

MULTI-PERIOD STOCHASTIC OPTIMIZATION PROBLEMS  
IN  
TRANSPORTATION MANAGEMENT

Thèse présentée en vue de l'obtention du grade  
de Docteur en Sciences Économiques et de Gestion

**Thierry PIRONET**

Promoteur:

Yves CRAMA, Université de Liège

Membres du comité:

Yasemin ARDA, Université de Liège

Frédéric SEMET, Ecole Centrale de Lille

Membres du jury:

Gerrit JANSSENS, Université de Hasselt

Sabine LIMBOURG, Université de Liège

Michaël SCHYNS, Université de Liège

# Contents

<b>I</b>	<b>Introduction</b>	<b>13</b>
<b>1</b>	<b>Thesis subject</b>	<b>15</b>
<b>2</b>	<b>Optimization in transportation</b>	<b>19</b>
2.1	Introduction . . . . .	19
2.2	Decision making in transportation management . . . . .	20
2.2.1	Transportation topic and mode . . . . .	20
2.2.2	Frequency of the decision . . . . .	21
2.2.3	Classical assumptions in transportation models . . . . .	22
2.3	Dynamism and time in transportation . . . . .	26
2.3.1	Time in transportation . . . . .	28
2.3.2	Dynamic decision process in transportation . . . . .	30
2.4	Stochasticity in transportation . . . . .	33
2.4.1	Stochastic information in transportation model . . . . .	33
2.4.2	Types of probability distributions . . . . .	35
2.5	Problems in transportation management tackled in the thesis . . . . .	36
<b>II</b>	<b>Methodology</b>	<b>37</b>
<b>3</b>	<b>Multi-period optimization</b>	<b>39</b>
3.1	Introduction . . . . .	39
3.2	Multi-period rolling horizon process . . . . .	39
3.3	Specific parts of the rolling horizon . . . . .	42
3.4	Dynamism of the rolling horizon process . . . . .	43
3.5	Policy versus solution . . . . .	44
3.6	Start and end of horizon biases . . . . .	46
3.7	Rolling horizon length . . . . .	48
3.8	Bounds with fully revealed information . . . . .	49
3.9	Value of information in multi-period optimization . . . . .	52

<b>4</b>	<b>Stochastic optimization problems</b>	<b>57</b>
4.1	Introduction . . . . .	57
4.1.1	Stochastic programming . . . . .	57
4.1.2	Expected cost minimization approaches . . . . .	61
4.1.3	Discarded optimization techniques . . . . .	64
4.1.4	Scenario-based models . . . . .	65
4.2	Scenario-based techniques . . . . .	67
4.2.1	Mono-scenario approximation . . . . .	67
4.2.1.1	Expected value scenario . . . . .	67
4.2.1.2	Modal value scenario . . . . .	68
4.2.1.3	Optimistic value scenario . . . . .	69
4.2.1.4	Customized scenario . . . . .	69
4.2.1.5	Bounds and mono-scenario approximation . . . . .	69
4.2.2	Multiple Scenario Approaches (MSA) . . . . .	70
4.2.2.1	Scenario generation . . . . .	71
4.2.2.2	Consensus algorithm . . . . .	72
4.2.2.3	Restricted Expectation algorithm . . . . .	73
4.2.2.4	Regret algorithm . . . . .	75
4.2.2.5	Scenario hedging and branch-and-regret algorithms . . . . .	76
4.2.2.6	Deterministic equivalent: scenario tree . . . . .	77
4.3	Bounds and expected value solution . . . . .	79
<b>5</b>	<b>Statistical validation</b>	<b>81</b>
5.1	Introduction . . . . .	81
5.1.1	Value of information . . . . .	83
5.1.2	Means and ratios . . . . .	84
5.1.3	Competitive analysis and approximation ratio . . . . .	84
5.1.4	Percentage deviation from the bounds . . . . .	85
5.1.5	Mean and variance trade-offs . . . . .	86
5.1.6	Statistical pairwise comparison test . . . . .	86
5.1.7	Conclusion: Test integration . . . . .	87
5.2	Statistical validation: a two-step method . . . . .	87
5.2.1	Normality check . . . . .	88
5.2.2	Statistically significant difference . . . . .	89
5.3	Robustness analysis . . . . .	90
<b>III</b>	<b>Applications</b>	<b>93</b>
<b>6</b>	<b>Multi-period vehicle loading with stochastic release dates</b>	<b>95</b>

