UNIVERSIDAD
DE MÁLAGA

Universidad de Málaga
Departamento de Arquitectura de Computadores

TESIS DOCTORAL
Ph.D. Thesis

Computational methods and parallel strategies in dynamic decision making

Autor: Alejandro Gutiérrez Alcoba
Directores: Dr. Eligius M.T. Hendrix y Dra. Gloria Ortega López
Programa de Doctorado: Tecnologías Informáticas
Centro: E.T.S. de Ingeniería Informática
Málaga, Octubre 2017

AUTOR: Alejandro Gutiérrez Alcoba

http://orcid.org/0000-0002-5996-1928

EDITA: Publicaciones y Divulgación Científica. Universidad de Málaga

Ph.D. Thesis

Computational methods and parallel strategies in dynamic decision making

Universidad de Málaga
Departamento de Arquitectura de Computadores

Author: Alejandro Gutiérrez Alcoba
Supervisors: Dr. Eligius M.T. Hendrix
Dra. Gloria Ortega López

Málaga, Octubre 2017

Carta de Aval

El Dr. Eligius M.T. Hendrix, Profesor Titular del Departamento de Arquitectura de Computadores de la Universidad de Málaga, la Dra. Gloria Ortega López, PDI del Departamento de Arquitectura de Computadores de la Universidad de Málaga, ambos como directores y, la Dra. Inmaculada García Fernández, Catedrática del Departamento de Arquitectura de Computadores de la Universidad de Málaga, como tutora,

CERTIFICAN:

Que D. Alejandro Gutiérrez Alcoba ha realizado la memoria titulada "Computational Methods and Parallel Strageties in Dynamic Decision Making" en el Departamento de Arquitectura de Computadores de la Universidad de Málaga y que ésta constituye la Tesis para optar al grado de Doctor en Tecnologías Informáticas. Estimamos que puede ser presentada y, para que conste a efectos de lo establecido en la legislación vigente, autorizamos la presentación de la la Tesis Doctoral en la Universidad de Málaga.

Málaga, Enero de 2018.

Firmado: Dr. Eligius M.T. Hendrix (Director)

Firmado: Dra. Gloria Ortega López (Codirectora)

Firmado: Dra. Inmaculada Garía Fernández (Tutora)

Agradecimientos

La realización de esta tesis ha sido posible gracias a la financiación recibida mediante el contrato predoctoral (FPI) con referencia BES-2013-064235, del Ministerio de Economía y Competitividad, a través de los proyectos de investigación TIN2012-37483 y TIN2015-66680, a las ayudas a la movilidad predoctoral para la realización de Estancias Breves en centros de I+D asociadas a dicho contrato predoctoral y a la Consejería de Economía, Innovación, Ciencia y Empleo de la Junta de Andalucía a través del proyecto P11-TIC-7176, cofinanciados por el Fondo Europeo de Desarrollo Regional (FEDER).

Me gustaría comenzar agradeciendo a mis supervisores de tesis, el Dr. Eligius M.T. Hendrix y la Dra. Gloria Ortega por su esfuerzo, tiempo y paciencia trabajando conmigo. En ambos veo un ejemplo a seguir por su admirable dedicación. También quiero agradecer a mi tutora, la Dra. Inmaculada García por su consejo, experiencia y por la confianza depositada en mí.

Un aspecto clave de esta tesis doctoral ha sido la experiencia ganada y el trabajo realizado durante estancias en la Universidad de Edimburgo, en Reino Unido y la Universidad de Bergen, en Noruega. En Edimburgo tuve la suerte de contar con la clara visión del Dr. Roberto Rossi y la Dra. Belén Martín y estoy muy agradecido de poder haber contado con su ayuda. También quiero mostrar mi agradecimiento al Dr. Dag Haugland, por su ayuda durante mi estancia en Bergen y por su ojo crítico, el cual valoro profundamente: Tusen takk!

Además, me gustaría agradecer, por su colaboración, al resto de coautores de los artículos que forman esta tesis: el Dr. René Haijema y las Dras. Karin G.J. Pauls-Worm y Elin E. Halvorsen-Weare.

Pero no solo a base de esfuerzos académicos se escribe una tesis. Soy consciente de que el crecer en un ambiente de amor y soporte sin duda facilitan el poder aventurarse en un reto como este, y por ello me siento afortunado. En particular, quiero dedicar esta tesis a mis padres, Francis y Gracia, a quienes les debo todo, y a mi hermana Irene, por su apoyo. A mis familiares, amigos y otros compañeros que he conocido a lo largo de estos años en Málaga, Almería, Edimburgo y Bergen: gracias por estar ahí y por vuestras palabras de ánimo y optimismo, que me han ayudado a continuar con más fuerzas.

Por último, el lugar de honor es sin duda para Mairi, pilar fundamental en mi vida. Gracias por tu paciencia, tu sacrificio y tu comprensión. Esta tesis no habría sido la misma sin ti, literalmente.

Preface

This thesis discusses dynamic decision making applications for a set of problems. Two main lines can be distinguished. The first deals with supply chain management problems for perishable products while the second studies the design of vessel fleets upon performing maintenance operations at offshore wind farms. The inventory models for perishable products studied in this thesis consider a single-item, single-stock location and production planning over a finite time horizon. The decision making problem of scheduling the maintenance operations at offshore wind farms is treated as a supply chain problem type: the installation requires to schedule maintenance operations and attend failures in turbines during the planning horizon. A fleet of vessels needs to be selected to support these operations. For this set of problems, decisions are not only dynamic, but are also made under uncertainty.

The main objectives of this thesis are the following: (1) to study which order policies are the most appropriate for the designed perishable lot sizing problems. In which cases an order policy gives an optimal solution?; (2) to analise the effect of using parallel computing to improve the performance of the algorithms derived designing policies for perishable lot sizing problems; (3) to explore how effective heuristics can be for dynamic decision making in lot sizing problems for perishables; (4) to elaborate an MILP model for selecting a fleet of vessels to support the maintenance operations at offshore wind farms; and (5), to design a heuristic for scheduling maintenance operations at offshore wind farms considering failures in turbines and weather uncertainty.

Each one of these objectives have been discussed in a separate chapter of this thesis. In the second Chapter, a stochastic programming model is presented for a practical production planning problem of a perishable product over a finite time horizon. A static policy is studied for that model. Such policy proved to be optimal assuming a static uncertainty strategy, which is considered for instances with a long lead time. The third Chapter addresses the use of parallel computing for the algorithms developed in the previous Chapter. Two implementations were developed for heterogeneous platforms: a multi-GPU version using CUDA and a multicore version using Pthreads and MPI. For the first implementation the Monte Carlo simulation (the most demanding task) is parallelised. The multicore version showed a good speedup, after dealing with an initially unbalanced workload among processors. The fourth Chapter discusses the effectiveness of heuristics for a similar lot sizing problem for perishables. The classical Silver heuristic is extended for perishable products and an analytical and a simulation-based variant of the approach are introduced. The results of the heuristics are compared with the optimal solutions given by a derived SDP model for the problem, showing that the heuristics feature costs that are, on average, 5% above the optimal cost for the simulation approach and 6% for the analytical approximation. In the fifth Chapter, a MILP model to select the optimal fleet of vessels to operate the maintenance of an offshore wind farm is derived. The model is presented as a bi-level problem, selecting the optimal fleet on the first level and optimising the schedule of operations, using the fleet, on the second. Since this model is deterministic, as others in literature aiming to solve long time horizon problems using small time periods, the sixth Chapter address the question of how the anticipation of stochastic events such as the failures in turbines or the weather conditions affect the decision of the optimal vessel fleet. This Chapter presents a heuristic decision rule that illustrates this effect.

Contents

1 Introduction

1.1 Background on Dynamic Decision Making

Decision making is present in all sort of forms and scopes in our daily live. From the common decisions individuals have to face as part of everyday life, to the ones made by governments and large corporations. From the irrelevant, to the ones that make an impact in our society or economy. Modelling a situation to take better decisions is often a complex task, due to the level of detail that real-world problems require with different interconnected and interdependent layers. Uncertainty, sometimes intrinsic, otherwise considered because of the unknown interdependencies of the environment, adds complexity to tackle the models and to find solutions. Finally, in situations in which the decisions are taken over time, the dynamic character arises when the interactions of actions affect forehead decisions.

A classical definition of dynamic decision making (DDM) can be found in [17]. Dynamic decision making problems can be defined as those that encompass a series of decisions over a set of possible actions that are taken in real time to achieve a single or several objectives in an environment that changes over time, both as a consequence of previous decisions or autonomously due to external factors. One of the complications of DDM problems comes from the fact that decisions are not independent; previous decisions affect and constrain later decisions. Also, previous decisions have an effect over the state of the environment. In this context, dynamic decisions cannot be taken, in general, without considering the long-term effect they may have over the state of the system.

Bellman's principle of optimality [7] lies at the core of dynamic decision theory: *An optimal policy has the property that whatever the initial state and decisions are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.* The mathematical optimization method known as dynamic programming follows the principle of optimality by breaking down a dynamic optimization problem into simpler subproblems. This method can be extended to account for uncertainty. Other combinatorial optimization methods such as branch and bound, artificial intelligence techniques like genetic algorithms and other metaheuristics have been used extensively in literature.

This thesis discusses DDM for several applications of real-world problems. Two main different threads have been studied relating these DDM problems: supply chain management for perishable products and maintenance of offshore wind farms (OWF). In these applications, decisions are made under uncertainty. In the first case, demand for products is stochastic. The shelf-life of perishable products can be subject to uncertainty as well, although this issue has not been contemplated in this thesis. For the OWF maintenance case, weather circumstances may be considered as they affect the energy outcome of the turbines and harsh conditions prevent maintenance vehicles to leave their bases. Also, the events of failures in the turbines occur randomly.

1.2 Inventory control

Inventory control is a typical example of making decisions in time, where mostly uncertainty plays a role. This thesis includes studies on lot sizing problems for perishable items. There are some general aspects of inventory control, regardless of perishability, that are defined for any model. The most important considerations are: number of items, inventory locations, holding capacity, review frequency, planning horizon, lead time (between placement and receipt orders), demand and backlogging [77].

A lot sizing problem may deal with single or multiple items, and single or multiple inventory locations. A maximum holding capacity may be considered at the inventory locations, or assumed to be infinite when space is not a concern. Reviewing the inventory status may be continuous or periodic. In a periodic system, the inventory levels are checked at time intervals, while in a continuous system they are checked for every period. The planning horizon is a time series that may have a finite number of periods or be infinite. The lead time refers to the time in periods, between the placement and the receipt of orders. Demand can be deterministic or stochastic. For the deterministic case, we can differentiate between static, stationary or non-stationary demand. Related to the demand, the system may allow backlogging: if demand exceeds the inventory on hand, the excess can be hold for the next replenishment of items.

We understand *replenishment cycle* as any set of periods between two consecutive replenishment periods. To measure the performance of inventory replenishment cycles, service level metrics are commonly used in the supply chain. The most widely used in industrial practice are the so called α-service and β-service level. The first measures the probability of not having a stock-out during a replenishment cycle. The second, also known as fill rate, denotes the expected percentage of the demand that can be fulfilled during a cycle.

From a modelling point of view, stochastic lot sizing problems can be classified according to the timing of the orders and their quantity. Different strategies can be considered to determine order quantities for the periods. Bookbinder and Tan defined strategies for the lot sizing problem with stochastic demand [10]:

- Static uncertainty model
- Static-dynamic uncertainty model
- Dynamic uncertainty model

For the *static uncertainty* model, the order quantities are defined at the beginning of the planning horizon, before demand is observed, and cannot be changed through the development of the periods. This strategy is appropriate for models in which the order periods and quantities must be known in advance, for instance due to a very long lead time.

For the *static-dynamic* model, order timing is set in advance, at the beginning of the time horizon, but the order quantities rely on the observation of the inventory for each cycle.

Finally, for the *dynamic* model, the most studied model in the literature, both the ordering periods and their quantities are flexible. In practice, the order quantities are decided at the beginning of each period, before demand is observed, knowing the inventory on hand from the previous period.

These models can be adapted for the case of perishable items. In that case, the model may take, or not, the stock age distribution into account, differentiating between a *stock age dependent* or *stock age independent* strategy, respectively. In the models studied in this thesis, both *stock age dependent* and *stock age independent* strategies are applied.

1.2.1 Perishable inventory control

According to the Food and Agriculture Organisation of the United Nations, around one-third of the food produced worldwide for human consumption is lost or wasted, amounting to about 1.3 billion tons per year. This loss can be translated into a waste of different valuable resources such as land, water or energy. For this reason, research on perishable item inventory control represents an area of increasing interest.

Perishable products are those whose quality or utility decays over time. In [88], perishability is defined as the decay, damage, spoilage, evaporation, obsolescence, pilferage, loss of utility or loss of marginal value of a commodity that results in decreasing usefulness from the original one. The shelf-life of a perishable product is the time it can be used or consumed, usually since they have been produced or acquired. From a modelling perspective, [59] describes two categories for perishable products: (1) products with a fixed shelf-life, in which the shelf-life is known beforehand and remains constant and (2) products with random shelf-life, in which the shelf-life is given by a stochastic variable. This second category can also be subdivided (see [5]) into two different types: (a) shelf-life deterioration rate depending on age and (b) deterioration rate depending on time or inventory level. More recently, [3] proposed a framework for classifying perishability distinguishing three dimensions: (a) physical product deterioration, (b) authority limits, and (c) customer value. Authority limits refer to external regulations that artificially affect the shelf-life of perishables. An example of that is human blood, for which tight shelf-lives are usually considered for prevention. Customer value refers to the perceived value of a product, which may decrease when the product is physically unaltered (newspapers, fashion, consumer electronics).

When it comes to mathematical modelling, in lot sizing models for perishable items, the structure of the optimal replenishment policy is typically complex: the replenishment quantity depends on the individual age categories of current inventory, as well as on all outstanding orders.

1.2.2 Problem statement

The inventory models for perishable products studied in this thesis consider a single-item, single-stock location and production planning problem over a finite time horizon. The considered perishable items have a fixed shelf-life, being scrapped after they reach that limit. The models operate under a FIFO (first in, first out) issuing policy for products. We suppose that items are delivered or produced instantaneously at the beginning of the period that they are ordered. The demand is stochastic and non-stationary. If demand exceeds the inventory volume, it is backlogged or lost. In the second case, the unmet demand is controlled by a service level constraint. Specifically, a β-service level has been used. Regarding cost parameters, the models consider a fixed setup or ordering cost, procurement and holding costs per unit, disposal cost for items that reach the end of their shelf-life and penalty costs for the unmet demand.

The aim of studying this setting is to gain insight for the following research questions:

1. Which order policies are the most appropriate for this problem setting?
2. In which cases an order policy gives an optimal solution?
3. How can the use of parallel computing improve the performance of the algorithms to find solutions?
4. To which extend the use of heuristics give good results in a DDM lot sizing problem for perishable products?

1.2.3 Related works

From the original Economic Order Quantity (EOQ) model, first described in [38], lot sizing problems have been studied profusely due to the important role of supply chain management in the economy. Dynamic lot sizing was first introduced by [86], who discuss a polynomial time exact solution method. One of the precursors of the modern lot sizing theory comes from the definition of the so-called (s, S) or order up to level S policy: if the size of the inventory falls below a level s, an order to reach level S is placed. For a general setting in which stochastic demand is stationary and ordering, unit, holding and shortage costs are considered, Scarf proofs that the (s, S) policy is optimal [72]. Later in [47], the (s, S) policy was proven to be optimal for an infinite time horizon as well. Many efficient heuristics appeared over the years in the literature, see e.g. the linear time heuristic introduced in [76].

