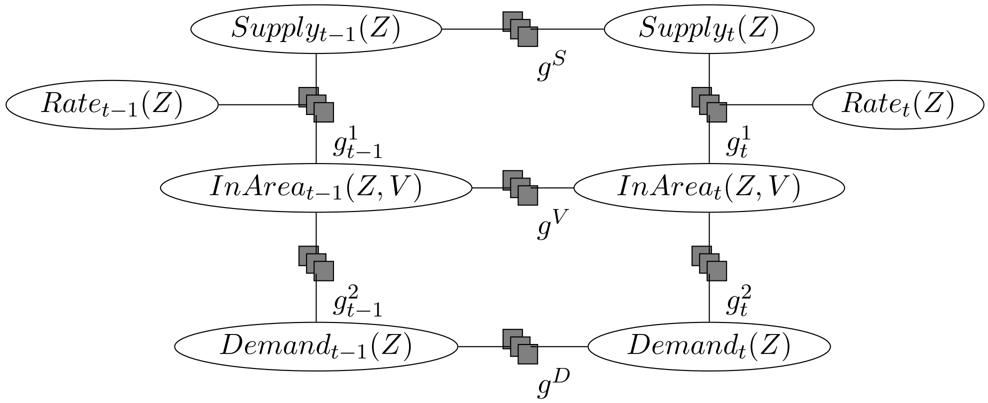

Figure 2: Two-slice parameterized probabilistic model $G_{\rightarrow}$

Definition 3 *Given a PDM G, a ground PRV Q_t, and evidence $\boldsymbol{E}_{0:t}$, the expression $P(Q_\pi \mid \boldsymbol{E}_{0:t})$, $\pi \in \{0, \dots, T\}$, $t \leq T$, denotes a* query *w.r.t. $P_{(G_0, G_{\rightarrow}),T}$. The problem of answering $P(A^i_\pi \mid \boldsymbol{E}_{0:t})$ is called* prediction *for $\pi > t$,* filtering *for $\pi = t$, and* smoothing *for $\pi < t$.*

PDMs allow for representing the relational and temporal aspects of the dry-bulk shipping scenario. The factorization in PDMs enables compactly encoding the full joint. The next step is to add actions and utilities to the PDM to cover the last open item, sequential decision making (Item d).

3.3 Parameterized Probabilistic Dynamic Decision Model

PDMs allow for (conditional) marginal queries, but their expressiveness is limited in terms of sequential online decision making. Besides asking questions on the overall setting of the current world, an agent's goal is to plan its fleet in a way that good freight rates can be expected such that the overall fleet performs the best. PDDecMs are PDMs with action and utility nodes added to efficiently solve an MEU problem (Gehrke et al., 2019). A rational agent should chose the action which maximizes the expected utility given evidence. More specifically w.r.t. our case, an operator of a fleet wants to benefit from good freight rates being paid for voyages performed between specific zones. The utility of having vessels in a certain zone depends on the freight rate in that zone.

A formal definition of PDDecMs follows in which actions and utilities are represented by PRVs as well. Additionally, for the temporal case, we define a utility transfer function which connects two utility PRVs to pass utility values of a time slice t to the next time slice $t + 1$.

Definition 4 *Let Φ_u be a set of utility factor names. A parfactor is a* utility parfactor *$\mu(\mathcal{A})_{|C}$ if its output is a utility U where C is a constraint on the logvars of $\mathcal{A}$ and μ is defined by $\mu : \times_{A \in \mathcal{A} \setminus \{U\}} \mathcal{R}(A) \mapsto \mathbb{R}$, with name $\mu \in \Phi_u$. The output of μ is the value of U. A* utility transfer function *λ has utility PRVs $\mathbf{U}$ and non-utility PRVs $\mathbf{A}$ as input and one utility PRV U_0 as output. The function λ specifies how the value of U_0 is additively changed, possibly depending on $\mathbf{A}$.*