6.1	Introduction . . . . .	96
6.1.1	Coordination of production and transportation . . . . .	96
6.1.2	Industrial setting . . . . .	97
6.1.3	Outline . . . . .	98
6.2	Formulation . . . . .	98
6.3	Literature review . . . . .	100
6.4	Deterministic optimization . . . . .	101
6.5	Stochastic optimization . . . . .	103
6.5.1	Deterministic equivalent: scenario tree . . . . .	104
6.5.2	Local Optimization algorithm ( <i>LO</i> ) . . . . .	104
6.5.3	Mean Release Date algorithm ( <i>Mean</i> ) . . . . .	105
6.5.4	Modal Release Date algorithm ( <i>Mod</i> ) . . . . .	105
6.5.5	Consensus algorithm ( <i>Cons</i> ) . . . . .	105
6.5.6	Restricted Expectation algorithm ( <i>RE</i> ) . . . . .	106
6.6	Numerical instances . . . . .	107
6.6.1	Deterministic parameters . . . . .	107
6.6.2	Stochastic release dates . . . . .	108
6.7	Algorithmic performance evaluation . . . . .	109
6.7.1	Evaluation scenarios . . . . .	109
6.7.2	Statistical analysis of results . . . . .	110
6.7.3	Fully or partially revealed information . . . . .	110
6.7.4	Value of information . . . . .	111
6.7.5	Computational limitations and bias . . . . .	112
6.7.5.1	Start and end of horizon . . . . .	112
6.7.5.2	Objective function . . . . .	112
6.7.6	Computing times . . . . .	113
6.8	Computational results . . . . .	113
6.8.1	First results . . . . .	113
6.8.2	Robustness analysis . . . . .	117
6.9	An optimistic algorithm . . . . .	118
6.9.1	Comparative performance of algorithm <i>Optimist</i> . . . . .	118
6.9.2	Additional results . . . . .	120
6.9.3	Interpretation . . . . .	120
6.10	Conclusions . . . . .	122
<b>7</b>	<b>Multi-period vehicle assignment with stochastic load availability</b>	<b>125</b>
7.1	Introduction . . . . .	126
7.1.1	Industrial problem . . . . .	126
7.1.2	Dynamics and stochasticity . . . . .	127
7.2	Literature review and theoretical framework . . . . .	128

7.3	Problem definition . . . . .	130
7.3.1	Rolling horizon procedure . . . . .	133
7.3.2	Time-space example . . . . .	134
7.4	Formulation . . . . .	137
7.4.1	Scenario-based approximate deterministic models . . . . .	137
7.4.2	Formulation for a mono-scenario model . . . . .	137
7.4.3	Complexity for a single scenario formulation . . . . .	142
7.4.4	Formulation for a deterministic subtree model . . . . .	142
7.4.5	Complexity for a subtree of scenarios . . . . .	146
7.5	Algorithms . . . . .	146
7.5.1	Statistical validation of policy performance . . . . .	147
7.5.2	Value of the stochastic information . . . . .	148
7.5.3	Single scenario algorithms . . . . .	149
7.5.3.1	Optimistic scenario algorithm: Opt . . . . .	149
7.5.3.2	Modal value scenario algorithm: Mod . . . . .	150
7.5.3.3	Expected value scenario algorithm: EG . . . . .	150
7.5.4	Multiple-scenario approximations . . . . .	151
7.5.4.1	Consensus algorithm: Cs . . . . .	151
7.5.4.2	Restricted Expectation algorithm: RE* . . . . .	153
7.5.5	Subtree algorithm: TR . . . . .	154
7.6	Instances . . . . .	155
7.6.1	Temporal data . . . . .	155
7.6.2	Spatial data . . . . .	156
7.6.3	Economic data . . . . .	157
7.6.4	Trucks . . . . .	157
7.6.5	Loads . . . . .	158
7.6.6	Stochastic parameters . . . . .	158
7.7	Statistical validation and algorithmic parameters . . . . .	160
7.7.1	Statistical validation parameters . . . . .	160
7.7.2	Algorithmic parameters . . . . .	161
7.8	Numerical results . . . . .	161
7.8.1	Results for availability linked to duration (150 loads) . . . . .	162
7.8.2	Results for availability linked to duration (200 loads) . . . . .	166
7.8.3	Results for the availability linked to departure city range . . . . .	168
7.8.4	Preliminary conclusions . . . . .	170
7.8.5	Subtree algorithm robustness analysis . . . . .	171
7.8.6	Subtree algorithm sensitivity to horizon length . . . . .	175
7.9	Conclusions . . . . .	175
7.10	Acknowledgments . . . . .	176