Other policies, heuristics and models in general considering uncertainty for the demand appeared later in the literature. Silver presented a heuristic that looks only one cycle ahead and that is based on three different stages: deciding when to order, the cycle length and the order quantity [75]. The heuristic introduced in [4] determines the replenishment levels minimising the incurred cost per period. Authors in [9] improve the latter both in cost and computational time. In [10] a heuristic in which first the replenishment periods are decided and after that the quantities are fixed is proposed . In [84] Tarim and Kingsman present an MILP (Mixed-Integer Linear Programming) model used to decide replenishment moments and quantities simultaneously; Authors in [69] generalise Tarim and Kingsman's model to handle a range of service level measures as well as lost sales. In [13] ordering policies are considered for systems in which the fixed cost is dependent on the order size, in a step function for two or multiple values, deriving policies for these cases. Authors in [89] determine a joint ordering and dynamic pricing strategy for three different models and characterize the optimal policy when inventory cost-rate functions are convex or quasi convex. All aforementioned works operate under a non-stationary demand assumption; the importance of developing models that are able to compute optimal or near-optimal non-stationary policies has been discussed by [85].

The earliest works in which perishability is considered as an aspect of lot sizing problems appear last century around the sixties. A review of the early literature on lot sizing for perishables is provided by [59]; it surveys inventory models for perishable products with a fixed lifetime from 1960 to 1982. Karaesmen et al. [49] make an extensive review of more recent literature for perishables with fixed or random lifetime and considering both discrete and continuous models. In [5], inventory models for perishables since 2011 are reviewed.

According to the above reviews, over the last ten years several inventory models have been derived for controlling perishable item inventory systems. For instance, [57] presents a model similar to the one discussed in this paper dealing with a periodic review with service-level constraints; however the role of the fixed ordering cost is not considered in their work. More recently, [41] introduces an SDP approach for a single perishable item subject to non-stationary demand and an α service level constraint. In [55] multi-modularity to three dynamic inventory problems is applied; they consider perishability in one of them, for clearance sales, following FIFO issuance. Authors in [14] analyse a joint pricing and inventory control problem for perishables, considering both a backlogging case and a lost-sales case; they allow that inventory can be discarded before perishing. In [65] an MILP approximation model for a YS policy is presented for an inventory control problem under α service level constraints, non-stationary demand and a single item with a fixed shelf life. In this policy, Y_t provides the order timing, i.e., it is an indicator variable that is set to one if there is a replenishment up to inventory level S_t in period t. In [62], an MILP approximation for a YQ policy obtaining costs that are less than 5% more than those of the optimal policy is presented.

In this thesis, an stochastic programming (SP) model is presented for a practical production planning problem of a perishable product over a finite time horizon. An YQ policy is studied assuming a static uncertainty strategy. In a different study, a similar model considering a dynamic strategy explores and discusses the effectiveness of two new heuristics.

1.3 Offshore wind farm maintenance

We can distinguish seven different renewable energy sources that are known and used. These are hydro power, wind, solar, tidal, wave, geothermal and biomass (including biofuels) [18]. The technology for generating energy from wind has experienced a rapid development during the last decades, whereas the offshore generation has been last exploited. The offshore wind energy industry is expected to continue its growth tendency in the near future. For instance, the European Wind Energy Association expects in its Central Scenario by 2030 a total installed capacity of 66 GW of offshore wind in the UE [20].

The increasing interest in investing, optimising and improving the technology of renewable energy sources such as wind and solar power, responds not only to new political policies, but also to a real concern for the environment and the limits of production of fossil fuels (oil, coal and natural gas), in a global economy paradigm of limitless growth.

While other technologies developed for renewable energy sources like hydroelectric have long been studied and implemented, proving to be very profitable, others like wind and sun power face more challenges to achieve profitable levels. In the case of wind power, wind farms are typically large infrastructures that rely on the use of heavy machinery powered with fossil fuels for their installation and maintenance. Therefore, optimising installation and maintenance processes for (offshore) wind farm constitutes an interesting field of research. From the Operations Research perspective, this includes several opportunities, such as determining optimal array cable layouts, minimising the cost of the installed cable [6]; determining the optimal turbine layout considering the wind wake effect in order to maximise electricity production on the farm [15]; and optimising the installation planning itself for a wind farm [73]. While the installation of an OWF constitutes its major cost, operations and maintenance (O&M) activity still accounts for about a 25% of the life-time cost of an OWF [1]. Optimising the resources used for O&M activity is an interesting and challenging problem, in which only a few approaches have been analysed and studied so far.

1.3.1 Problem statement

This thesis focuses on the decision making problem of scheduling the O&M at OWF's and the selection of an appropriate fleet of vessels to support these operations. A more detailed description of the problem at hand is presented in Chapter 5. The model is a supply chain problem type; there is an OWF which turbines require maintenance during a time horizon. The aim is to determine an optimal fleet of vessels to support all the O&M activities needed. The decision maker may choose from a variety of vessel types to charter during the time horizon. The vessels operate at the OWF from their bases, that are at a certain distance to the OWF and have a certain capacity for holding vessels at a cost. Each vessel may perform activities during each shift, going from their base and returning to it by the end of each shift. Weather conditions apply preventing vessels to sail when the conditions are not adequate.

The type of maintenance activities that are performed at the OWF can be classified into two groups: preventive and corrective. Preventive activity types correspond to those that have the aim of prolonging the shelf-life of the turbines and prevent malfunctioning. A number of each preventive type is supposed to be performed during the time horizon. The corrective types aim to fix their corresponding failure types in the turbines. A corrective activity can be performed since the moment a failure is diagnosed, updated at the beginning of each period.

At the end of the time horizon, the activity types (preventive or corrective) that have not been performed incur a penalty cost. Downtime costs apply since the moment a turbine presents a failure until it is fixed. While performing a preventive activity, the turbine must be shut down and downtime costs apply as well, due to loss of energy generation. Other costs are associated to the missions the vessels perform, the chartering costs and the use of the selected bases.

This problem setting is the base for formulating the following research questions for this thesis:

1. Is an MILP model suitable for an application for selecting a fleet of vessels to support the maintenance at OWF's?

2. Is it possible to find an efficient and realistic heuristic for scheduling O&M activities at OWF's with failures and weather uncertainty? What are the differences with a perfect information MILP model?

1.3.2 Previous works

Optimization for maintenance operations at OWF is a novel area with few research papers that nonetheless is rapidly gaining interest. A literature review on DSS for OWF's is given by [45]. Recently, a mathematical model for maintenance operations at OWFs using a fleet of vessels has been presented in [66]. The authors also propose to solve it using a rolling horizon heuristic. Other recent deterministic and stochastic model formulations for vessel composition and maintenance optimization can be found in [26] and [37]. In [48], a model for maintenance routing and scheduling at offshore wind farms based on the Dantzig-Wolfe decomposition method has been implemented. In that work, a mixed integer linear program is solved for each subset of turbines to generate all feasible routes and maintenance schedules for the vessels for each period. The routes take several constraints into account, such as weather conditions, the availability of vessels, and the number of technicians available at the operation and maintenance base. In [82], a two-stage stochastic programming model is presented to determine a cost-optimal fleet size and mix for O&M activities at offshore wind farms for the total expected lifetime of the OWF. For that, the study considers time periods fixed to three months. The uncertainty about the failures in turbines constitutes a high cost for the maintenance of OWF's. In [40], a big data approach is used to gain insights to predict failures in turbines.

In this thesis, a DDS model for selecting a fleet of vessels to perform O&M at OWF's is presented. The model considers scenarios of deterministic weather circumstances and failures in turbines that are fixed by technicians attending the turbines using the fleet of vessels available. A heuristic is proposed and confronted with the solutions given by the MILP model based on perfect information.

1.4 Thesis outline

This thesis is organised as follows. The first part of this thesis focuses on perishable lot sizing problems. In Chapter 2, a YQ policy for a lot sizing problem for perishable items is discussed. In this policy, Y_t represents the order timing, an indicator variable that is set to one if there is a replenishment of size Q_t in period t. Chapter 3 follows analysing and evaluating parallel implementations for the model presented in the previous chapter. In Chapter 4, another lot sizing model following a dynamic uncertainty strategy is studied: an extension of Silver's heuristic for perishables is confronted with the optimal policies given by a SDP model. The second part of this thesis focuses on studying decision support systems for the maintenance of an OWF. Chapter 5, proposes a decision support system to select a fleet of vessels and to schedule maintenance operations at an offshore wind farm. In Chapter 6, the previous model is extended considering weather circumstances and a broader set of possible patterns of maintenance operations. A heuristic for the operational stage (decisions for scheduling the operations during the time horizon) is discussed. This paper has been submitted recently to an international journal indexed in JCR and it is under peer review revision at the moment. Finally, Chapter 7 summarises the contributions of this thesis.

1.5 Overview of papers

There are five papers presented in this thesis, from Chapter 2 to Chapter 6, including two journal papers, two conference papers and a submitted paper to a journal. The corresponding papers for each chapter are referenced and listed below.

Chapter 2 Reference [28]:

Chapter 3 Reference [34]:

Chapter 4 Reference [36]:

Chapter 5 Reference [35]:

Chapter 6 Reference [30]:

2

On computing order quantities for perishable inventory control with non-stationary demand

[28]

DOI: https://doi.org/10.1007/978-3-319-21407-8_31

Abstract: The determination of order quantities in an inventory control problem of perishable products with non-stationary demand can be formulated as a Mixed Integer Nonlinear Programming problem (MINLP). One challenge is to deal with the β-service level constraint in terms of the loss function. This paper studies the properties of the optimal solution and derives specific algorithms to determine optimal quantities.

3

Accelerating an algorithm for perishable inventory control on heterogeneous platforms

Reference:

[34] Impact factor JCR 2016: 1.93. Q2 (Computer Science, Theory & Methods)

DOI: https://doi.org/10.1016/j.jpdc.2016.12.021

Abstract: This paper analyses and evaluates parallel implementations of an optimization algorithm for perishable inventory control problems. This iterative algorithm has high computational requirements when solving large problems. Therefore, the use of parallel and distributed computing reduces the execution time and improves the quality of the solutions. This work investigates two implementations on heterogeneous platforms: (1) a MPI-PTHREADS version; and (2) a multi-GPU version. A comparison of these implementations has been carried out. Experimental results show the benefits of using parallel and distributed codes to solve this kind of problems. Furthermore, the distribution of the workload among the available processing elements is a challenging problem. This distribution of tasks can be modelled as a Bin-Packing problem. This implies that the selection of the set of tasks assigned to every processing element requires the design of a heuristic capable of efficiently balancing the workload statically with no significant overhead. This heuristic has been used for the parallel implementations of the optimization for perishable inventory control problem.

4

A simple heuristic for perishable item inventory control under non-stationary stochastic demand

[36] Impact factor JCR 2016: 2.325. Q1 (Operations Research & Management Science)

DOI: https://doi.org/10.1080/00207543.2016.1193248

Abstract: In this paper we study the single-item single-stocking location non-stationary stochastic lot sizing problem for a perishable product. We consider fixed and proportional ordering cost, holding cost, and penalty cost. The item features a limited shelf life, therefore we also take into account a variable cost of disposal. We derive exact analytical expressions to determine the expected value of the inventory of different ages. We also discuss a good approximation for the case in which the shelf-life is limited. To tackle this problem we introduce two new heuristics that extend Silver's heuristic and compare them to an optimal Stochastic Dynamic Programming (SDP) policy in the context of a numerical study. Our results demonstrate the effectiveness of our approach.

5

A model for optimal fleet composition of vessels for offshore wind farm maintenance

[35]

DOI: https://doi.org/10.1016/j.procs.2017.05.230
Abstract: We present a discrete optimisation model that chooses an optimal fleet of vessels to support maintenance operations at Offshore Wind Farms (OFWs). The model is presented as a bi-level problem. On the first (tactical) level, decisions are made on the fleet composition for a certain time horizon. On the second (operational) level, the fleet is used to optimise the schedule of operations needed at the OWF, given events of failures and weather conditions.

6

On offshore wind farm maintenance scheduling for decision support on vessel fleet composition

The following chapter is an extension of Chapter 5.

6.1 Introduction

The offshore wind energy industry is expected to continue its growth tendency in the near future. The European Wind Energy Association expects in its Central Scenario by 2030 a total installed capacity of 66 GW of offshore wind in the EU [20]. Offshore wind farms (OWFs) are large scale infrastructures, requiring a large fleet of vessels able to perform operations and maintenance (O&M) tasks on the installed turbines. The O&M constitutes a large part of the costs of running an OWF installation, being up to one third of the OWF costs [79]. Moreover, the fleet makes the installations depend on non-renewable energy resources. Therefore, optimising the efficiency of the resources used for the O&M tasks of an OWF becomes extremely important in order to make them economically viable and to reduce CO2 emissions.

Recent deterministic and stochastic model formulations for vessel composition and optimization of maintenance operations at OWF's can be found in [26] and [37]. A recent literature review on DSS for OWF's is given by [45]. In [16, 48] and [81], a model for maintenance routing and scheduling at offshore wind farms based on the Dantzig-Wolfe decomposition method has been implemented. In that work, a mixed integer linear program is solved for each subset of turbines to generate all feasible routes and maintenance schedules for the vessels for each period. The routes take several constraints into account, such as weather conditions, the availability of vessels and the number of technicians available at the operation and maintenance base. In [82], a two-stage stochastic programming model is presented to determine a cost-optimal fleet size and mix for O&M tasks at offshore wind farms for the total expected lifetime of the OWF. For that, the study considers time periods fixed to three months.

The basis of our investigation is a scenario based MILP model which like the models in [37] and [82] decides on the vessel fleet composition. All these models evaluate the value of the vessel fleet composition and base selection based on scheduling with perfect information; the weather conditions and breakdowns happening during a scenario of a year are known beforehand. The research question in the current paper is whether the vessel fleet composition may be affected when maintenance scheduling is done in a heuristic way following a practical decision rule given the available information at the time of maintenance scheduling.

We investigate this question in the following way. Section 6.2 describes the practical decision problem of operating an OWF and selecting a fleet to support its maintenance tasks. Section 6.3 describes an MILP model, which simultaneously determines the maintenance scheduling as well as the fleet composition. In Section 6.4, a heuristic for the operational stage of the model is presented. Section 6.5 presents a computational study used to compare the outcomes of both procedures. Finally, Section 6.6 summarises our findings.

6.2 Problem definition

This section describes the maintenance planning problem related to a fleet of vessels for an offshore wind farm during a planning horizon, based on a more extensive model of fleet size and mix decisions in [80]. The aim is to find an optimal fleet of vessels and a collection of maintenance tasks to be performed on the wind turbines. That model contains a detailed description of the operational scheduling dealing with each individual action. Our vision also distinguishes periods (shifts) of 12 hours, but aggregates a number of tasks in each period (shift).

Preventive as *corrective* maintenance tasks are considered. Preventive maintenance tasks are meant to prevent failures and prolong the lifetime of wind turbines. Examples include visual inspection, changing of consumables, oil sampling, and tightening of bolts

[60]. Corrective maintenance tasks are needed to repair broken down wind turbines. There is a one-to-one correspondence between failure types and corrective task types.

The number of necessary preventive tasks of each type to be performed is predefined at the beginning of the year. Corrective tasks are only needed after a specific failure occurs in a wind turbine. The planner is confronted in each scenario with failures occurring dynamically. There is a downtime cost associated to the lack of electricity production in turbines during the execution of a maintenance task. Downtime costs are also considered for broken down turbines, incurred for the shifts from diagnose until reparation.

To perform the maintenance tasks, a fleet of vessels is needed. Vessel types have properties such as the type of maintenance tasks they can perform, capacity for transferring technicians, a depreciation cost over the planning horizon, a sailing speed and a threshold for wind speed and wave height that prevents to transfer technicians to the turbines or sailing if they are exceeded. Every vessel is associated to a base, from which it travels to the wind farm to perform maintenance tasks. Each base has a certain vessel capacity, a capacity to accommodate technicians, an associated cost and coordinates which provide its distance to the wind farm.