A PDDecM *G is a PDM $(G_0, G_{\rightarrow})$ with an additional set G^u_0 of utility parfactors for G_0 and an additional set $G^u_{\rightarrow}$ of utility parfactors and utility transfer functions for $G_{\rightarrow}$. The term $rv(G^u)$ refers to all probability PRVs in $G^u = G^u_0 \cup G^u_{\rightarrow}$. G^u represents the combination of all utilities $U_{G,T} = \sum_{v \in \times_{r \in rv(gr(G^u,T))} \mathcal{R}(r)} f$ with $gr(G, T)$ denoting unrolling and grounding G for T timesteps.*

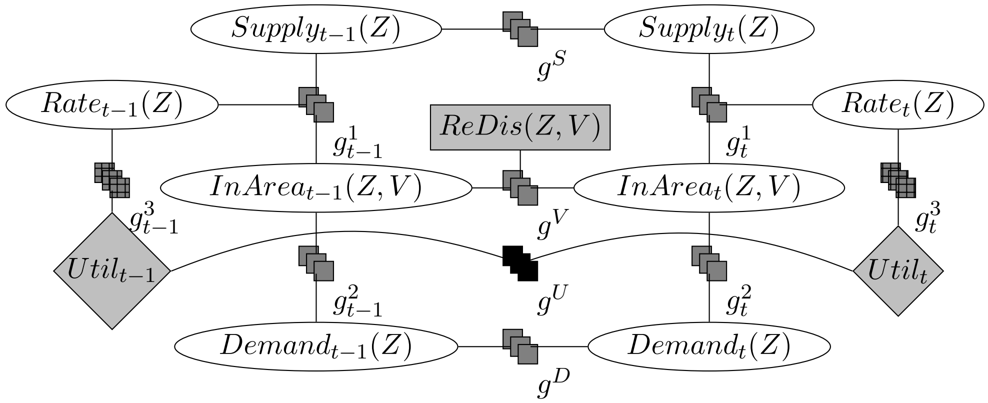

Figure 3: Parameterized probabilistic dynamic decision model (G_0: nodes with t, $t = 0$), $G_{\rightarrow}$: all)

With utility transfer functions, PDDecMs transfer utility values to the next time step and allow for discounting. Figure 3 shows the PDDecM resulting from the dry-bulk shipping PDM. We add to $G_{\rightarrow}$ one action PRV $ReDis(Z, V)$ (square), one utility PRV (diamond), utility parfactors (grid pattern), and a utility transfer parfactor g^U (black). G_0 would consist of the PRVs and parfactors in Fig. 3 indexed t with $t = 0$. Actions correspond to redistributing a vessel V to a zone Z in the next time step. The utility in a time step is calculated based on the level of freight rates and the vessels of an agent benefiting from them.

W.r.t. the requirements derived in Section 2, PDDecMs are able to describe our temporal, relational domain (Items a and b) in a compact matter utilizing the idea of lifting (Item e). PDDecMs also allow for efficient marginal query answering (Item c) and sequential decision support (Item d). Next, we examine more precisely how query answering is done in PDDecMs.

4. Query Answering

For decision support and marginal query answering, we use the inference algorithm meuLDJT (Gehrke et al., 2019). Next, we define the temporal MEU problem that meuLDJT solves and recap meuLDJT, and look at an empirical study.

4.1 The Temporal MEU Problem

Before looking at meuLDJT, we define the temporal MEU problem, for which we need utility queries in addition to the (probability) queries, defined in 3.

Definition 5 *Given a PDDecM G, a ground PRV Q, and evidence E, the expression $P(Q|E)$ denotes a probability query w.r.t P_G and the expression $U(Q, E)$ refers to a utility w.r.t. U_G.*

The expected utility of a PDDecM G, given an assignment to action PRVs $\mathbf{a}$ and a maximum number of timesteps T, is calculated as:

$$eu(\mathbf{E}, \mathbf{a}, T) = \sum_{v \in \times_{r \in rv(G,T)} \mathcal{R}(r)} P(v|\mathbf{E}, \mathbf{a}) \cdot U(v, \mathbf{E}, \mathbf{a}) \tag{1}$$

where $rv(G, T)$ refers to all PRVs in G if G were unrolled for T timesteps.