<b>IV</b>	<b>Conclusions and perspectives</b>	<b>177</b>
<b>8</b>	<b>Conclusions and perspectives</b>	<b>179</b>
8.1	Conclusions . . . . .	179
8.2	Perspectives . . . . .	184
<b>V</b>	<b>Bibliography</b>	<b>185</b>



## Acknowledgments

This thesis is a personal and collaborative project that has been performed with the support of several persons over many years. Following an historical sequence that is not related proportionally to the magnitude of my acknowledgments, I would like to thank first my parents which gave me the opportunity and the will to learn and to improve myself. I hope that this thesis is also a reward for them.

I also want to warmly give thanks to the different boards of directors as well as the present and former Research and Education Unit Operations presidents that supported me in this process by giving me working time and research funds: Marc Dubru & Yves Crama, Thomas Froehlicher & Louis Esch & François Pichault & Wilfried Niessen & Axel Gauthier, Gérard Colson & Daniel Dubois and Michaël Schyns.

Even if he spent time and energy to teach me the importance of precise, clear and concise formulations to get optimal sentences and reasonings, I must admit once again that it is impossible for me to formulate my acknowledgments, according to these advices, to my supervisor, Professor Yves Crama. From an operational research point of view, I would say that my gratitude is equal to the optimal solution of a maximization problem without upper bound, meaning that no words can transmit my feeling. Indeed, he was a guide for every step of this doctoral education process from the initial courses, through the research topic definition, problem formulation and modeling and the results analysis up to the oral and written communications skills developments. For his constant availability for discussions, his kind and enlightened advices, his wiseness on the aims of the doctorate, for sharing his international network of colleagues and mostly for being an example himself, I would like here to express my deep and hearty respect and gratefulness to Yves Crama for his invaluable support during these years.

I would also like to bring to light the important roles of Professors Yasemin Arda and Frédéric Semet, both committee members. They were of great support along the course of this thesis from the early beginning up to the latest stages. Personally, it was a pleasure every year to receive a positive reinforcement for my research during committee meetings. One-to-one discussions at conferences or in the building proved also a permanent interest for my research. Most of all, their comments, remarks and advices after the reading of the first version of the thesis report were very helpful to improve its quality for the formal and the scientific facets. They pointed out missing or unclear sections which turned out to be important issues. Moreover, advices about related works published in scientific articles and remarks for problem modeling improved significantly the contributions of the thesis. Finally, these deep involvements lead to a teamwork which I particularly appreciate when I was finish-



ing this report. So, it was a great chance for me to be surrounded by such persons which I warmly thank. May this work be a piece of reward for them.

All jury members, Professors Gerrit Janssens, Sabine Limbourg and Michaël Schyns are warmly thanked for the advised comments they have provided after reading the present document. I hope that they enjoyed the topic and will appreciate the contributions for their future research.

I also thank Professor Pascal Van Hentenryck for welcoming me during a research period inside the computing department of the Brown University in Rhode Island. He was very helpful for understanding the advantages and the drawbacks from theory to practice when implementing the algorithms exposed in his reference book about On-line Stochastic Optimization.

David Kronus, as a post-doctoral student, and co-researcher was also of great assistance during the early stages of the research. For his constant questioning and creativity as well as the initial results he provided, I would like to warmly thank him for the nice time we spent discussing together.

It is also my pleasure to bring to light the contribution of Isabelle Peere as a linguistic reviewer for chapter 7. May she be rewarded for this cumbersome piece of work.

Different organisms which assisted me to travel to international conferences or remote meetings can find here an official acknowledgment for their important and often shadowed assistance: FNRS (Fonds National de la Recherche Scientifique), the Belgian Scientific Policy (BELSPO) Grant P7/36 in the framework of the Interuniversity Attraction Poles program, the ARD (Administration de la recherche et du développement) of the university of Liège (among others Isabelle Halleux and Evelyne Favart), PRISME of HEC-ULg, the doctoral college (Michaël Schyns, Cédric Heuchenne, Caroline Michotte), the UER Operations of HEC-ULg, UGR (University of the Great Region) project GreatROAD, the Walloon region project "TranslogisTIC". Special thanks to the staff of the accountancy department of HEC-ULg that administratively followed and kindly supported my external activities: Monique Reul, Virginie Siquet, Françoise Putzeys and Lydie Ezanno. I acknowledge the computing department for the technical support.