The decision problem includes a number of candidate bases that can be used and a number of vessel types associated to them. Each vessel type is able to support a particular set of patterns, from the base they are associated with. A pattern consists of one or several maintenance tasks to be performed at the OWF that fits in a shift, including the time it takes the vessel type to perform a round trip visiting the OWF from their base. For each shift the available vessels are able to perform a single pattern of the possible ones that are associated to their type and their base. Some patterns from different vessel types and associated to different bases might be virtually the same, containing the same list of tasks to be performed during the shift. Their cost and time required may vary, considering the speed of the vessel or the distance from their base to the OWF. Some task types do not require the vessel to be present at the turbine. This facilitates performing several tasks in parallel in a single time shift. It is irrelevant whether a pattern contains tasks that run in parallel or sequentially, as long as they meet the time constraints of a shift and the vessel type can accommodate enough technicians to perform the tasks. Moreover, some task types take longer than the time available in a single shift. These long tasks are split into smaller parts that fit with the duration of the shifts. If a long task is initiated in one shift, it does not necessarily have to be continued in the following. However, for corrective tasks, downtime costs are incurred for all shifts until the task is finished and the failure in the turbine has been repaired.

Decisions actually take place on two levels: the first (tactical) level decides which bases to use and which vessels should be available during the planning horizon period under consideration. The second (operational) level schedules operations including which patterns to support by which available vessel in every shift of the planning horizon. The random events the planner is confronted with consist of weather conditions preventing use of vessels for maintenance and the possible failures of turbines that require corrective maintenance tasks.

6.3 MILP model description

Like in the models of [37] and [82], the tactical level decisions are evaluated based on a scenario approach, where the planner has perfect information to schedule the operational maintenance tasks. The following symbols are used to describe the mathematical optimisation model.

Sets

$\mathcal{K}$	Set of bases
$\mathcal{V}_k$	Set of vessel types at base k
$\mathcal{S}$	Set of scenarios
$\mathcal{T}_{vs}$	Set of shifts not suitable for sailing due to weather limitations for vessel type v during scenario s
Γ	Set of maintenance task types
$\mathcal{NP}$	Subset of planned preventive maintenance task types, $\mathcal{NP} \subset \Gamma$
$\mathcal{NC}$	Subset of corrective maintenance task types, $\mathcal{NC} \subset \Gamma$
$\mathcal{P}$	Set of all possible patterns
$\mathcal{P}_{kv}$	Set of possible patterns for a vessel of type v operating from base k

Parameters	
T	Number of shifts in the planning horizon
F_k	Fixed cost per year of operating base k
G_v	Charter or depreciation cost for using a vessel of type v over the complete planning horizon
D_{st}	Income loss due to downtime of performing a maintenance task in scenario s in shift t
H_{ts}	Hourly income loss due to downtime of performing a maintenance task in scenario s in shift t
C_p	Cost of executing pattern $p \in \mathcal{P}_{kv}$ from base k and a vessel of type v
CP_i	Penalty cost for not executing a preventive maintenance task of type $i \in \mathcal{NP}$
N_i	Number of hours required to execute maintenance task of type $i \in \Gamma$ during the planning horizon
PP_i	Number of planned preventive maintenance tasks of type $i \in \mathcal{NP}$
M_k	Number of maintenance technicians available at base $k \in K$ in each shift
MP_p	Required number of maintenance technician personnel to execute pattern p
Q_{kv}	Maximum number of vessels of type v that can operate from base k
B_i	Hours spent on a task of type i in one shift, being $B_i \leq N_i$
A_{ip}	Number of tasks of type i in pattern p
P_s	Probability of scenario s
Y_{its}	Number of failures of type $i \in \mathcal{NC}$ accumulated in shifts. $1, \ldots, t$ in scenario s
Tactical decision variables	
$y_k \in \{0, 1\}$	Equal to 1 if base k is used, 0 otherwise
$x_{kv} \in \{0, \ldots, Q_{kv}\}$	Number of vessels of type v operated from base k
Operational decision variables	
$w_{its} \in \mathbb{Z}^+$	Number of corrective maintenance tasks of type $i \in \mathcal{NC}$ supported during shift t in scenario s
$q_{its} \in \mathbb{Z}^+$	Number of preventive maintenance tasks of type $i \in \mathcal{NP}$ supported during shift t in scenario s
$u_{pts} \in \mathbb{Z}^+$	Number of vessels executing pattern p during shift t in scenario s
$\bar{w}_{its} \in \mathbb{Z}^+$	Number of corrective maintenance tasks of type $i \in \mathcal{NC}$ that are not (yet) completed in scenario s in shift t
$\bar{q}_{is} \in \mathbb{Z}^+$	Number of preventive maintenance tasks of type $i \in \mathcal{NP}$ not completed in scenario s at the end of the planning horizon

In order to solve the model, the bounds on the variables should be set as sharp as possible to facilitate pre-solving operations of an LP solver that filters out those variables that have a value of zero and those constraints that are not binding. We define the following bounds:

$$0 \leq x_{kv} \leq Q_{kv}, \quad \forall k, v \tag{6.1}$$

$$0 \leq u_{pts} \leq \sum_{k \in \mathcal{K}} \sum_{v \in \mathcal{V}_k} Q_{kv} \quad \forall p, s, t \tag{6.2}$$

$$0 \leq w_{its} \leq \sum_{k \in \mathcal{K}} \sum_{v \in \mathcal{V}_k} Q_{kv} \max_{p \in \mathcal{P}_{kv}} A_{ip} \quad \forall i \in \mathcal{NP}, \forall t, s \tag{6.3}$$

$$0 \leq q_{its} \leq \sum_{k \in \mathcal{K}} \sum_{v \in \mathcal{V}_k} Q_{kv} \max_{p \in \mathcal{P}_{kv}} A_{ip} \quad \forall i \in \mathcal{NC}, \forall t, s \tag{6.4}$$

$$0 \leq \bar{w}_{its} \leq Y_{its} \quad \forall i \in \mathcal{NC}, \forall t, s \tag{6.5}$$

$$0 \leq \bar{q}is \leq PP_i \quad \forall i \in \mathcal{NP}, \forall s \tag{6.6}$$

The value of parameter Q_{kv} is an upper bound on the number of vessels that can be used from each base. Therefore, the number of patterns that can be performed in a shift for a particular scenario is bounded by the total capacity of vessels for the considered bases. The number of corrective tasks not finished at a certain shift is bounded above by the total occurrences of failures Y_{its} thus far. The number of preventive tasks not performed at the end of the horizon is bounded by the total number of planned preventive tasks PP_i.

6.3.1 Objective function

The objective is to minimise the fixed costs of operating the bases and the charter cost of the selected vessels, the costs of all performed patterns throughout the planning horizon, the downtime costs associated with the running maintenance tasks or persistent failures and the penalty costs of preventive and corrective task types that are not finished within the planning horizon:

$$\min \sum_{k \in \mathcal{K}} F_k y_k + \sum_{k \in \mathcal{K}} \sum_{v \in \mathcal{V}_k} G_v x_{kv} + \sum_{s \in \mathcal{S}} P_s \left(\sum_{k \in \mathcal{K}} \sum_{v \in \mathcal{V}_k} \sum_{p \in \mathcal{P}_{kv}} \sum_{t=1}^{T} C_p u_{pts} \right) + \tag{6.7}$$

$$\sum_{s \in \mathcal{S}} P_s \left(\sum_{i \in \mathcal{NP}} \sum_{t=1}^{T} H_{ts} B_i q_{its} + \sum_{i \in \mathcal{NC}} \sum_{t=1}^{T} D_{st} \bar{w}_{its} + \sum_{i \in \mathcal{NP}} CP_i \bar{q}_{is} + \sum_{i \in \mathcal{NC}} CP_i \bar{w}_{iTs} \right)$$

The first two terms of the objective function (6.7) cover the costs for the tactical decisions: cost of bases and vessels. The first term refers to the fixed costs for operating the chosen base(s) during the planning horizon. The second defines the charter costs for the available fleet of vessels during the planning horizon.

The following terms cover the expected operational costs of the model. Therefore, the cost of each scenario is multiplied by its probability. The third term of the objective function (6.7) determines the cost of operating the patterns during the planning horizon. Terms four and five describe the downtime costs of preventive and corrective task types, respectively. While the downtime costs for preventive task types are only incurred while a task is taking place on a turbine, downtime for corrective task types initiate from the moment the breakdown occurs and continues until the shift in which it has been repaired. The last two terms are related to penalty costs. Term six is the penalty incurred for the preventive maintenance task types that are not performed within the planning horizon while term seven is the penalty for not finishing all corrective tasks.

6.3.2 Constraints for tactical decisions

There is only one constraint on the tactical level describing the usual relation that base k should be in use, if one wants to station vessels there, and set the bounds of the maximum number of each vessel type for each base.

$$x_{kv} \leq Q_{kv} y_k \qquad \forall k, v \tag{6.8}$$

The tactical decision directly influences the possibilities of the operational planning. A larger fleet allows to perform more patterns each shift.

6.3.3 Constraints on operational decisions

Constraints on the operational level are given by the following inequalities:

$$\sum_{p \in \mathcal{P}_{kv}} u_{pts} \leq x_{kv}, \quad \forall k, v, t, s \tag{6.9}$$

$$\sum_{v \in V_k} \sum_{p \in \mathcal{P}_{kv}} MP_p u_{pts} \leq M_k, \quad \forall k, s, t \tag{6.10}$$

$$\sum_{p \in \mathcal{P}_{kv}} A_{ip} u_{pts} - q_{its} \geq 0, \quad \forall i \in \mathcal{NP}, \forall k, v, s, t \tag{6.11}$$

$$\sum_{p \in \mathcal{P}_{kv}} A_{ip} u_{pts} - w_{its} \geq 0, \quad \forall i \in \mathcal{NC}, \forall k, v, s, t \tag{6.12}$$

$$\frac{N_i}{B_i}(Y_{its} - \bar{w}_{its}) \leq \sum_{\tau=1}^{t} w_{i\tau s} \leq \lceil \frac{N_i}{B_i} Y_{its} \rceil, \quad \forall i \in \mathcal{NC}, \forall s, t \tag{6.13}$$

$$N_i \bar{q}_{is} + \sum_{t=1}^{T} B_i q_{its} \geq N_i PP_i, \quad \forall i \in \mathcal{NP}, \forall s \tag{6.14}$$

$$u_{pts} = 0, \quad \forall p \in \mathcal{P}, \forall t \in \mathcal{T}_{vs}, \forall s \tag{6.15}$$

Constraint (6.9) bounds operations on the availability of sufficient vessels at each base. Each available vessel has the potential to contribute performing one of its possible patterns each shift. Constraint (6.10) limits operations due to available personnel at the bases. Constraints (6.11) and (6.12) link the assignment of individual tasks to planned patterns and availability of vessels, for preventive and corrective types respectively. It is necessary to keep track of not finished corrective tasks every shift, as the turbines affected incur downtime costs until they are fixed. The lower bound of constraint (6.13) keeps track of the number of not finished corrective tasks (breakdowns). The planner cannot complete more corrective tasks than the ones present at the OWF every shift. The upper bound ensures that the number of single type tasks does not exceed the number of broken down turbines caused by the same failure type. Also, a single corrective task of type i contributes with B_i hours during a shift, while $N_i \geq B_i$ hours are needed to complete the a complete task of type i. Therefore, the number $\frac{N_i}{B_i} Y_{its}$ should be rounded up allowing the last part of a large task to be performed. Otherwise, in case just one corrective task of type i is needed and $B_i \nmid N_i$, the task could not be finished until another failure of type i occurs, increasingY_{its}. For preventive tasks it is only necessary to check the number of not performed tasks at the end of the time horizon. Constraint (6.14) keeps track of the number of preventive tasks for each type that have not been finalised in scenario s.

Implicitly, constraints (6.13) and (6.14) imply that the individual task schedule follows from a FIFO approach for preventive and corrective task types, where the first task that has been started is the first to be ended. Such assumption is needed: if each task was treated as an independent task the model would become intractable for small instances. Finally, constraint 6.15 prevents patterns to be performed during shifts in which the weather conditions exceed the threshold of wind speed or wave height for the vessel type used to execute the pattern.

6.3.4 Generating columns (bundles and patterns) for the model

The basic decisions of the scheduler of the maintenance operations are based on feasible patterns, previously crafted by the decision maker. This section describes an automatic procedure to generate the feasible maintenance patterns for every base and vessel type combination. A recursive algorithm can be used for this task, considering constraints such as the number of technicians needed or the time limitations to complete the pattern.

As sketched in Section 6.2, some task types do not require the vessel to be present at the turbine during the operation such that these types can be run in parallel. We will indicate them by the set Γp_v. Therefore, the generation of columns is based on a two-step procedure. First, we generate the feasible set of bundles of tasks that include only task types from Γp_v that can be run in parallel using a recursive procedure described in Algorithm 1. In the second step, we combine these bundles with the non-parallel task types in Γn_v to create the final set of patterns.

A bundle b is specified as a quadruple (List, Time, Cost, Tech) specifying the list of activites, the time to execute it, the cost and the number of technicians required, respectively. Each task in List is performed at a different turbine and they are run in parallel. During the execution of a bundle, the vessel docks to the first turbine, offloads the task materials and technicians required and then moves to another turbine until all the tasks are started. When finished, the vessel recollects the technicians and returns to base. The duration (Time) of a bundle consists of the set up time (setupTime_i) for its tasks and the docking time (docktime_v) at each turbine when dropping off and when picking up the technicians. With respect to the time B_i spent on a single task, we have to keep in mind they are run in parallel. The procedure has to take into account the number of technicians Tech_v allowed on vessel v and the number of hours TMX_v it may stay at the Offshore wind farm. Finally, the cost of a task Cost_i and the number of required technicians Tech_i is updated.

Algorithm 1 build_bundle(Bundle b, Vessel v)

```
for all i ∈ Γp_v do
    T_vi= 2*dockTime_v+setupTime_i;
    temp_time=max(Time+T_vi,B_i+T_vi)
    temp_cost= Cost+Cost_i
    temp_tech= Tech + Tech_i
    temp_list by adding i to List
    if temp_time ≤ TMX_v and temp_tech ≤ Tech_v then
        define new bundle b̂=(temp_list,temp_time,temp_cost,temp_tech)
        𝓑_v = 𝓑_v ∪ {b̂}
        build_bundle(b̂,v)
    end if
end for
```

The procedure, starting with $b = (\emptyset, 0, 0, 0)$ builds a set $\mathcal{B}_v$ of bundles of tasks for each vessel using the set of tasks Γ specified for each vessel Γ_v. The lists of the bundles are unordered with repetitions of the same task types. To create a sharp set description, a dominance procedure is run over the bundle sets. Let List_1 and List_2 be such that $\text{List}_1 \subseteq \text{List}_2$, then the bundle with List_1 is removed.

After Algorithm 1 is run for every vessel type and the dominance procedure is performed to each vessel type bundle set, they can be used to build the final pattern set. Algorithms 2 and 3 use the dominated bundle sets $\mathcal{B}_v$ and the non-parallel task types to derive the pattern sets $\mathcal{P}_{kv}$. Since the time to arrive to the OWF depends on the cruising speed of the vessel types and the distance from the departure base to the OWF, pattern sets are associated with a base and a vessel type. Algorithm 2 goes over each base and vessel type combination starting by adding the time of a return trip to the OWF and the fuel costs as a base for the time and the costs of the patterns of each vessel type and base combination. Then, patterns are build recursively in Algorithm 3, which is called from Algorithm 2 $|B_v|$ times (existing bundles) and $|\Gamma n_v|$ times, i.e. the number of non-parallel task types.