The expected utility eu as per Eq. (1) is basically determined by summing out all probability PRVs (iterate over range values of PRVs involved) given evidence and actions $\mathbf{a}$. Transferred to our model, as depicted in Fig. 3, calculating, e.g., the expected utility of $Util_0$ without any evidence given corresponds to summing out all PRVs in G_0. The MEU problem is defined as follows:

$$meu[G|\mathbf{E}, T] = (\mathbf{a}^* = \arg\max_{\mathbf{a}} eu(\mathbf{E}, \mathbf{a}, T), eu(\mathbf{E}, \mathbf{a}^*, T)). \tag{2}$$

Using the utility transfer function, the expected utility is calculated for one time step and then its value is passed to the next time step. The utility value of the last time step corresponds to the overall utility value. Equation (2) refers to selecting the sequence of actions with maximum expected utility.

4.2 meuLDJT

The algorithm, meuLDJT, is an algorithm that forms a helper structure for handling temporal aspects and multiple queries efficiently, see the work by Gehrke et al. (2019) for details. meuLDJT answers queries for a finite horizon T. For each time step, it calculates a utility, which is multiplied by the number of groundings, and then passed to the next time step. When answering an MEU query, meuLDJT constructs all action sequences by setting them as evidence to answer the expected utility query. In the end, the action sequence with the maximum expected utility value is returned.

The important part for this paper is that meuLDJT uses lifted variable elimination (LVE) (Poole, 2003; Taghipour et al., 2013a) as a subroutine during its calculations. LVE exploits symmetries that lead to duplicate calculations: query answering is done by eliminating PRVs, which are not part of the query, by so called *lifted summing out*. Basically, VE is computed for one instance and exponentiated to the number of isomorphic instances represented. A lifted solution to a query on a model means that meuLDJT computes an answer without grounding a part of the model. LVE as the subroutine carrying out the computations is guaranteed to not ground a logvar during its calculations if individual parfactors do not contain more than two logvars or if individual PRVs do not contain more than one logvar (Taghipour et al., 2013b). The dry-bulk shipping model in its current form contains only two logvars per parfactor and as such has a lifted solution regarding LVE.

Next, we look at an empirical evaluation, considering available implementations for actually answering queries on our model.

4.3 Empirical Evaluation

For the empirical evaluation, we have considered the following implementations: (i) Alchemy[1] for lifted, static models, (ii) Forclift[2] for lifted, static models, (iii) UUMLN[3] for lifted, temporal models but its calculations are all performed on a ground level, (iv) the implementation of the static version of LDJT[4] for PMs (lifted, static), and (v) the implementation of LDJT[5] for PDMs, which also includes a propositional version of LDJT. Since the first three *implementations* can only handle boolean ranges and do not support decision-theoretic constructs, we use the PDM $(G_0, G_\rightarrow)$ as depicted in Figs. 1 and 2 with boolean ranges as inputs. For comparison, we also use $(G_0, G_\rightarrow)$ with the original ranges as input for LDJT. Given $(G_0, G_\rightarrow)$, we ask a filtering query $P(Rate_t(z_1))$

1. `http://alchemy.cs.washington.edu`
2. `https://github.com/UCLA-StarAI/Forclift`
3. `https://www.uni-ulm.de/en/in/ki/inst/alumni/thomas-geier/`
4. `https://www.ifis.uni-luebeck.de/index.php?id=518`
5. `https://www.ifis.uni-luebeck.de/index.php?id=483`

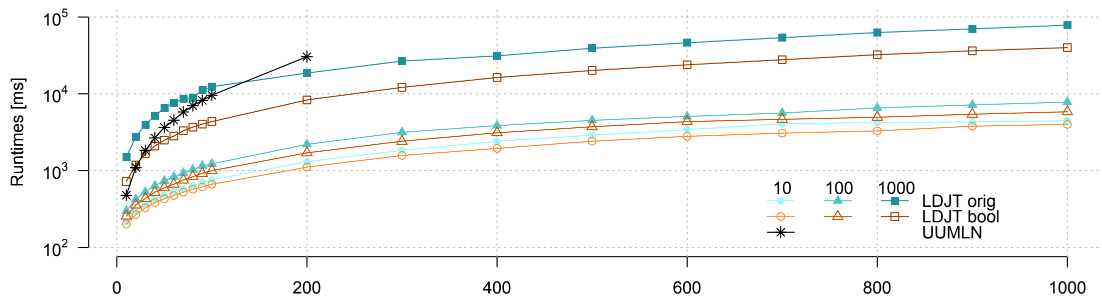

Figure 4: Runtimes in milliseconds [ms] with varying T on the x-axis; $|\mathcal{D}(V)| \in \{10, 100, 1000\}$, ranges as given in paper ("orig") or all boolean ("bool")