Last but not least, I would like to express how glad I am to be part of the research group QuantOM from which colleagues create a daily positive atmosphere leading to self-fulfillment : Stéphanie Aerts, Guillaume Amand, Maud Bay, Véronique François, Julien Hambuckers, Sara Handerek, Anne-Sophie Hoffait, Hande Kucukaydin, Géraldine Laurent, Virginie Lurkin, Stephano Michelini, Martine Mostert, Célia Paquay, Elisabeth Rodriguez-Heck, Mahmood Rezaei, Christine Tawfik.

Finally, thanks to anyone, colleagues from HEC-ULg or relatives, that helped to the fulfillment of this work.

Dedicated to my wife Bénédicte for her optimal understanding and logistic support in a multi-period and stochastic thesis environment; to my children Valentine and Emile for our mutual patience; to my parents Yvette and Jean-Marie and my in-laws as babies-sitters; to my brother Fabian; to my sister Isabelle and my nieces Marie and Lucie and finally to my grand-mother and guardian angel Elisabeth.



## Thesis Abstract

The topic of this thesis in management science is to propose a methodology to tackle multi-period decision problems including stochastic forecasts and to apply it to the field of transportation management. This methodology relies mostly on a sequence of numerical experimentations based on a set of algorithms to assess the value of the multi-period setting and the interest to use the stochastic information contained in the forecasts. Moreover, a statistical validation method to compare the performances of algorithms enables us to rank them meaningfully. From theory to practice, the thesis is structured into four parts.

*Firstly*, we define the thesis subject and content. Then, based on a literature review, we present the past and present issues within the research field: "Optimization in Transportation". Mainly, we define the differences between multi-period stochastic models and classical deterministic mono-period ones. We explain how dynamism and stochasticity are taken into account within transportation problems.

*Secondly*, our methodology, which is our main contribution, is exposed from a generic point of view in the theoretical research field: "Multi-period Stochastic Optimization Problems". On the one hand, temporal parameters and computational issues in multi-period optimization are detailed; on the other hand a summary of optimization techniques and algorithms for stochastic optimization problems is provided. Afterwards, the statistical validation of algorithmic performance is discussed.

Then, *part three* contains two applications that lead us to set up the methodology, following an inductive method. The first problem, based on an industrial application at the start of the research, investigates a "multi-period vehicle loading problem with stochastic release dates". The second application, based on a more generic approach for the deployment of the methodology, deals with a "multi-period vehicle assignment problem with stochastic load availability".

Finally, in the *fourth part*, we conclude on the thesis contributions and propose some perspectives.



**Part I**

**Introduction**



# Chapter 1

## Thesis subject

The aim of this thesis is to present a methodology in the field of decision making and to apply it to long-haul road transportation problems. These problems have two major specificities: they are multi-period decision making processes and they contain stochastic information. For example, in order to reduce cost in transportation, it is often profitable to reduce the total number of trucks by grouping efficiently items to be carried. This could be done by postponing over time the delivery even when facing penalties for keeping the item in inventory or even for a late delivery at the customer plant. Indeed, some items currently available could be collectively transported with items expected to be ready for shipment in the forthcoming days. So, at each decision step the dispatcher faces a loading problem which consists in deciding which and how items should be dispatched. This leads to a repetitive decision problem. When the information available for the future decision steps is stochastic, this leads to investigate multi-period stochastic problems.

In the present thesis, the range of repetitive decision problems is restricted to multi-period ones. Typically, periods are days. This means that the decision frequency is regular and is not induced by external elements which might occur, for instance, according to the realization of a stochastic event. Moreover, projections, or forecasts, are provided over a constant number of periods leading to a so-called "rolling horizon" process. The cumulative value of the successive decisions at each decision period is called the policy value. Furthermore, for each problem, the probability distributions are related to periods. In the first application, the distribution refers to the probability for a transportation order to become available during a specific period. In the second one, the distribution refers to the probability for an item to be released from production within the periods of a time-window. The distributions that we consider are discrete and defined over a small finite set.



Therefore, the thesis subject in terms of problems consists of:

- multi-period stochastic optimization problems,
- faced in road long-haul transportation,
- and containing stochastic information,
- represented by discrete distributions,
- over a rolling horizon.