Algorithm 2 Generate Patterns

for all $k \in \mathcal{K}$ and $v \in \mathcal{V}_k$ **do**
  determine Time from 2*(distance to OWF)/(vessel speed of v);
  determine Cost from 2*(distance to OWF)*(CostFuel per km of v);
  p=pattern($\emptyset$, Time, Cost, 0)
  for tasks/bundles n in Γn_v and $\mathcal{B}_v$ **do**
    build_pattern(p,n)
  end for
end for

return $\mathcal{P}_{kv}$

6.4 Operational scheduling based on available information

In this Section, we discuss a heuristic (a scheduler) for the operational stage of the model, where a plan should be generated for every shift of a particular scenario, depending on the weather constraints. In contrast to the MILP approach, no anticipation of the weather conditions and the failures in the turbines is taken. With respect to the available information, the parameters of the stochastic failure events are assumed not to be known. For the weather realisations, only the monthly averages of wind speed are known, based on historic weather data. At the beginning of each shift, the weather conditions for it and the new failures are realised.

The scheduler consists of two parts. First, the part called OWFscheduler harvests and deals with the available information. The second part (heuristic) evaluates the possible patterns to be performed for every shift, deciding which of the available vessels to use and which patterns to perform. At the beginning of each shift, the OWFscheduler administrates the occurring failures and the weather circumstances for the current shift. This is fed to the heuristic with the decisions for the current shift. The scheduler administrates the incurred costs for the performed pattens and the number of hours invested in each task type at the OWF.

6.4.1 The scheduler

The scheduler starts by taking the realisation of the weather circumstances, including wind speed, wind_t, and wave height, wave_t, for the current shift t.

Algorithm 3 build_pattern(Pattern p, Activity n)

```
if n ∈ Γn_v (vessel needed at turbine) then
  temp_time=Time+2*dockTime_v+setupTime_n+actTime_n
  temp_cost= Cost+ cost_n
  temp_tech= Tech + Tech_n
  temp_list= List ∪{n}
  if temp_time ≤ v.TMX then
    new pattern π := (temp_list, temp_time,temp_cost,temp_tech)
    P_kv = P_kv ∪ π
    build_pattern(π,n + 1)
  end if
else
  temp_time= Time+Time_n, i.e. n is a bundle
  temp_cost= Cost+Cost_n
  temp_tech= Tech+Tech_n
  temp_list= List ∪ List_n
  if temp_time ≤ v.TMX then
    w=(temp_list, temp_time, temp_cost, temp_tech)
    P_kv = P_kv ∪ {w}
    build_pattern(w,n + 1)
  end if
end if
```

From these values, the vessels that cannot sail during shift t are discarded via the data maxWind$_v$ and maxWave$_v$ for the maximum wind speed and wave height supported by each vessel type. Let $\mathcal{VP}_t$ represents the list of vessels that execute perform patterns during shift t. The scheduler puts observed new failures occurring at shift t on a stack. The downtime costs due to corrective failures is administrated considering the number of failures at the OWF after the operations have taken place, matching with the MILP formulation in Section 6.3.

The scheduler calls the heuristic which chooses the patterns to be performed (associated to one of the available vessels operating from its base) and updates the remaining time of the tasks with respect to shift t. The scheduler continues after the heuristic finishes executing, moving to the next shift or, at the end of the time horizon, calculating

Algorithm 4 OWFscheduler

for $i \in \mathcal{NP}$ **do**
  RemainHours$_i = PP_i \times N_i$
end for
for $t \in \{1, \ldots, 2T\}$ **do**
  Observe realised wind$_t$ and wave$_t$; $\mathcal{VP}_t = \emptyset$
  for v with a k for which $x_{kv} > 0$ **do**
    $\mathcal{VP}_t = \mathcal{VP}_t \cup \{v\}$ if wind$_t$ <maxWind$_v$ AND wave$_t$ < maxWave$_v$
    Add observed failure type i to DownAct$_i$
    Update downtime costs
  end for
  Call Heuristic
end for
Calculate total cost

and returning the final cost for the operational stage.

6.4.2 The heuristic

The heuristic chooses the patterns to be performed during the current shift t, $\mathcal{P}_t \subset \mathcal{P}$. It considers the available vessels $\mathcal{VP}_t$, the state of the needed tasks RemainHours$_i$ (which resembles the number of needed hours to finish all of the tasks of type i) and the realisation of the weather data for the current shift t,in order to evaluate the downtime costs. The set $\mathcal{P}_t$ is fed from the existing patterns for the vessels that, after realising the weather conditions, are able to sail, $\mathcal{VP}_t$. For that, procedure patternfitness, gives a fitness value f_p for each of the possible patterns to be performed during the shift.

The heuristic selects the pattern with the minimum fitness value f_p and compares it with the cost value of not performing more patterns during the current shift, captured in parameter idleCost. This parameter reflects the downtime cost for the turbines that are down due to a failure, the only cost that applies during a particular period in case no patterns are performed. If a minimum fitness pattern is selected, the heuristic updates the corresponding downtime cost for preventive tasks and the number of remaining hours for every task type, RemainHours$_i$, are reduced. Moreover, the vessel used by pattern r is removed from $\mathcal{VP}_t$ and the set of possible patterns is updated, removing

Algorithm 5 Heuristic

Set $\mathcal{P}_t$ of possible patterns given ($t, \mathcal{VP}_t$, DownAct$_i$,RemainHours$_i$)
Determine fitness f_p for each pattern $p \in \mathcal{P}_t$
Find $r = \arg\min_{p\in\mathcal{P}_t} f_p$
Determine IdleCost
while patterns possible and IdleCost$< f_r$ **do**
  for Choosen pattern r and tasks $i \in$ List$_r$ **do**
    Update downtime costs for $i \in \mathcal{NP}$;
    RemainHours$_i$ =RemainHours$_i - B_i * A_{ir}$
    Update DownAct$_i$ for $i \in \mathcal{NC}$
    Remove the used vessel and update $\mathcal{P}_t$ correspondingly
    Update $f_p, p \in \mathcal{P}_t$, idleCost and r
  end for
end while

from $\mathcal{P}_t$ the remaining patterns associated with the chosen vessel. This process continues until there are no available pattern in $\mathcal{P}_t$. In practice this means that all of the available vessels have been chosen to either perform one of their patterns or staying at their bases.

6.4.3 Procedure patternfitness

For a greedy like heuristic that iteratively selects the most promising pattern, the evaluation of the fitness f_p based on the available (updated) information is essential. In our case, the following information can be weighted.

- Pattern cost: the cost of performing pattern p
- Downtime costs for turbines that are shut down to perform the tasks included in pattern p
- Potentially saved penalty costs SPC_p obtained by performing pattern p
- Saved downtime costs $Scosts_p$ of performing pattern p

The sum of this costs give the fitness f_p of pattern p. Including pattern costs and downtime costs for shut down turbines is straightforward. The calculation of SPC

and *Scosts* is detailed in the next paragraphs. Unlike the first two terms, SPC and *Scosts* are negative. SPC_p refers to the proportion of the penalty costs based on the

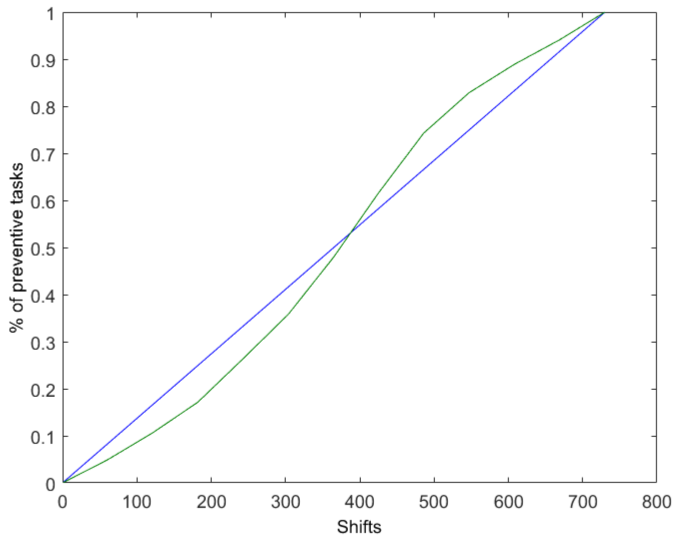

Figure 6.1: Linear and monthly average approaches to guide scheduling preventive tasks

time pattern p spends on its different tasks types. If, for instance, one of the tasks of p constitutes $\beta\%$ of the total time of a particular task type, the $\beta\%$ of the penalty cost of that task type is counted. However, the penalty for not performing preventive tasks during the year is rather high and patterns consisting on those tasks could be overly chosen. One way to proceed could be to limit the number of preventive tasks that can be scheduled linearly with time. In that case, the number of preventive tasks of type i that can be finished up by shift t would be $\frac{PP_i t}{2T}$. However, gains can be obtained by scheduling more tasks when, seasonally, wind speed is expected to be slower. Performing preventive tasks when the wind speed is low saves downtime costs due to performing preventive tasks, as reflected by parameter D_{st}. In practice, the weather conditions are not known to the planner beforehand, but the scheduler has insight in the monthly average conditions for wind speed, based on historic data. From these averages it can be derived the expected power loss for a single turbine during month τ, which we assume to be captured by the average values w_τ. One can normalise these values to obtain the proportion of expected loss of month τ respect to the total year: $\bar{w}_\tau = \frac{w_i}{\sum_{j=1}^{12} w_i}$. One would like to perform in month τ a fraction of the total planned preventive tasks that is inversely proportional to the number $\bar{w}_\tau$. This fraction is given by $\varphi_\tau = \frac{1}{\bar{w}_\tau} \frac{1}{\sum_{j=1}^{12} \bar{w}_j}$, dividing again by the total summation of $\bar{w}_j$ to normalise results. Figure 6.1 depicts the accumulative values of φ_t interpolated between the month averages confronted with a linear approach. Consequently, parameter BehindSch is determined for each $i \in \mathcal{NP}$ according to

$$BehindSch_i = \max((1-\varphi_t)PP_i - \lceil \frac{RemainHours_i}{N_i} \rceil, 0)$$

For corrective task types $i \in \mathcal{NC}$, we focus on the unfinished work

$$BehindSch_i = DownAct_i$$

Finally, to have a valuation SPC_p for the pattern, we can add penalties p_i leading to

$$SPC_p = \frac{t}{2T} \sum_{i \in \Gamma} p_i A_{ip} \frac{B_i}{N_i} Behindsch_i,$$

Notice, this factor has a higher weight in the fitness when approaching the end of the horizon. However, if $Behindsch_i = 0$ the valuation of SPC_p is set to zero. If $Behindsch_i = 0$ and for patterns that include one or more preventive type tasks, the wind speed is checked. In case it is lower than the expected average, the fitness valuation of p is different: the pattern costs are not considered for the fitness and the number of hours of type i that that pattern performs is calculated and subtracted from the fitness value.

The term *scosts* is a negative value that refers to the potential savings in electricity production if a break down turbine is repaired. After calculating the contribution of pattern $p \in \mathcal{P}_{kv}$ for each corrective task type $i \in \mathcal{NC}$ in terms of number of tasks performed, *scosts* accumulates the value:

$$scosts = -nact(i)(2T - t)ydown$$

The term *ydown* represents the average downtime cost of a single turbine for the total duration of the planning horizon. Notice, this factor gets a higher weight at the beginning of the planning horizon that than at the end.

6.5 Computational illustration

In the MILP model, the lower level operational planning cost provides a lower bound for the incurred cost due to failures and downtime. By formulating the operational tasks in a one-shot model, in principle all the scenario is known beforehand and earlier tasks can be planned based on knowledge of failures that will occur later, i.e. anticipation is allowed. This makes the planning in principle cheaper than what is possible in reality. On the other hand, due to the nature of the variable $\bar{w}_{its}$, there is a tension in the optimal outcome to start repairing a failure as soon as possible by a corrective task.

In this section, we discuss the amount of underestimation for specific realistic data confronting the optimal MILP outcome of the lower level for scenarios with the heuristic decision rule defined in Section 6.4. The model and the heuristic have been compared for an instance similar to the one published in [35].

The MILP has been modelled for the bi-level model using GAMS interface [2], and solved using the CPLEX solver, setting the optimality gap at 1%.

6.5.1 A case study

We consider an OWF consisting of 125 turbines. The planning horizon is one year and the periods represent 12 hour shifts and include a return trip from the base the vessel is located to the OWF and a bundle of activities. In practical terms there are 730 periods. There are three available bases B_1, B_2, B_3 around the OWF, each of which can accommodate up to 48 technicians and they are located at 110, 61 and 86 kilometers respectively from the OWF. The cost of using each of them, for the entire time horizon is 2, 6 and 7 million monetary units (mu) respectively.

Four types of vessels are considered: V_1, V_2, V_3, V_4. Each base has space to allow two vessels of type V_1, two of type V_2, four of type V_3 and one of type V_4. Vessel type V_4 is able to accommodate up to 30 technicians, while the rest has space for only 12. The cost of having a vessel during the whole planning horizon for vessel types V_1, V_2, V_3 and V_4 is, respectively, 122,4000, 2,500,000, 750,000 and 7,200,000 mu. Vessel types V_1 and V_2 can travel at a speed of 20 knots, while vessel types V_3 and V_4 can travel at 40 knots. In practical terms this means that vessel types V_1 and V_2 require about 5.94, 3.3 and 4.64 hours to perform a return trip between bases B_1, B_2 and B_3 respectively while vessel types V_3 and V_4 would require half of that time, allowing more time to perform activities in each shift.

There are two preventive activity types A_1, A_2 and two corrective activity types A_3, A_4. All vessel types are able to perform all the activity types considered. Activity type A_4 requires the vessel supporting the operation to be present at the turbine while the activity is performed, whereas activity types A_1, A_2, A_3 can be run in parallel. The vessel drops a group of technicians at each turbine that is going to be supported during the shift. The time required to perform activity types A_1, A_2, A_3 and A_4 is 60 , 100 , 3 and 7.5 hours respectively. The maximum time per period and turbine that a group of technicians can support an activity type is 6 hours. Consequently, only activities of type A_3 can be performed in a single period. The penalty cost for not executing a preventive activity type is 10 million mu. For corrective activities of type A_3, the cost is 50,000 mu, and for type A_4, the penalty cost rises to 500,000 mu. The patterns for each combination of base and vessel type are generated following the procedure described in Algorithms 1 and 3.

For our case study, there are 125 planned activities of type A_1 and 60 of type A_2. The number of corrective activity types corresponds to the number of failures of the turbines and depends on the scenario. A scenario consists of the events of the failures of the turbines and the weather conditions for every period. Failures that require corrective activity types A_3 and A_4 follow a binomial distribution. The rate of failures for a corrective activity of type A_3 is 5 times per turbine per year, and 3 times per turbine a year for failures that require an activity of type A_4. Weather conditions are taken from historical weather data. For each scenario, a report file containing a year of wind speed and wave height data of the OWF area is picked for feeding these variables.

6.5.2 Discussion of results

For comparing the performance of the heuristic for the operational stage with the optimal solution of the MILP problem, two different tactical stage decisions have been considered. The first one is the optimal solution for the MILP, (S1), which consists of using three vessels of type V_3 from base B_1. An instance consisting on a tactical stage decision using less vessels than the optimal MILP solution based on perfect information, would not generate significant results. The second instance (S2) consists of using four vessels of type V_3 from base B_1.

A set of 20 scenarios have been generated. For each scenario, the heuristic has been run and the MILP problem has been solved for the tactical stage decisions studied, (S1) and (S2) the optimal solution for the MILP; using three vessels of type V_3 from base B_1 and (S2). Table 6.1 presents the average value of the 20 scenarios for the total cost, executing patterns cost, preventive and corrective downtime costs and operational stage costs for tactical stage decisions S1 and S2, running the heuristic and solving the MILP problem. Preventive and corrective penalty costs are not included in Table 6.1, since they result to be zero for the generated scenarios.