for each $t \in \{1, \ldots, T\}$ with $T \in \{10, 20, \ldots, 100, 200, \ldots, 1000\}$ without any evidence (Forclift does not support lifted evidence), which leads to T queries per run. For $\mathcal{D}(V)$, we use $|\mathcal{D}(V)| \in \{10, 100, 1000\}$. We keep $\mathcal{D}(Z)$ constant at $|\mathcal{D}(Z)| = 10$. We unroll the PDM for T timesteps to get static models. As the first three implementations use (dynamic) MLNs as inputs, we transform the parfactor-based model into an equivalent MLN as described in Van den Broeck (2013).

We report total runtimes averaged over five runs, with a cutoff time of 3 hours and 16GB of working memory. Since UUMLN always answers queries for each possible grounding, we interpolate from the total number of queries UUMLN answers, to the number of queries UUMLN would only need to answer. We expect the lifted, static models to perform worse than the lifted, temporal models since they are not able to handle the temporal aspect properly. Additionally, we expect LDJT to perform better than its propositional counterpart as well as UUMLN, since they do their calculations on a propositional level. For LDJT, we expect runtimes to depend at maximum linearly on T with only filtering queries and polynomially on domain sizes.

Figure 4 shows runtimes in milliseconds for the LDJT implementation with original (filled symbols) and boolean ranges (hollow) for all three domain sizes (10: circles, 100: triangles, 1000: squares) over all T as well as runtimes for UUMLN for a domain size of 10 and T until 200 (star). For larger T, UUMLN has triggered the cutoff time, which shows that lifted calculations have an advantage over ground calculations. All other implementations run into memory errors, except with the smallest model ($|\mathcal{D}(V)| = 10$, $T = 10$), triggering the cutoff time, which fulfills our expectation that algorithms for static models cannot handle temporal models very well. The numbers show that the LDJT runtimes exhibit a linear dependence over time. The runtimes of LDJT for different domain sizes exhibit a polynomial increase since the difference between the lines indicate a close to linear factor. The runtimes of LDJT with the original ranges are higher than the runtimes with boolean ranges as LVE has to sum out only two instead of three values for each elimination.

The empirical evaluation supports that lifted, temporal formalisms work best for temporal relational probabilistic problems.

5. Open Challenges

Given our modeling of the dry-bulk shipping market, we have made some assumptions that may not hold and have identified three main challenges in terms of query answering.

Non-stationary Process The assumption of stationarity is often not compatible with real-world applications such as the one described here. Non-stationarity can manifest itself in something as simple as domain sizes changing over time since ships are constantly built or dumped depending on commercial and environmental factors. Non-stationary processes are barely covered in propositional or lifted inference as they further complicate an already hard problem.

Liftability The model here addresses the core fundamentals of the dry-bulk shipping market. More aspects might be worth incorporating in the model, e.g., different commodities C or different operating agents A from Section 1. Adding more logvars may lead to losing liftability. In terms of scalability by liftability, we need new lifting operators that can handle three-and-more-logvar parfactors or ways to transform an input model into an equivalent liftable model. Friedman and Van den Broeck (2020) work on solving a similar problem, reformulating database queries with three variables into equivalent queries with two variables under certain conditions.

Action Modelling For decision support in dry-bulk shipping, a more sophisticated action model is necessary, where actions depend on each other or do not take effect directly in the next timestep, which would require leaving behind the Markov-1 assumption of one timeslice only depending on the previous timeslice. One example are vessels being sent to new regions ($ReDis(Z, V)$). Travel times between the different zones may simply vary due to their geographical position. Thus, redistributing vessels does not always take effect in the next timeslice.