As the size of the problems become larger, and due to the large number of potential combinations of the stochastic parameter realizations, such problems become hard to solve to optimality. Memory and computation time issues arise at each decision period. Consequently, in the present work, approximate models are solved. The range of approximate models is part of the methodology proposed in this thesis. Mainly, the optimization techniques are scenario-based. A scenario is a deterministic representation over time of a specific outcome of the stochastic parameter. Usually, scenario-based deterministic models approximating the stochastic one are easier to solve. For the applications tackled in this thesis, optimal solutions for a single scenario-based model can be found in reasonable time. Single scenario and multiple scenario approaches are exposed and briefly described hereunder.

Policies issued from the optimization of the deterministic models associated with certain single scenarios are proposed in order to approximate the best policy for the stochastic problem: among them, the **modal scenario**, the **expected value scenario**, the **optimist scenario**. Moreover, policy values issued from the optimization of models generated by specific scenarios are used as bounds for the value of the best policy. Typically, these scenarios contain a restricted or a fully revealed part of the deterministic information. More precisely, our bounds are derived from: the myopic scenario containing no projections, from the scenario containing a-posteriori fully revealed information over the rolling horizon and from the scenario containing a-posteriori fully revealed information over the whole horizon. In addition, the policy values provided by the optimization over these approximate models allow us to compute several measures of the value of information.

To improve the performance of policies approximating the optimal one for the stochastic problem, three multiple scenario approaches are also presented, namely:

1. the **Consensus algorithm**, that generates a compromise solution based on the most frequent decisions issued from solutions obtained when models generated by a set of scenarios are separately optimized;

2. the **Restricted Expectation algorithm** that selects a solution by estimating its value over a sample of scenarios;
3. the **Subtree algorithm** that solves a reduced model of the deterministic equivalent representation of the problem. This subtree model is based on a reduced set of scenarios linked by non-anticipativity constraints. This leads to simultaneously consider several scenarios.

A simulation framework using scenarios is deployed to check the relative performances of these algorithms, and a statistical validation is mandatory to assess the significance of the results. Biases linked to the simulation framework are also discussed.

Thus, the thesis subject and contribution in terms of optimization methodology relies on proposing scenario-based algorithms and a framework for policy performance comparisons.

In order to demonstrate the practicability of this contribution, the complete methodology is deployed over two applications.

The first application is a "**multi-period vehicle loading problem with stochastic release dates**". The decision problem consists in loading trucks with items on a daily basis. Deliveries must take place in a time window including an early delivery period, regular delivery periods and a late delivery period per item. The combinatorial choice lies in the choice of the items that have to be transported in the same truck according to their relative weights and to the capacity of the truck while taking into account the penalty faced for the selected delivery period. The aim is to minimize the total cost, over a long horizon, arising from the number of trucks used and the cumulated penalties for the items. Data on projected items to be released from production are provided over a rolling horizon ahead of the current decision period. The probability distribution of release dates for an item is provided initially as a time-window spread over 4 periods. As the horizon rolls over, either the item is released, or the distribution reduces to a 3- or 2-period time-window. One characteristic of this problem is that it leads to the consideration of conditional distributions.

The second application is a "**multi-period vehicle assignment problem with stochastic load availability**". It models a full-truckload (FTL) situation. The combinatorial decision resides in the selection of shipments to be performed by a fleet of trucks. Moreover, unladen trips can be performed or the trucks can wait in their present locations. The decision process is performed daily. The aim is to maximize profit stemming from the reward obtained for transporting shipments and the cost due to empty moves or staying at standstill. Data are provided over a rolling horizon

and include forecasts on transportation orders which might occur or not in a particular period. The probability distribution of availability for a transportation order is represented by a Bernoulli distribution.

Both applications lead to multi-period stochastic models, with a discrete distribution defined over a small finite set for the data inside the rolling horizon. So, they comply with the preliminary statements and the optimization techniques and policy performance comparisons are applied according to the proposed methodology.

This methodology and related algorithmic techniques might seem to be particularly dedicated to a restricted set of problems. But, it is also possible to adapt this framework to other environments. Discussing the modifications to be performed to continuous time optimization, continuous distributions or distributions defined over an infinite set of values for the stochastic parameter or other transportation modes as well as a scheduling environment is out of the scope of this thesis.