Table 6.1: Associated costs for the MILP optimal solution and the heuristic for tactical decisions S1 and S2

	Total	Pattern	P. D.	C. D.	Op. S. Cost
MILP S1	10986350	5060220	1117923	558265	6736408
MILP S2	11472400	5126880	1028245	314890	6470015
HEUR. S1	13401952	5346330	2296092	1509528	9151951
HEUR. S2	12958671	5435595	1235124	1287951	7958671

The MILP complete information solution for S1 has a total cost of nearly 11 million monetary units (MU), while the heuristic for S1 has a cost of 13.4 million MU. Considering only the costs of the operational stage, the MILP complete information solution is 6.73 million MU, while the heuristic reaches 9.15 million MU. Downtime costs for corrective tasks is about 2.7 times higher than the MILP cost. For preventive tasks the cost doubles that of the MILP solutions. This shows that the heuristic does not perform well for S1 with the optimal MILP perfect information setting. In a real setting, when failures and weather conditions are uncertain, that tactical decision might not be optimal.

However, for S2 the deviation between the two solutions is quite different. The MILP complete information solution for the operational stage is 6.47 million MU, reducing only slightly the costs of S1. However, the heuristic reduces that cost to 7.95 million MU, and this reduction comes mostly by handling preventive tasks much better, reducing the cost by half compared to the tactical decision S1. It can be observed that the downtime cost for corrective tasks, incurred by broken down turbines until they are repaired, is the only cost significantly higher for the heuristic compared to the MILP solution costs, for both tactical decisions S1 and S2. This can be explained considering that the MILP model has exact information for when the failures occur for all the periods of the problem, being able to anticipate corrective tasks early in time. In contrast, the cost of performing the patterns, which constitutes the major cost of operating the OWF and is not related with uncertain events, is only 6% above the one provided by the MILP.

6.6 Conclusion

Models in the literature on selecting an optimal vessel fleet composition for operations and maintenance tasks at OWFs during a planning horizon typically apply a complete information approach to evaluate the fleet composition. The models are confronted with weather conditions and turbine failures. Weather conditions may prevent vessels sailing and execute tasks at the OWF, while turbine failures result in new corrective maintenance tasks. However, weather conditions and failures are unknown in practice. Therefore, a deterministic complete information approach to find the optimal solution for the operational stage only provides a lower bound on the maintenance costs in the operational stage. In the current paper, a similar MILP model for the fleet composition is presented. The question is: What are the costs if the scheduler applies a heuristic rule based on the information available in practice? This means, the heuristic is not based on perfect information, realising the weather and failure events at the beginning of each shift. The results show that the heuristic performs well when the tactical decisions include enough vessels to cover the demand of O&M activities at the OWF and allows for slack in the scheduling compared to the optimal complete information plan. Although the performance costs of the heuristic for the chosen scheduling are only 6% above the optimal lower bound, for the corrective tasks, where (stochastic) failures have to be repaired, the cost is about four times higher than that given by the MILP. This illustrates the effect of anticipation in a perfect information situation. The value of evaluating the fleet composition in a realistic setting is that probably the chosen vessel plan will contain more vessels, as this facilitates recourse actions on random events.

7 Conclusions

This chapter summarises the findings of the works presented in this thesis and shows guidelines for possible future research.

We can distinguish two main areas of research in this thesis. The first one dealt with lot sizing problems for perishable items. The second, with the optimization of the maintenance processes at offshore wind farms. Each of them provided findings for the research questions investigated in this thesis, related to dynamic decision making.

The first research question of this thesis is to study which order policies are the most appropriate for a general lot sizing problem for a perishable item. Can the optimal policy be found? Chapter 2 describes that problem and discusses a solution method. The model studied is a lot sizing problem for a single perishable item following a static strategy over a finite time horizon and considering ordering, holding, unit and waste costs. Demand is stochastic and non-stationary, a β-service level for the demand is considered and a FIFO issuing policy is followed. For this problem setting theoretical properties were derived of the optimal solution aiding a possible solution approach. Moreover, a specific algorithm exploiting these properties has been developed to find an YQ policy derived for the static-dynamic case, using Monte Carlo simulation to estimate the expected value of the inventory levels. This policy proved to be optimal for the static-dynamic case. By enumerating the feasible timing order policies, the optimal solution for the static strategy can be found by finding the optimal quantities for each replenishment schedule.

However, this solution method requires a high computational demand due to the Monte Carlo simulation and enumeration of replenishment schedules. This led to a new research question: How can the use of parallel computing improve the performance of the algorithms to find solutions for the problem?

Two implementations were developed for heterogeneous platforms: a multi-GPU version using CUDA and a multicore version using Pthreads and MPI. For the multi-GPU version, the Monte Carlo simulation (the most demanding task of the solution method) was parallelised using the CUDA interface. For the multicore version, the initial approach for a multi-core parallelisation suggested an embarrassingly parallel implementation, as finding the optimal quantities for each replenishment schedule are in fact independent subproblems. However, randomly dividing the workload among processors led to an unbalanced workload for the processors. The reason behind this is that, rather than using Monte Carlo, an exact analytical approach to calculate the inventory levels for a cycle can be followed for the periods in which the inventory levels are equal to zero. Calculating the order quantities for a cycle analytically is much faster than using Monte Carlo simulation, even when using small samples. After identifying the computational workload of each replenishment schedule subproblem, balancing the workload among processors resembles a bin-packing problem. Three fast heuristics with reduced overhead were proposed for the bin-packing problem. The one that procured the best balancing in the testbed was used for parallelising the multicore version of the lot sizing problem.

The last research question concerning lot sizing problems involved considering the use of heuristics for such problems: to which extend does the use of heuristics give good results in a DDM lot sizing problem for perishable products? In this case, a similar lot sizing model was used. The difference with the former studied model is to use a penalty cost for unfulfilled demand, rather than using a β-service level and allowing backordering. For this model, exact analytical expressions to compute the expected value of the inventory for different product ages were found, assuming the product can age indefinitely. This derived an analytical approximation for the inventory levels for the case in which the product age is discrete and finite. From here, an extension of Silver's heuristic has been developed for the case of perishable products, introducing an analytical and a simulation-based variant of the approach. An SDP model for the problem was derived as well, in order to compare the optimal solution with the heuristic. Results showed that the simulation approach featured an average cost performance only 5% above the optimal cost. For the analytical approximation this figure was 6%.

A second practical dynamic decision making problem was studied for continuing the research in this thesis. The topic was related to optimising the maintenance activities at a OWF. Chapters 5 and 6 address this problem. The first research question is addressed in Chapter 5: is an MILP model suitable for an application for selecting a fleet of vessels to support the maintenance at offshore wind farms? A discrete, scenario based and deterministic optimisation model was derived, presented as a bi-level problem: On the first level, decisions are made in order to select a fleet of vessels. On the second level, the fleet is used to optimise the schedule of O&M activities at a OWF, dealing with turbine failure events and considering weather conditions that may prevent performing activities for safety reasons. The model solves instances of more than 700 periods (The planning horizon is set to one year, deciding activities in twelve hours shifts). However, such a model is based on perfect information for failure events and weather conditions, since adding non-anticipative constraints would leave the model unsolvable. Previous models in literature, aiming to solve long time horizon problems using small time periods ([37], [82]) are deterministic as well. This situation set new research questions, as the vessel fleet composition may be affected when maintenance scheduling is done when the failures events and the weather conditions are uncertain.

The next research question was: Is it possible to find an efficient and realistic metaheuristic for scheduling O&M activities at offshore wind farms with failures and weather uncertainty? What are the differences with a perfect information MILP model? Chapter 6 discusses a heuristic that schedules the O&M activities and shows to what extend a non-anticipative method affects the optimal solution. The experiments show that the optimal fleet composition given by the MILP model is not sufficient to operate the OWF efficiently using the heuristic. However, adding only an extra vessel provides a fair solution: the cost of performing the activities, which constitutes the major cost of operating the OWF and is not related with uncertain events, is only 6% above the one provided by the MILP. In contrast, the cost associated with the downtime in turbines due to failure events, which are uncertain for the heuristic, are about three times higher than the given by the perfect information MILP model.

Appendix I: Publications arising from this thesis

The research work carried out for the present thesis resulted in a number of publications. This appendix lists them along with their respective quality indicators and sorted by their year of publication (oldest first) within each category.

Publications in International Journals (JCR)

[36] Impact factor JCR 2016: 2.325. Q1 (Operations Research & Management Science)

[34] Impact factor JCR 2016: 1.93. Q2 (Computer Science, Theory & Methods)

Publications submitted to International Journals (JCR)

[30]

Publications in International Journal (not JCR)

[35]

Publications in Book Chapters

[28]

Publications in International Conferences

[29]

[33]

[32]

[31]

[19]

Publications in National Conferences

[27]

Appendix II: Other publications produced during the elaboration of this thesis

The research effort invested during the time span in which this thesis was elaborated produced additional publications as the result of other research lines not included in the present dissertation. Those lines were Perishable Inventory Control and Procedural Content Generation using population based algorithms. This appendix lists them along with theirs respective quality indicator.

Publications in International Journals

[62] Impact factor JCR 2016: 2.22. Q1 (Operations Research & Management Science), Q1 (Economics and econometrics), Q1 (Industrial and Manufacturing Engineering)

Publications in Proceedings of International Conferences with DOI

[52]

[42]

Publications in other International Conferences

[63]

Resumen en español

Introducción

Esta tesis analiza aplicaciones de toma de decisiones dinámica para un conjunto de problemas. Pueden diferenciarse dos líneas principales. La primera trata problemas de gestión de la cadena de suministro para productos perecederos, mientras que la segunda estudia el diseño de flotas de embarcaciones para realizar labores de mantenimiento en parques eólicos marinos. Los modelos de inventario para productos perecederos estudiados en esta tesis consideran un único producto, una única localización de suministro y una planificación de producción sobre un horizonte de tiempo finito.

Los principales objetivos de esta tesis son los siguientes: (1) estudiar que políticas de pedido son las más apropiadas para los problemas de tamaño de lote. ¿En qué casos una política de pedido da una solución óptima?; (2) analizar el efecto del uso de computación paralela para mejorar el rendimiento de los algoritmos derivados y así diseñar políticas para problemas de tamaño de lote de productos perecederos; (3) explorar cómo de efectivas pueden ser las heurísticas para problemas de toma de decisiones dinámica sobre el tamaño de lote de productos perecederos; (4) elaborar un modelo MILP para seleccionar una flota de embarcaciones con el fin de realizar las operaciones de mantenimiento en parques eólicos marinos; y (5) diseñar una heurística para programar las operaciones de mantenimiento en parques eólicos marinos considerando fallos en turbinas e incertidumbre meteorológica.

En el primer capítulo de esta tesis se realiza una introducción a la teoría de control de inventarios y se justifican las motivaciones que han llevado a cabo el desarrollo del trabajo que se incluye en esta tesis. Los capítulos posteriores tratan independientemente cada uno del los objetivos anteriormente mencionados. En el segundo capítulo, un modelo de programación estocástica es presentado para un problema práctico de planificación de producción de un producto perecedero en un horizonte de tiempo finito. Una política estática es estudiada para el modelo. Tal política ha demostrado ser óptima asumiendo una estrategia de incertidumbre estática, que es considerada para instancias con un tiempo de espera largo. El tercer capítulo trata el uso de computación paralela para los algoritmos desarrollados en el capítulo previo. Dos implementaciones fueron desarrolladas sobre plataformas heterogéneas: una versión multi-GPU usando CUDA y una versión multicore usando Pthreads y MPI. Para la primera implementación, la simulación de Monte Carlo (la tarea más costosa), es paralelizada. Ambas implementaciones mostraron una buena escalabilidad. El cuarto capítulo trata la efectividad de heurísticas para problemas de tamaño de lote de productos perecederos similar. La clásica heurística de Silver es extendida para productos perecederos y se presentan variantes del procedimiento: una analítica y una basada en simulación. Los resultados de la heurística son comparados con las soluciones óptimas dadas por un modelo SDP (Stochastic Dynamic Programming) generado para el problema, mostrando que los costes de las heurísticas presentan, de media, un 5% sobre el coste óptimo para la estrategia basada en simulación y un 6% para la aproximación analítica. En el quinto capítulo, se presenta un modelo MILP para seleccionar la flota de embarcaciones óptima para el mantenimiento de un parque eólico marino. El modelo se presenta como un problema de dos niveles, seleccionando la flota óptima en el primer nivel y optimizando la selección de las operaciones, usando dicha flota, en el segundo. Dado que el modelo es determinístico, como otros en la literatura que aspiran a resolver problemas con un horizonte temporal largo usando periodos cortos, el sexto capítulo trata la cuestión de cómo la anticipación de los eventos estocásticos como los fallos en las turbinas o las condiciones meteorológicas afectan la decisión de la flota de embarcaciones óptima. Este capítulo presenta una heurística que ilustra este efecto.

Resumen del capítulo 2

La base de las implementaciones que se presentan en este capítulo es un algoritmo desarrollado en Matlab para resolver un problema MINLP (Mixed Integer NonLinear Programming). Se trata de planificar, a lo largo de un número finito de periodos T, las cantidades que se deben proveer de cierto producto perecedero para satisfacer la demanda bajo una restricción que establece un nivel de servicio β que necesariamente se debe satisfacer. En concreto, esta restricción establece que para cada periodo (siempre hablando en términos de esperanza matemática) a lo sumo una fracción β de la demanda no pueda ser satisfecha y sea perdida por falta de stock, ya que se supone que esta no puede ser servida en un periodo posterior. Esta condición es equivalente a que al menos una fracción $(1-\beta)$ de la demanda sea cubierta en cada periodo. La duración de cada item producido desde que está disponible para el consumidor hasta que ha de ser retirado es de $J < T$ periodos. Además, se supone que los productos se distribuyen siguiendo la regla FIFO: los productos son expedidos comenzando por los más antiguos.

El problema de optimización que se plantea es el de encontrar la cantidad de producto perecedero que hay que producir en cada periodo de forma que se satisfgan todas las restricciones del problema y que además se minimice una función coste. A continuación se detallan las principales variables del modelo:

Índices

t	índice del periodo, $t = 1, \ldots, T$, siendo T el número total de periodos
j	índice de edad, $j = 1, \ldots, J$, siendo J la vida útil de cada unidad

Parámetros

$\boldsymbol{d}_t$	demanda en cada periodo con distribución normal dada por su media $\mu_t > 0$ y varianza $(cv \times \mu_t)^2$ dado por un coeficiente de variación cv, idéntico en cada periodo
k	coste por periodo en el que se decide realizar un pedido, $k > 0$
c	coste unitario de producto, $c > 0$
h	coste por almacenamiento, $h > 0$
w	coste unitario de desecho, puede ser negativo con la condición, $w > -c$
β	nivel de servicio, $0 < \beta < 1$

Variables

$Q_t \geq 0$	cantidad de producto producido y disponible en el periodo t. Denotamos por Q al vector completo $(Q_1, \ldots, Q_T)$
$\mathrm{Y}_t \in \{0, 1\}$	indica si se produce un pedido en el periodo t. Es 1 si y solo si $Q_t > 0$. Denotamos por Y al vector completo $(Y_1, \ldots, Y_T)$
$\boldsymbol{X}_t$	ventas perdidas en el periodo t
$\boldsymbol{I}_{jt}$	inventario de edad j al final del periodo t, considerando un periodo inicial fijo, $I_{j0} = 0$, $\boldsymbol{I}_{jt} \geq 0$ para $j = 1, \ldots, J$

Además, se usará la notación $(\cdot)^+ = max(\cdot, 0)$.