6. Conclusion and Outlook

This paper examines existing formalisms and inference algorithms for the dry-bulk shipping context with the following requirements and characteristics considered: (a) probabilistic, relational nature, (b) temporal behavior, (c) efficient marginal query answering, (d) sequential decision support, and (e) compact representation (large domain). PDDecMs together with meuLJDT fulfill the requirements of compactly representing relations and entities over time and answering (conditional) marginal queries while also enabling sequential decision support on a lifted instead of a propositional level. Additionally, the evaluation has shown that using an approach explicitly handling time outperforms static approaches if time moves on.

Nonetheless, the model described in this paper addresses only the core fundamentals of the dry-bulk shipping market, abstracting away certain entities and relations as well as specific behavior regarding changing dynamics over time. We have identified three main challenges about the stationarity of the underlying process, the liftability of the calculations, and the effect of actions, which are interesting avenues for future work.

References

T. Braun and R. Möller. Lifted Junction Tree Algorithm. In *Proceedings of KI 2016: Advances in Artificial Intelligence*, pages 30–42. Springer, 2016.

T. Friedman and G. Van den Broeck. Symbolic Querying of Vector Spaces: Probabilistic Databases Meets Relational Embeddings. In *9th International Workshop on Statistical Relational AI at the 34th AAAI Conference on Artificial Intelligence*, 2020.

M. Gehrke, T. Braun, and R. Möller. Lifted Dynamic Junction Tree Algorithm. In *Proceedings of the International Conference on Conceptual Structures*, pages 55–69. Springer, 2018.

M. Gehrke, T. Braun, and R. Möller. Lifted Temporal Maximum Expected Utility. In *Proc. of the 32nd Canadian Conference on Artificial Intelligence*, pages 380–386. Springer, 2019.

K. Kersting, B. Ahmadi, and S. Natarajan. Counting Belief Propagation. In *Proc. of the 25th Conference on Uncertainty in Artificial Intelligence*, pages 277–284. AUAI Press, 2009.

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

A. Nath and P. Domingos. A Language for Relational Decision Theory. In *Proc. of the 6th International Workshop on Statistical Relational Learning*, 2009.

M. Niepert and G. Van den Broeck. Tractability through Exchangeability: A New Perspective on Efficient Probabilistic Inference. In *Proc. of the 28th AAAI Conference on Artificial Intelligence*, pages 2467–2475. AAAI Press, 2014.

D. Poole. First-order Probabilistic Inference. In *Proc. of the 18th International Joint Conference on Artificial Intelligence*, pages 985–991. IJCAI Organization, 2003.

M. Richardson and P. Domingos. Markov Logic Networks. *Machine Learning*, 62(1–2):107–136, 2006.

S. Ross, J. Pineau, S. Paquet, and B. Chaib-draa. Online Planning Algorithms for POMDPs. *Journal of Artificial Intelligence Research*, 32:663–704, 2008.

S. Sanghai, P. Domingos, and D. Weld. Relational Dynamic Bayesian Networks. *Journal of Artificial Intelligence Research*, 24:759–797, 2005.

S. Sanner and K. Kersting. Symbolic Dynamic Programming for First-order POMDPs. In *Proc. of the 24th AAAI Conference on Artificial Intelligence*, pages 1140–1146. AAAI Press, 2010.

N. Taghipour, D. Fierens, J. Davis, and H. Blockeel. Lifted Variable Elimination: Decoupling the Operators from the Constraint Language. *Journal of Artificial Intelligence Research*, 47(1): 393–439, 2013a.

N. Taghipour, D. Fierens, G. Van den Broeck, J. Davis, and H. Blockeel. Completeness Results for Lifted Variable Elimination. In *Proceedings of the 16th International Conference on Artificial Intelligence and Statistics*, pages 572–580. AAAI Press, 2013b.

UNCTAD. *Review of Maritime Transport 2019*. UNCTAD, 2019.

G. Van den Broeck. *Lifted Inference and Learning in Statistical Relational Models*. PhD thesis, KU Leuven, 2013.

C. Wang and J. Schmolze. Planning with POMDPs using a Compact, Logic-based Representation. In *Proc. of the 17th IEEE International Conference on Tools with Artificial Intelligence*, pages 8–530, 2005.

N. L. Zhang and D. Poole. A Simple Approach to Bayesian Network Computations. In *Proc. of the 10th Canadian Conference on Artificial Intelligence*, pages 171–178. Springer, 1994.