This thesis is structured into four parts.

**Part I Introduction** contains the present chapter dealing with the description of the thesis subject. Then, chapter 2 positions the transportation problems faced in this thesis within the scientific literature.

The **Methodology** is described in Part II.

As they represent specific features of the investigated models, two chapters are dedicated, respectively, to the multi-period decision making environment and to the stochastic one. Chapter 3 presents the multi-period optimization setting and the rolling horizon process. The different parts of the horizon are defined and the dynamism of the rolling procedure is detailed. Deterministic bounds are introduced and several values of information are valuated.

Stochastic optimization techniques are described in chapter 4, with an emphasis on heuristics based on sampling and deterministic scenario optimization.

A statistical validation procedure for algorithmic performance comparison is proposed in chapter 5.

Then, Part III proposes two applications illustrating the methodology as mentioned above. Chapter 6 contains an article published in the EURO Journal on Transportation and Logistics. It investigates the multi-period vehicle loading problem with stochastic release dates.

Chapter 7 deals with the multi-period vehicle assignment problem with stochastic load availability.

Finally, in Part IV, chapter 8 concludes with a discussion of the thesis contributions and highlights some perspectives for future research.

## Chapter 2

# Optimization in transportation

### 2.1 Introduction

As this thesis is entitled "multi-period stochastic optimization problems in transportation management", a major contribution of this work is to study the intersection of three domains : multi-period decision making, stochastic optimization problems and transportation models. Many works have been performed in each of these areas and some at the intersections of two of them. Nevertheless, regarding the intersection of the three domains, one must admit that the literature reflecting problems, models, algorithms and methodological issues is quite sparse. Indeed, these multi-period stochastic problems in transportation accumulate all difficulties from their relative sub-domains. So, developments should be based on preliminary research performed in each restricted area and be integrated in a new methodological framework. Yet, the time has come to propose some progress in this central and composite domain. This necessity is related to the industry that requires to handle real-life problems with all their features and not only a reduced or approximate version of a practical problem. Most of all in transportation, wherein communication systems have improved the availability of information. Moreover, the commercial competition increases the need for efficient business processes, among others in decision making. Therefore, optimization of decisions requires to use any piece of information and algorithmic procedures that might lead to a commercial advantage such as cost reduction or profit increase.

As this thesis is in the field of management science, the decision process in transportation management might be exposed from the strategic bird's-eye view down to the operations issued from the ground level. However, in the present work, the focus lies on the operational level. In fact, the concern is on a generic methodology for decision making in transportation illustrated by two applications performed daily in

the dispatching department of a transportation company (see chapters 6 and 7).

As this thesis is also in the operational research area, the literature review inserted in chapter 2 introduces the two major specificities of the handled transportation problems: dynamic decision process (section 2.3) and stochasticity (section 2.4) in transportation. This leads to specify the thesis subject and to highlight its characteristics (section 2.5).

## **2.2 Decision making in transportation management**

Transportation management represents a wide range of decisions problems. So, in this section, the aim is to characterize the differences of the transportation problems tackled in the thesis from other issues inside this domain. Successively the restrictions over the transportation topic and mode, the frequency of the decision and some classical assumptions in the modeling approach are depicted in order to differentiate the present work from previous ones.

### **2.2.1 Transportation topic and mode**

First of all, some domains in transportation must be discarded such as: infrastructure design, people or public transportation, multiple modes (maritime, air, rail), national or international regulations and social, technical, environmental issues. This bundle of subjects that might be related to transportation management is for example (among other classifications) brought to light by the division of the domains in the series of journals from Elsevier: Transportation Research.

- *Part A: Policy and Practice*
- *Part B: Methodological*
- *Part C: Emerging technologies*
- *Part D: Transport and Environment*
- *Part E: Logistics and Transportation Review*