La función coste que se pretende minimizar depende del vector $Q = (Q_1, \dots, Q_T)$ y se puede definir como:

$$f(Q) = \sum_{t=1}^{T} \left(C(Q_t) + E\left(h \sum_{j=1}^{J-1} \boldsymbol{I}_{jt} + w \boldsymbol{I}_{Jt} \right) \right), \tag{7.1}$$

siendo

$$C(x) = k + cx, \quad \text{if} \;\; x > 0, \;\; \text{and} \;\; C(0) = 0 \tag{7.2}$$

El nivel de inventario para cada periodo $t = 1, \dots, T$ y cada edad j siguiendo la regla FIFO puede calcularse como sigue:

$$\boldsymbol{I}_{jt} = \begin{cases} \left(Q_t - (\boldsymbol{d}_t - \sum_{j=1}^{J-1} \boldsymbol{I}_{j,t-1})^+ \right)^+ & j = 1, \\ (\boldsymbol{I}_{J-1,t-1} - \boldsymbol{d}_t)^+ & j = J, \\ \left(\boldsymbol{I}_{j-1,t-1} - (\boldsymbol{d}_t - \sum_{i=j}^{J-1} \boldsymbol{I}_{i,t-1})^+ \right)^+ & otro\ j \end{cases} \tag{7.3}$$

Por otra parte, la restricción del nivel de servicio puede expresarse como:

$$E\left(\boldsymbol{X}_t\right) \leq (1-\beta)\mu_t, \;\; t = 1, \dots, T \tag{7.4}$$

Para controlar el cumplimiento de esta restricción es necesario calcular las ventas perdidas que se producen en cada periodo t, lo cual viene dado por:

$$\boldsymbol{X}_t = \left(\boldsymbol{d}_t - \sum_{j=1}^{J-1} \boldsymbol{I}_{j,t-1} - Q_t \right)^+ \tag{7.5}$$

El valor esperado de las ventas perdidas es una función conocida como *loss-function* que en general no admite una expresión en términos elementales. Algunas aproximaciones factibles pueden verse en [51, 71, 74, 87]. Para nuestro modelo hemos decidido utilizar la simulación Monte Carlo para obtener una estimación de la *loss-function.* Con las condiciones impuestas, el problema de encontrar las cantidades de producto perecedero que se deben producir en cada periodo y que minimizan la función coste $f(Q)$ dada en (7.1), y con ello la política $Y \in \{0,1\}^T$ de periodos de pedido óptima, es un problema MINLP. Como veremos, la técnica usada presenta características adecuadas para su implementación en computadores de alto rendimiento.

Resumen del capítulo 3

El objetivo de este capítulo consiste en determinar hasta que punto el uso de una arquitectura heterogénea (multicore-multiGPU) facilita la resolución de un problema de optimización del control de inventarios de productos perecederos. Pretendemos aprovechar la capacidad computacional de estas arquitecturas para obtener soluciones mas exactas, para ejemplos mas pesados desde el punto de vista de la computación, manteniendo tiempos de respuesta aceptables. El problema del control de inventarios queda definido a lo largo de una serie finita de T periodos de tiempo en los que se ha de satisfacer la demanda (estocástica) de un determinado producto perecedero que desde que se produce tiene una vida útil de J periodos. En el modelado de este problema se supone que la distribución se realiza siguiendo la política de distribución FIFO, entregando el producto demandado con mayor antigedad. Se supone, además, que la demanda que no se satisfaga en un periodo queda perdida, no pudiéndose acumular al periodo siguiente. La solución a este problema consiste en encontrar que cantidades de pedido a lo largo de todos los periodos resulta óptima, en el sentido de minimizar el coste asociado a la producción, distribución, almacenamiento y desecho de los productos que sobrepasen su vida útil.

Actualmente, las arquitecturas de computación de altas prestaciones más extendidas son las plataformas heterogéneas basadas en sistemas de memoria distribuida, donde cada nodo tiene una arquitectura multicore que podría albergar un número distinto de cores [44]. Por lo tanto, las implementaciones paralelas tienen que ser adaptadas para poder ser ejecutadas en dichas arquitecturas heterogéneas. En este contexto, es necesario tener un conocimiento detallado tanto del algoritmo a paralelizar como de los recursos computacionales que se van a utilizar para la implementación [53]. Además, a estas arquitecturas se les pueden incorporar aceleradores, como son FPGAS, GPUs, coprocesadores Intel Xeon Phi, etc. En concreto, en el problema del control de inventario para productos perecederos se ha optado por la combinación de clústeres de Multi-GPUs. De este modo, el uso de plataformas masivamente paralelas (GPUs) permite la aceleración de las tareas computacionalmente más costosas, porque estas unidades tienen mucha potencia de cálculo para los esquemas de computación vectorial. De forma adicional, el uso de plataformas de memoria distribuida permite obtener unos resultados más precisos debido a que el uso de computación paralela permite incrementar el número de simulaciones realizadas para resolver un caso particular sin que el tiempo de ejecución se incremente.

El modelo de computación paralela asociado a este problema se puede describir en términos de un conjunto de tareas que no presentan dependencias entre sí. Sin embargo, la carga computacional de cada una de estas tareas es variable y por lo tanto pueden aparecer problemas de desbalanceo de la carga si se hace un reparto de la carga a ciegas. Este problema de asignación de tareas a elementos de procesamiento se conoce en la literatura de complejidad como problema de Bin packing [24]. Dado que es un problema NP-Completo, se han desarrollado varias heurísticas que permiten tener una solución en un tiempo razonable.

El orden de complejidad del problema, partiendo del algoritmo secuencial, está relacionado con el número vectores Y (que indica los periodos en los que se realiza un pedido) posibles, lo cual depende de los valores de J y T. Independientemente del valor de J, el número de casos posibles a tratar aumenta de forma exponencial con el valor de T, es decir, $O(e^T)$. Más aún, la complejidad para hallar las cantidades óptimas de cada vector Y depende del número de veces que es necesario recurrir a simulación de Monte Carlo, limitada a T en cada caso. Cada vez que se realiza la simulaión, el inventario y las ventas perdidas son calculadas para N casos independientes. Por tanto, el orden de complejidad para el método completo, es decir, hallar el vector de pedidos Y óptimo y las cantidades óptimas de pedido, es aproximadamente del orden de $O(N \cdot T \cdot e^T)$.

En la sección anterior se ha puesto de manifiesto la necesidad de realizar simulaciones para obtener aproximaciones de la función *floss*. Esta función es la que consume la mayor parte del tiempo computacional de la ejecución del problema de inventarios. Es importante destacar la necesidad de realizar un elevado número de simulaciones para que las aproximaciones que lleva a cabo la función *floss* sean suficientemente exactas. Por tanto, para realizar aproximaciones relativamente precisas del valor de las ventas perdidas (X), es necesario realizar un número N de simulaciones del problema suficientemente alto, lo que constituye la verdadera carga computacional del problema. Como caso de estudio, se ha considerado un ejemplo del problema de control de inventarios en el que $T = 15$ y $J = 3$, que es bastante realista. Dicho ejemplo genera un total de 5768 vectores Y posibles.

Una vez analizada la estructura algorítmica del problema, pasamos a describir los detalles de las implementaciones que hemos llevado a cabo sobre una arquitectura heterogénea formada por un clúster de Multi-GPUs (multicores y dispositivos GPUs). El hecho de explotar una plataforma heterogénea de un clúster tiene dos ventajas fundamentales: poder abordar la resolución de problemas de mayor tamaño y reducir el tiempo de ejecución de un caso concreto. Las implementaciones consideradas en este trabajo han sido:

- MPI-PTHREADS: Esta implementación obtiene el paralelismo de los procesadores multicore y de los nodos disponibles en el clúster. Para ello, se utiliza programación basada en hebras [12] y MPI [78].

- Multi-GPU: Esta implementación está basada en el uso de GPUs para realizar las simulaciones de Monte Carlo, las cuales son la parte computacionalmente más costosa del problema a resolver. Para ello, la interfaz de programación que se utiliza es CUDA.

Centrando nuestra atención en la implementación MPI-PTHREADS, se ha explotado el paralelismo en dos niveles: a nivel de nodo (memoria distribuida) y a nivel de multicore (memoria compartida). Por un lado, existen múltiples formas de paralelizar rutinas en modelos de memoria compartida, aunque la librería estándar es Pthreads (POSIX threads). Pthreads provee un conjunto unificado de rutinas en una librería de C cuyo principal objetivo es facilitar la implementación de threads o hilos en el programa. Por otro lado, debido a su portabilidad, MPI ha sido el interfaz considerado para explotar el paralelismo a nivel de nodo.

Partiendo del algoritmo de optimización del problema de inventarios, se ha realizado una paralelización híbrida (MPI y Pthreads), en la cual el conjunto de vectores Y que se van a evaluar en el problema de optimización son repartidos entre los procesadores de acuerdo a las heurísticas de balanceo de la carga propuestas. La evaluación de esta implementación se ha realizado en un clúster Bullx y los resultados se describen en las figuras 3.3 y 3.4 del capítulo 3.

El reparto inicial de la carga de trabajo entre los procesadores disponibles puede considerarse como un problema de Bin packing con algunas restricciones. El problema de Bin packing se enmarca dentro de la optimización combinatoria (NP-completo), y en nuestro caso se puede modelar de la siguiente forma: Dado un conjunto de E ejecuciones independientes del Algoritmo *MinQ()* (items), cada una de ellas con una carga computacional $0 < w_i < B$ y dado un conjunto de P procesadores (Bins), repartir las ejecuciones del algoritmo entre los procesadores de forma que la carga computacional máxima asignada a un procesador sea mínima (ver [24] para una formulación general del problema de Bin packing).

Debido a la dificultad de encontrar soluciones óptimas para este tipo de problemas, habitualmente se utilizan técnicas heurísticas y metaheurísticas, que son capaces de encontrar una solución aceptable en un tiempo razonable. Algunas de estas heurísticas están inspiradas en computación evolutiva [8]. Para resolver el problema de balanceo de la carga que se ha modelado como un problema de tipo Bin packing, proponemos tres algoritmos heurísticos (H1, H2 y H3) para repartir la carga de trabajo (en nuestro caso, los posibles vectores Y) entre todos los elementos de procesamiento disponibles de forma que se minimice el tiempo de ejecución del problema de optimización.

1. (H1): Heurística basada en Round Robin: Ordenando previamente, de mayor a menor, el peso de las tareas a asignar, estas se reparten entre los P procesadores siguiendo el patrón $(1, \ldots, P, P, P-1, \ldots, 1, 1, \ldots)$

2. (H2): Heurística basada en asignar sucesivamente los items w_i al procesador que menos carga de trabajo haya acumulado.

3. (H3): Similar a la heurística H2, pero previamente ordenando los items de mayor a menor carga.

Para valorar las heurísticas se han utilizado tres instancias del problema denominadas γ_1, γ_2 y U, en las que la carga computacional estimada que se asocia a cada vector Y es diferente. (γ_1) y (γ_2) están basadas en distribuciones gamma con parámetros de forma y escala (10,4) y (1,25), respectivamente, y (U) sigue una distribución uniforme con valores entre 0 y 100.

Para medir el grado de balanceo de la carga asignada a cada procesador, se ha utilizado el coeficiente de Gini (G), ampliamente utilizado en el campo de la economía para medir el grado de desigualdad de la distribución de la riqueza en poblaciones [25]. Este índice varía entre 0 (equidad absoluta) y 1 (un solo individuo (procesador) posea toda la riqueza de la población (carga computacional)). G se define como la media de la diferencia entre cada posible par de procesadores, divididos por su carga media. Para un número de ejecuciones E asignadas a P elementos de proceso, siendo w_i la carga computacional asignada al procesador i, ordenadas de forma ascendente, G se calcula como sigue:

$$G = \frac{2\sum_{i=1}^{P} i \cdot w_i}{P\sum_{i=1}^{P} w_i} - \frac{P+1}{P} \tag{7.6}$$

Gráficamente, G representa el ratio entre la diferencia del área rodeada por la línea de uniformidad y la curva de Lorenz de la distribución, y el área triangular que hay debajo de la línea de uniformidad. G toma valores entre un mínimo de 0, cuando todos los procesadores tienen la misma carga, a un máximo de 1, cuando todos los procesadores (excepto uno) tienen una carga de cero. Por lo tanto, cuando G se acerca a 0 la carga está bien balanceada, y cuando se acerca a 1 está desbalanceada.

En la Tabla 3.1 se resume el comportamiento de las diferentes heurísticas a través del valor del coeficiente de Gini (G) para los ejemplos planteados (comparándolos con un reparto a ciegas. Claramente se demuestra en esta tabla que la heurística H3 es, al menos, un orden de magnitud mejor que las heurísticas H1 y H2 y que el reparto aleatorio de las tareas entre los procesadores (HR) es al menos dos ordenes de magnitud peor que H3 y un orden de magnitud peor que H1 y H2. De los datos de la Tabla 3.1, se concluye que la heurística H3 es la que presenta mejores resultados, consiguiendo balancear la carga de forma casi exacta, por lo tanto esta es la heurística que produce mejores tiempos de ejecución en la evaluación de las implementaciones paralelas del problema del control de inventarios.

La versión Multi-GPU se ha basado en la explotación de diversas GPUs para la paralelización de las simulaciones del método de Monte Carlo, realizadas por la función *flossGPU*. Por una parte, cada una de las N simulaciones son independientes entre sí. Al mismo tiempo, para el cálculo de todo el inventario de cada una de las posibles edades (7.3), este solo depende del inventario del periodo anterior. Por tanto, separando por periodos, un kernel de CUDA puede realizar en paralelo el cálculo de las N simulaciones y, al mismo tiempo, la actualización del inventario de J edades diferentes. Por tanto, la computación que se realiza con la GPU es la simulación de Monte Carlo. Al mismo tiempo, el modelo se ha implementado de forma que cada proceso MPI abre uno o dos hilos, que a su vez abren una o dos GPUs del nodo en el que se encuentra. Las tareas se reparten usando la heurística (H3) entre las GPUs que se vayan a utilizar en cada caso.

La función *flossGPU* resume el cambio realizado en el Algoritmo *floss(q,a,b)* para adaptarlo a su ejecución en una o varias GPU. Tal y como se ha mencionado, el bucle que recorre los periodos se sitúa en el primer nivel.

Para la evaluación de las implementaciones paralelas hemos utilizado un clúster compuesto de ocho nodos Bullx R424-E3 Intel Xeon E5 2650 (cada uno con 16 cores), interconectados por un puerto InfiniBand QDR/FDR embebido en la placa madre, 8-GB RAM y 16-GB SSD) con ocho GPUs TeslaM2075 (de los ocho nodos, cuatro de ellos tienen dos GPUs por nodo). El driver de CUDA que se ha utilizado es CUDA 6.5. La arquitectura Multi-GPU y las características de las GPUs se muestran en la Figura 3.2.

En cuanto a la paralelización MPI-PTHREADS, en la Figura 3.4 se puede apreciar un buen nivel de speed-up. La rapidez con la que las heurísticas se ejecutan las hace apropiadas incluso para problemas en los que el desbalanceo no es muy acusado, mejorando un reparto aleatorio de las tareas. Para casos en los que el desbalanceo es mucho mayor (ejemplos con las distribuciones gamma y uniforme) el beneficio es mucho mayor y se observa que la heurística (H3) presenta mejores resultados.

Con respecto a la implementación Multi-GPU, en la Tabla 3.3 se recogen los tiempos de ejecución del problema completo, tomando hasta las 8 GPUs existentes en el clúster para la paralelización del método Monte Carlo en CUDA. En dicha tabla se aprecia como se pueden conseguir buenos resultados paralelizando una escasa porción del código. En cambio, al usar múltiples GPUs, la escalabilidad está penalizada por el tiempo de inicialización de las GPUs (aproximadamente 5 segundos), lo cual hace que no se obtenga un buen rendimiento con más de 2 GPUs para este ejemplo concreto. Para el ejemplo de inventarios considerado, el uso de los dieciséis cores de un solo nodo resulta más beneficioso que el uso de las dos GPUs disponibles. En definitiva, la escalabilidad con el uso de múltiples GPUs se ve limitada por el tiempo de inicialización requerido, aunque podría ser beneficiosa en comparación con la paralelización MPI-PTHREADS si el problema a tratar requiere más precisión, siendo necesarias más simulaciones del método de Monte Carlo, de forma que el tiempo de inicialización de las GPUs se hiciera comparativamente irrelevante.

Resumen del capítulo 4

Mientras que en el capítulo 2 se describía un problema de control de inventarios para productos perecederos y un método para encontrar su solución óptima, el capítulo 3 explotaba el uso de computación paralela en plataformas heterogéneas basadas en sistemas de memoria distribuida para acelerar la resolución del problema. No obstante, dicho método se ve limitado por el hecho de incrementar, de forma exponencial, el cómputo necesario para hallar la solución óptima según aumenta el número de periodos del problema.

En este capítulo se proponen heurísticas que encuentran soluciones cuyo coste, de media, es solo un 5% superior a al óptimo. Más en concreto, este capítulo realiza las siguientes contribuciones a la literatura de control de inventarios no perecederos con demanda estocástica y no estacionaria:

- Se introducen expresiones analíticas exactas para realizar el cálculo del valor esperado de inventario de diferentes edades cuando el producto puede envejecer indefinidamente; estas expresiones sirven tanto para distribuciones discretas como continuas para la demanda.

- Se derivan aproximaciones analíticas para el caso en el que la edad límite de los productos es discreta y finita.

- Utilizando estos resultados, se propone una extensión de la heurística de Silver[75] para el caso concreto de productos perecederos; en particular introduciomos una variación analítica y otra basada en simulación para el procedimiento.

- Se realiza un estudio computacional con un extenso conjunto de datos que prueban que las heurísticas propuestas encuentran soluciones cuyo coste, de media, es solo un 5% superior al óptimo.

En este caso consideramos un problema de producción de un único producto, una única localización de suministro y una planificación de producción sobre un horizonte de T periodos. El producto considerado es perecedero y su edad, en periodos, se denota por $a \in \{1, \ldots, A\}$; $a = 1$ denota los productos nuevos recién llegados al principio del periodo actual. Al final de un periodo dado t, todos los productos de edad A son descartados; consideraremos $A < T$ para asegurar que el carácter perecedero de los productos afecta al modelo.

La demanda es estocástica y no estacionaria, es decir, su distribución varía entre periodos. La demanda en el periodo t es una variable aleatoria no negativa D_t con función de distribución conocida F_t. La distribución de estas variables aleatorias se supone independiente entre los periodos. Los productos se distribuyen siguiendo la regla FIFO. La demanda que no pueda satisfacerse no se considera perdida en este caso, sino que es satisfecha en el periodo siguiente. Suponemos que el tiempo de entrega de los productos es cero, aunque las heurísticas propuestas pueden adaptarse facilmente al caso en el que el tiempo de suministro es mayor.

Existe un coste fijo por ordenar un pedido o y un coste v proporcional a la cantidad de pedido solicitada; un coste por almacenamiento h por cada producto que es llevado de un periodo al siguiente, independientemente de su edad; se incurre un coste de penalización p por cada unidad de demanda no satisfecha en cada periodo; un coste de desecho w por cada producto de edad A que sea descartado al final de cada periodo. El objetivo es encontrar una política de suministro que minimice el coste total esperado, que está compuesto por los costes de pedido, los costes de almacenamiento y los costes de penalización y desecho, a lo largo del horizonte de T periodos planificado.

Consideremos el inventario neto como el inventario almacenado menos la posible cantidad de demanda no satisfecha. El modelo supone que los eventos en cada periodo se suceden como se describe a continuación. Al comienzo de un periodo el inventario de distintas edades es observado. Si es necesario se produce un pedido de producto por la cantidad deseada. Tras esto, la demanda es observada y los niveles de inventario son actualizados siguiendo la regla FIFO. Tras esto, los productos de edad A que queden en stock son descartados y se incurre un coste de desecho por ellos. Si por el contrario el inventario neto fuese negativo se incurre un coste de penalización por la demanda no satisfecha. La Tabla 4.1 del capítulo 4 recoge la notación de parámetros y otras variables usada durante este capítulo.

Los lemas 4.1-4.5 del capítulo 4 muestran como obtener expresiones analíticas para la esperanza de los diferentes niveles de inventario para el caso en el que el producto pueda envejecer indefinidamente. Esto sirve de base para obtener aproximaciones analíticas para el caso de productos perecederos.

Nuestra heurística hace uso de estos resultados para calcular, dado un inventario inicial $\mathbf{I}_{t-1}$, los niveles de inventario $\mathrm{E}(\mathbf{I}_t)$ durante el ciclo $t \in \{t, \ldots, r\}$. Usando el lema 4.6 calcula la cantidad de pedido óptima Q_t para el ciclo (t, r) así como la esperanza de los costes totales por periodo.

Como en la heurística de Silver, se incrementa el valor de r, empezando por t, hasta que la esperanza de los costes totales por periodo asociados al ciclo (t, r) aumenta por primera vez. Sea $r+1$ tal valor. La acción óptima en el periodo t es pedir una cantidad Q_t que minimiza la esperanza de los costes totales para el ciclo (t, r).

La tabla 4.2 muestra un resumen de los resultados obtenidos en nuestro estudio computacional. En dicha tabla se observa como la heurística que hace uso de simulación se comporta, por lo general, mejor que la aproximación analítica. No obstante, esta última es 10 veces más rápida que la primera. De media, la heurística analítica encuentra soluciones cuyo coste es, de media, 5.96% por encima del coste óptimo para las 54 instancias, mientras que la heurística que hace uso de simulación reduce esta diferencia al 4.76%.

Resumen de los capítulos 5 y 6

El problema de toma de decisiones para programar las operaciones de mantenimiento en parques eólicos marinos es tratado como un problema de cadena de suministro: la instalación requiere programar operaciones de mantenimiento y atender los fallos en turbinas durante el horizonte planificado. Una flota de embarcaciones tiene que ser seleccionada para realizar estas operaciones. Para este conjunto de problemas, las decisiones no son solo dinámicas, sino que además se realizan bajo incertidumbre.

El problema de la planificación del mantenimiento de un parque eólico marino mediante una flota de embarcaciones propuesto en los capítulos 5 y 6 está basado en un modelo descrito en [80]. El propósito es encontrar la flota óptima de embarcaciones y una colección de actividades de mantenimiento en las turbinas eólicas. El modelo contiene una descripción detallada de la planificación de las operaciones relativas a cada acción individual.

Se consideran actividades de mantenimiento preventivas y correctivas. Las actividades preventivas son aquellas que están destinadas a prolongar la vida útil de las turbinas eólicas y a prevenir fallos. Las actividades correctivas son aquellas destinadas a resolver fallos en las turbinas. Existe una corresponcencia biunívoca entre los posibles fallos en las turbinas y los tipos de actividades correctivas del modelo.

El número de actividades preventivas de cada tipo que debe ser realizado durante la planificación temporal está predefinido de antemano y este tipo de actividades pueden realizarse en cualquier periodo siempre que las condiciones meteorológicas lo permitan. Sin embargo, las actividades correctivas solo pueden realizarse desde el periodo en el que un determinado fallo ocurrre en una turbina eólica. Los fallos en turbinas se presentan por escenarios. Existe un coste por inactividad relativo a la pérdida de producción energética en las turbinas durante la ejecución de cualquier tipo de actividad de mantenimiento. Asimismo, se consideran costes por inactividad para las turbinas averiadas, que se incurre hasta que se produce la reparación.

Para realizar las actividades de mantenimiento es necesaria una flota de embarcaciones. Los diferentes tipos de embarcaciones tienen propiedades como la clase de actividades que pueden realizar, capacidad para transportar a los técnicos, un coste de depreciación anual, una determinada velocidad de crucero y un límite para el nivel de velocidad del viento y oleaje para los que es seguro navegar. Cada embarcación está asociada a una base, desde la que viaja al parque eólico para realizar las actividades. Cada base tiene una determinada capacidad de embarcaciones, técnicos, un coste asociado y el valor de la distancia hacia el parque eólico.

El problema de decisión incluye un número de bases posibles y un número de tipos de embarcaciones asociadas a ellas. Cada tipo de embarcación es capaz de realizar un determinado conjunto de patrones de actividades de mantenimiento desde la base a la que está asociada. Un patrón consiste en una o varias actividades de mantenimiento que serán realizadas en el parque eólico durante un periodo, incluyendo el tiempo que conlleva realizar un viaje de ida y vuelta de la base al parque eólico. En cada periodo las embarcaciones disponibles pueden realizar un patrón de los disponibles asociados al tipo de embarcación y a la base. Algunos patrones de distintas embarcaciones y asociados a distintas bases pueden ser virtualmente los mismos, conteniendo la misma lista de actividades a ser realizadas durante el periodo. El coste y el tiempo puede variar, considerando la velocidad de crucerod e cada embarcación o la distancia entre la base y el parque. Algunos tipos de actividades no requieren que la embarcación esté presente durante la actividad. Esto permite que varias actividades puedan ser ejecutadas paralelamente en un mismo periodo. Es irrelevante si un patrón contiene actividades que pueden ejecutarse en paralelo o no, siempre y cuando cumplan con las restricciones de tiempo durante un periodo y la embarcación permita transportar al número necesario de técnicos para realizar todas las actividades. Más aún, algunos tipos de actividades pueden llevar más del tiempo disponible en un periodo. Estos tipos son cortados en pequeñas partes que puedan ser realizadas durante los periodos. Si una actividad larga es iniciada en un determinado periodo, no es necesario que sea continuada en los siguientes. Sin embargo, para actividades de tipo correctivo, los costes por inactividad son incurridos en todos los periodos hasta que la actividad es finalizada y la turbina queda reparada.

Las decisiones del modelo tienen lugar en dos niveles: en un primer nivel (táctico) se deciden las bases y las embarcaciones que van a ser usadas durante la planificación temporal. El segundo nivel (operacional) programa las operaciones, incluyendo los patrones que realizará cada embarcación disponible durante cada uno de los periodos. Los eventos aleatorios incluyen las condiciones meteorológicas que pueden prevenir el uso de las embarcaciones y los posibles fallos en turbinas que tienen lugar durante la planificación.

La formulación matemática del modelo del capítulo 5 se encuentra recogida en la sección 5.3. Esta formulación matemática no incluye las restricciones meteorológicas que prevén el uso de embarcaciones. Es en el capítulo 6 en la sección 6.3 donde la formulación matemática del problema MILP (Mixed Integer Linear Programming) es refinada, incluyendo estas nuevas restricciones. Del mismo modo, el capítulo 6 incluye, en la sección 6.3.4, una descripción de los algoritmos recursivos usados para generar automáticamente todos los patrones posibles para cada combinación de base y tipo de embarcación considerando las restricciones del problema.

El capítulo 6 incluye una heurística para la fase operacional del modelo. Esta heurística no realiza anticipación sobre los eventos aleatorios como la formulación MILP y es por tanto más realista. En resumen, la heurística consiste en un planificador de tareas que observa, al comienzo de cada periodo, los eventos de los nuevos fallos en turbinas y las condiciones meteorológicas. En función de ellos, toma decisiones sobre las embarcaciones a usar durante cada periodo y los patrones a realizar. Para ello, evalúa cada una de las posibles decisiones según una función de fitness y selecciona las decisiones de menor coste, incluyendo el estado ocioso de las embarcaciones.

Bibliography

[1] A guide to UK Offshore Wind Operations and Maintenance. URL: `https://www.scottish-enterprise.com/knowledge-hub/articles/guide/offshore-wind-operations-and-maintenance-opportunities`.

[2] GAMS development corporation. GAMS documentation 24.7. `https://www.gams.com/help/index.jsp`, 2016. Accessed: 2017-10-15.

[3] P. Amorim, H. Meyr, C. Almeder, and B. Almada-Lobo. Managing perishability in production-distribution planning: a discussion and review. *Flexible Services and Manufacturing Journal*, 25(3):389–413, Sep 2013.

[4] R.G. Askin. A Procedure for Production Lot Sizing with Probabilistic Dynamic Demand. *A I I E Transactions*, 13(2):132–137, 1981.

[5] M. Bakker, J. Riezebos, and R.H. Teunter. Review of inventory systems with deterioration since 2001. *Eur J Oper Res*, 221(2):275–284, September 2012.

[6] J. Bauer and J. Lysgaard. The offshore wind farm array cable layout problem: A planar open vehicle routing problem. *J Oper Res Soc*, 66(3):360–368, 2015.

[7] R. Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[8] C. Blum and A. Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Comput. Surv.*, 35(3):268–308, September 2003.

[9] S. Bollapragada and T.E. Morton. A simple heuristic for computing nonstationary (S, S) policies. *Oper. Res.*, 47(4):576–584, April 1999.

[10] J.H. Bookbinder and J.Y. Tan. Strategies for the Probabilistic Lot-Sizing Problem with Service-Level Constraints. *Manage Sci*, 34(9):1096–1108, 1988.

[11] A. Burkimsher, I. Bate, and L. Indrusiak. Scheduling HPC Workflows for Responsiveness and Fairness with Networking Delays and Inaccurate Estimates of Execution Times. In *Euro-Par 2013*, volume 8097 of *Lecture Notes in Computer Science*, pages 126–137. Springer Berlin Heidelberg, 2013.

[12] D.R. Butenhof. *Programming with POSIX Threads*. Professional Computing Series. Addison-Wesley, 1997.

[13] O. Caliskan-Demirag, Y.F. Chen, and Y. Yang. Ordering policies for Periodic-Review inventory systems with Quantity-Dependent fixed costs. *Oper Res*, 60(4):785–796, August 2012.

[14] X. Chen, Z. Pang, and L. Pan. Coordinating inventory control and pricing strategies for perishable products. *Oper Res*, 62(2):284–300, April 2014.

[15] S. Chowdhury, J. Zhang, A. Messac, and L. Castillo. Optimizing the arrangement and the selection of turbines for wind farms subject to varying wind conditions. *Renewable Energy*, 52:273–282, 2013.

[16] L. Dai, M. Stålhane, and I.B. Utne. Routing and scheduling of maintenance fleet for offshore wind farms. *Wind Engineering*, 39(1):15–30, 2015.

[17] W. Edwards. Dynamic decision theory and probabilistic information processing. *Human Factors*, 4(2):59–74, 1962.

[18] O. Ellabban, H. Abu-Rub, and F. Blaabjerg. Renewable energy resources: Current status, future prospects and their enabling technology. *Renewable and Sustainable Energy Reviews*, 39(Supplement C):748 – 764, 2014.

[19] E.M.T. Hendrix and A. Gutierrez-Alcoba, I. García, D. Haugland, E.E. Halvorsen-Weare, and G. Ortega. On generating a fleet composition for offshore wind farm maintenance operations. In N. Kliewer et al, editor, *OR2017*, page 69, 2017.

[20] European Wind Energy Association. Wind energy scenarios for 2030. http://www.ewea.org/fileadmin/files/library/publications/reports/EWEA-Wind-energy-scenarios-2030.pdf. Accessed: 2017-10-15.

[21] A. Federgruen and M. Wang. Inventory Models with Shelf-Age and Delay-Dependent Inventory Costs. *Operations Research*, 63(3):701–715, 2015.

[22] D.G. Feitelson. Workload modeling for performance evaluation. In *Performance Evaluation of Complex Systems: Techniques and Tools, Performance 2002, Tutorial Lectures*, pages 114–141, London, UK, 2002. Springer-Verlag.

[23] L. Feng, J. Zhang, and W. Tang. Optimal inventory control and pricing of perishable items without shortages. *IEEE T Autom Sci Eng*, 13(2):918–931, 2016.

[24] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman & Co., New York, USA, 1979.

[25] C. Gini. Measurement of inequality of incomes. *Economics Journal*, 31:124–126, 2014.

[26] C. Gundegjerde, I.B. Halvorsen, E.E. Halvorsen-Weare, L. Magnus, and L. Magne. A stochastic fleet size and mix model for maintenance operations at offshore wind farms. *Transportation Research Part C: Emerging Technologies*, 52:74 – 92, 2015.

[27] A. Gutierrez-Alcoba, E.M.T. Hendrix, I. García, and G. Ortega. Implementaciones paralelas para un problema de control de inventarios de productos perecederos. In *Actas XXVI Jornadas de Paralelismo*, pages 161–168, 2015. 978-84-16017-52-2.

[28] A. Gutierrez-Alcoba, E.M.T. Hendrix, I. García, G. Ortega, K.G.J. Pauls-Worm, and R. Haijema. On computing order quantities for perishable inventory control with non-stationary demand. In O. Gervasi et al, editor, *Computational Science and Its Applications. ICCSA 2015*, volume 9156 of *Lecture Notes in Computer Science*, pages 429–444, 2015.

[29] A. Gutierrez-Alcoba, E.M.T. Hendrix, I. García, K.G.J. Pauls-Worm, and R. Haijema. On computing order quantities for perishable inventory control with nonstationary demand. In *Proceedings of the XII Global Optimization Workshop MAGO14*, pages 5–8, 2014. 978-84-16027-57-6.

[30] A. Gutierrez-Alcoba, E.M.T. Hendrix, G. Ortega, E.E. Halvorsen-Weare, and D. Haugland. On offshore wind farm maintenance scheduling for decision support on vessel fleet composition. *Submitted to an international peer reviewed journal*, pages 1 – 20, 2017.

[31] A. Gutierrez-Alcoba, E.M.T. Hendrix, G. Ortega, D. Haugland, and E.E. Halvorsen-Weare. On location and vessel fleet composition for offshore wind farm maintenance. In M. Baldomero-Naranjo et al, editor, *VIII International workshop on locational analysis and related problems*, pages 39 – 40, 2017.

[32] A. Gutierrez-Alcoba, E.M.T. Hendrix, G. Ortega, K.G.J. Pauls-Worm, and R. Haijema. Perishable inventory control with a service level constraint and non stationary demand. In *Proceedings of the XIII EuroPT Workshop on Advances in Continuous Optimization*, pages 45–46, 2015.

[33] A. Gutierrez-Alcoba, E.M.T. Hendrix, and R. Rossi. Piecewise linearisation of the first order loss function for families of arbitrarily distributed random variables. In *Proceedings of the XXVII European Conference on Operations Research*, pages 298–298, 2015.

[34] A. Gutierrez-Alcoba, G. Ortega, E.M.T. Hendrix, and I. García. Accelerating an algorithm for perishable inventory control on heterogeneous platforms. *J. Parallel Distrib. Comput*, 104(Supplement C):12 – 18, 2017.

[35] A. Gutierrez-Alcoba, G. Ortega, E.M.T. Hendrix, E.E. Halvorsen-Weare, and D. Haugland. A model for optimal fleet composition of vessels for offshore wind farm maintenance. In *Procedia Computer Science*, volume 108(Supplement C), pages 1512 – 1521, 2017.

[36] A. Gutierrez-Alcoba, R. Rossi, B. Martín-Barragan, and E.M.T. Hendrix. A simple heuristic for perishable item inventory control under non-stationary stochastic demand. *Int. J. Prod. Res.*, 55(7):1885–1897, 2017.

[37] E.E. Halvorsen-Weare, C. Gundegjerde, I.B. Halvorsen, L. Magnus, and L. Magne. Vessel fleet analysis for maintenance operations at offshore wind farms. *Energy Procedia*, 35:167 – 176, 2013.

[38] F.W. Harris. How many parts to make at once. *The Magazine of Management*, 10(2):135–136, 1913.

[39] R. Hedjar, M. Bounkhel, and L. Tadj. Predictive control of periodic-review production inventory systems with deteriorating items. *Top*, 12(1):193–208, 2004.

[40] J. Helsen, G.D. Sitter, and P.J. Jordaens. Long-term monitoring of wind farms using big data approach. In *2016 IEEE Second International Conference on Big Data Computing Service and Applications (BigDataService)*, pages 265–268, March 2016.

[41] E.M.T. Hendrix, R. Haijema, R. Rossi, and K.G. Pauls-Worm. On solving a stochastic programming model for perishable inventory control. In *Computational Science and Its Applications ICCSA 2012*, volume 7335 of *Lecture Notes in Computer Science*, pages 45–56. Springer Berlin Heidelberg, 2012.

[42] E.M.T. Hendrix, K.G.J. Pauls-Worm, R. Rossi, A. Gutierrez-Alcoba, and R. Haijema. A sample-based method for perishable good inventory control with a service level constraint. In *ICCL2015: Computational Logistics*, volume 9335 of *Lecture Notes in Computer Science*, pages 526–540. Springer International Publishing, Cham, 2015.

[43] E.M.T. Hendrix and B.G. Tóth. *Introduction to Nonlinear and Global Optimization*. Springer, New York, 2010.

[44] J.L. Hennessy and D.A. Patterson. *Computer Architecture: A Quantitative Approach*. Morgan Kaufmann, 2011.

[45] M. Hofmann. A review of Decision Support Models for offshore wind farms with an Emphasis on Operation and Maintenance Strategies. *Wind Engineering*, 35:1–16, 2011.

[46] S.H. Hung, M.Y. Tsai, B.Y. Huang, and C.H. Tu. A Platform-Oblivious Approach for Heterogeneous Computing: A Case Study with Monte Carlo-based Simulation for Medical Applications. In *Proceedings of 2016 FPGA*, pages 42–47, New York, NY, USA, 2016. ACM.

[47] D.L. Iglehart. Optimality of (s,s) policies in the infinite horizon dynamic inventory problem. *Manage Sci*, 9(2):259–267, 1963.

[48] C.A. Irawan, D. Ouelhadj, D. Jones, M. Stålhane, and I.B. Sperstad. Optimisation of maintenance routing and scheduling for offshore wind farms. *Eur J Oper Res*, 256(1):76–89, 2017.

[49] I. Karaesmen, A. Scheller-Wolf, and B. Deniz. *Planning Production and Inventories in the Extended Enterprise*, volume 151 of *International Series in Operations Research & Management Science*, chapter 15. Managing Perishable and Aging Inventories: Review and Future Research Directions, pages 393–436. Springer, 2011.

[50] I.Z. Karaesmen, A. Scheller-Wolf, and B. Deniz. Managing perishable and aging inventories: Review and future research directions. In Karl G. Kempf, Pınar Keskinocak, and Reha Uzsoy, editors, *Planning Production and Inventories in the Extended Enterprise*, volume 151 of *International Series in Operations Research & Management Science*, pages 393–436. Springer US, 2011.

[51] M.H. Kurawarwala. Forecasting and inventory management of short life-cycle products. *Oper Res*, 44:131–150, 1996.

[52] R. Lara-Cabrera, A. Gutierrez-Alcoba, and A.J. Fernández-Leiva. A spatially-structured pcg method for content diversity in a physics-based simulation game. In Giovanni Squillero and Paolo Burelli, editors, *EvoApplications 2016: Applications of Evolutionary Computation*, volume 9597 of *Lecture Notes in Computer Science*, pages 653–668. Springer International Publishing, Cham, 2016.

[53] A. Lastovetsky. Heterogeneity in parallel and distributed computing. *J. Parallel Distrib. Comput.*, 73(12):1523 – 1524, 2013.

[54] P. Vijaya Laxmi and M.L. Soujanya. Perishable inventory system with service interruptions, retrial demands and negative customers. *Appl Math Comput*, 262:102 – 110, 2015.

[55] Q. Li and P. Yu. Multimodularity and its applications in three stochastic dynamic inventory problems. *Manufacturing & Service Operations Management*, 16(3):455–463, July 2014.

[56] S. Li, J. Zhang, and W. Tang. Joint dynamic pricing and inventory control policy for a stochastic inventory system with perishable products. *Int J Prod Res*, 53(10):2937–2950, 2015.

[57] S. Minner and S. Transchel. Periodic review inventory-control for perishable products under service-level constraints. *OR Spectrum*, 32(4):979–996, September 2010.

[58] S. Miranda, J. Feldt, F. Pratas, R.A. Mata, N. Roma, and P. Tomás. Efficient parallelization of perturbative Monte Carlo QM/MM simulations in heterogeneous platforms. *Int J High Perform C*, pages 1–13, 2016.

[59] S. Nahmias. Perishable inventory theory: A review. *Oper Res*, 30(4):680–708, July 1982.

[60] T. Obdam, H. Braam, L. Rademakers, and P. Eecen. Estimating costs of operation & maintenance for offshore wind farms. In *Proceedings of European Offshore Wind Energy Conference*, pages 4–6, 2007.

[61] G. Ortega, J. Lobera, I. García, M.P. Arroyo, and E.M. Garzón. Parallel resolution of the 3D Helmholtz equation based on multi-graphics processing unit clusters. *Concurr Comput*, 27(13):3205–3219, 2015.

[62] K.G.J. Pauls-Worm, E.M.T. Hendrix, A. Gutierrez-Alcoba, and R. Haijema. Order quantities for perishable inventory control with non-stationary demand and a fill rate constraint. *Int J Prod Econ*, 181(Part A):238–246, 2016.

[63] K.G.J. Pauls-Worm, E.M.T. Hendrix, R. Haijema, A. Gutierrez-Alcoba, and J.G.A.J. van der Vorst. Order policies for a perishable product with non-stationary demand under a fill rate constraint. In *Book of Abstracts, 18th International Symposium on Inventories*, pages 130–130, 2014.

[64] K.G.J. Pauls-Worm, E.M.T. Hendrix, R. Haijema, and J.G. A.J. van der Vorst. Inventory control for a perishable product. *Int J Prod Econ*, 157:133–146, 2014.

[65] K.G.J. Pauls-Worm, E.M.T. Hendrix, R. Haijema, and J.G.A.J. van der Vorst. An MILP approximation for ordering perishable products with non-stationary demand and service level constraints. *Int J Prod Econ*, 157:133–146, November 2014.

[66] N.T. Raknes, K. Ødeskaug, M. Stålhane, and L. Magnus. Scheduling of maintenance tasks and routing of a joint vessel fleet for multiple offshore wind farms. *Journal of Marine Science and Engineering*, 5(1), 2017.

[67] Z. Ren, J. Wan, W. Shi, X. Xu, and M. Zhou. Workload Analysis, Implications, and Optimization on a Production Hadoop Cluster: A Case Study on Taobao. *IEEE T. Serv. Comput.*, 7(2):307–321, April 2014.

[68] R. Rossi. Periodic review for a perishable item under non stationary stochastic demand. In Bakhtadze Natalia, editor, *Manufacturing Modelling, Management, and Control, Volume 7, Part 1*, pages 2021–2026. International Federation of Automatic Control, June 2013.

[69] R. Rossi, O.A. Kilic, and S.A. Tarim. Piecewise linear approximations for the static-dynamic uncertainty strategy in stochastic lot-sizing. *Omega*, 50:126–140, January 2015.

[70] R. Rossi, S.A. Tarim, B. Hnich, and S. Prestwich. Computing the non-stationary replenishment cycle inventory policy under stochastic supplier lead-times. *Int J Prod Econ*, 127(1):180–189, September 2010.

[71] R. Rossi, S.A. Tarim, S. Prestwich, and B. Hnich. Piecewise linear lower and upper bounds for the standard normal first order loss function. *Appl Math Comput*, 231:489–502, 2014.

[72] H. Scarf. The optimality of (s,s) policies in the dynamic inventory control. In *Proceedings of the first stanford symposium on mathematical methods in social sciences.* Stanford University Press, 1960.

[73] B. Scholz-Reiter, J. Heger, M. Ltjen, and A. Schweizer. A MILP for installation scheduling of offshore wind farms. *International Journal of Mathematical Models and Methods in Applied Sciences*, 5(2):371–378, 2011.

[74] S.K. De Schrijver, E. Aghezzaf, and H. Vanmaele. Double precision rational approximation algorithm for the inverse standard normal first order loss function. *Appl Math Comput*, 219(3):1375 – 1382, 2012.

[75] E.A. Silver. Inventory control under a probabilistic time-varying, demand pattern. *A I I E Transactions*, 10(4):371–379, December 1978.

[76] E.A. Silver and H.C. Meal. A heuristic for selecting lot size quantities for the case of a deterministic time-varying demand rate and discrete opportunities for replenishment. *Production and Inventory Management*, 14(2):64–74, 1973.

[77] E.A. Silver, D.F. Pyke, and R. Peterson. *Inventory Management and Production Planning and Scheduling*. Wiley, 1998.

[78] M. Snir, S. Otto, S. Huss-Lederman, D. Walker, and J. Dongarra. *MPI-The Complete Reference, Volume 1: The MPI Core*. MIT Press, Cambridge, MA, USA, 1998.

[79] B. Snyder and M.J. Kaiser. Ecological and economic cost-benefit analysis of offshore wind energy. *Renewable Energy*, 34(6):1567 – 1578, 2009. URL: `http://www.sciencedirect.com/science/article/pii/S0960148108004217`, `doi:https://doi.org/10.1016/j.renene.2008.11.015`.

[80] M. Stålhane, E.E. Halvorsen-Weare, and L. Magne. Nowitech fleet-size and mix model. `http://www.sintef.no/globalassets/project/maritime-logistics-optimization-for-offshore-wind/staalhaneetal_wp_2016.pdf`, MARINTEK, MT2016 F-114, 2016.

[81] M. Stålhane, L. Magnus, and V. Skaar. Optimization of routing and scheduling of vessels to perform maintenance at offshore wind farms. *Energy Procedia*, 80(Supplement C):92 – 99, 2015. 12th Deep Sea Offshore Wind R&D Conference, EERA DeepWind'2015. URL: `http://www.sciencedirect.com/science/article/pii/S1876610215021438`, `doi:https://doi.org/10.1016/j.egypro.2015.11.411`.

[82] M. Stålhane, H. Vefsnmo, E.E. Halvorsen-Weare, L. Magnus, and L. Magne. Vessel fleet optimization for maintenance operations at offshore wind farms under uncertainty. *Energy Procedia*, 94:357 – 366, 2016. 13th Deep Sea Offshore Wind R&D Conference, EERA DeepWind'2016.

[83] S. Tabik, A. Villegas, E.L. Zapata, and L.F. Romero. Optimal tilt and orientation maps: a multi-algorithm approach for heterogeneous multicore-GPU systems. *J Supercomput*, 66(1):135–147, 2013.

[84] S.A. Tarim and B.G. Kingsman. Modelling and computing (Rn,Sn) policies for inventory systems with non-stationary stochastic demand. *Eur J Oper Res*, 174(1):581–599, October 2006.

[85] H. Tunc, O.A. Kilic, S.A. Tarim, and B. Eksioglu. The cost of using stationary inventory policies when demand is non-stationary. *Omega*, 39(4):410–415, August 2011.

[86] H.M. Wagner and T.M. Whitin. Dynamic version of the economic lot size model. *Manage Sci*, 5(1):89–96, 1958.

[87] G.R. Waissi and D.F. Rossin. A sigmoid approximation of the standard normal integral. *Appl Math Comput*, 77(1):91 – 95, 1996.

[88] H.M. Wee. Economic production lot size model for deteriorating items with partial back-ordering. *Computers & Industrial Engineering*, 24(3):449 – 458, 1993.

[89] Y. Yang, Y.F. Chen, and Y. Zhou. Coordinating Inventory Control and Pricing Strategies Under Batch Ordering. *Oper Res*, 62(1):25–37, February 2014.