In this thesis, even if the present work can be transposed to other transportation modes, the methodology proposed and the related applications are based on long-haul road transportation, meaning freight transported by trucks over a long period of time. The choice of road transportation is due to the fact that most applications and models are related to it within transportation. Besides this, long-haul concerns long distance journeys, so long durations trips leading to take a few decisions along time for each vehicle. Moreover, the cost or reward of that kind of transportation

is expensive, so that it might be profitable to postpone or to select transportation orders by taking into account forthcoming orders. So, decisions can be simultaneously performed on a regular basis for the whole fleet of vehicles. This leads to a multi-period decision framework. Typically in long-haul transportation, decision epochs equal a single day. Actually, long-haul road transportation is situated in between maritime transportation, that uses periods of several days or weeks, and at the opposite daily distribution or emergency services that manage their operations within minutes or hours. Owing to the flow of in-coming and out-going information in each period and to the rhythm of decisions, the reader should keep in mind that the operational multi-period problems investigated in the thesis are situated between on-line optimization (e.g. emergency service) and mid-term tactical decision making (e.g. maritime transportation). As air and rail transportation usually uses a highly regulated network with time-slots reservation and long-term schedules, these modes do not fit well with a day to day decision process except for some chartering applications. So, the words "vehicle" and "shipments" might be used by the reader to transfer its potential applications from the present setting to other modes even if long-haul road transportation is the main concern of this thesis.

### **2.2.2 Frequency of the decision**

Returning to the decision frequency, within transportation some decisions are performed typically once or a few times in a relatively long horizon as they are of a strategic nature. For example:

- facility location problem: to locate depots close to customers in order to minimize distribution costs or to locate depots far away from customers within a risk pooling strategy,
- road network design: to define the connections to be built between cities,
- multi-modal network design: to locate platforms where freight is changed from one mode to another,
- city logistics problems mixing a bit of the previous issues (warehouses, meeting points or cross-docking platform locations).

Yet, our concern when dealing with dynamic applications in transportation is mainly at an operational level. This means a repetitive decision process performed on a short term basis. Therefore, even if they are important issues in transportation management, these problems are out of the scope of this thesis which investigates an operational repetitive decision making process.

As a consequence, in the present research, repetitive decisions in road transportation consist for instance in defining, on a daily basis, which trucks will take which loads, when, from where to where, at which cost, for which reward, etc.

### 2.2.3 Classical assumptions in transportation models

This thesis investigates multi-period stochastic optimization problems. This kind of problem deals with a much more complex reality than the standard well-known problems which are usually mono-period, deterministic and short distances ones. These assumptions and others are often hidden by default in the models. So, let us review some classical assumptions of transportation models, starting with well-known models up to more complex problems in transportation. It is known that the standard Vehicle Routing Problem (VRP) is already hard to solve (NP-Hard). As a consequence of this algorithmic difficulty to solve a basic problem, the earliest developments in optimization for problems in transportation focused on simplified models and formulations caricaturing sometimes the reality (for the earliest developments, see the literature review from [Laporte & Osman, 1995] and [Toth & Vigo, 2001]).

The caricature comes out from several assumptions in the models. The first assumption resides in the aim of the decision. In transportation, the objective function is typically the total distance minimization as a way to minimize cost but when some customers can be retrieved by selection or by facing a penalty, a profit maximization objective function can be used. Consequently, problems such as the "*Travelling Salesman Problem*" (TSP) with profit have also been tackled and therefore the minimal distance might not be the only objective (for an introduction to profit optimization in transportation see [Feillet et al., 2005]).

A second assumption resides in the constraints that should be taken into account or not in the model. Indeed, several additional features have created models that fit better with real problems: heterogeneous fleet of vehicles, limited capacity of the vehicle or service level requirement. However, some constraints are sometimes discarded without reporting the reasons for that selection. Successively, this leads to introduce the following models.

The "*Vehicle Routing Problem*" (VRP) deals with a fleet of vehicles that should visit a set of customers while minimizing the total cost represented by the traveled distance with or without the fix vehicle cost; and with or without a maximum tour length per vehicle (for an introduction to the VRP, its variants and solutions techniques, see [Toth & Vigo, 2001]).

To fit with tonnage regulations, the "*Truck Dispatching Problem*" (TDP) was introduced by [Dantzig & Ramser, 1959]. This "TDP" is now known as the "*Ca-*

*pacitated Vehicle Routing Problem" (CVRP)*. It includes a limited capacity for the vehicles (often weight) which exclusively either deliver or pick up goods from or to a single depot.

The CVRP can also consider options such as heterogeneous fleet, mono-product or multi-products transportation, multi-tanks vehicles, 2D or 3D loading constraints, drivers regulations... These options create a wide family of problems more or less concerned by the loading or the routing constraints, making them difficult to be solved by a common algorithm.

As already mentioned in chapter 1, as an example for the diversity of models, two applications are proposed in chapters 6 and 7. Both problems have different characteristics.

1. The first application considers a "vehicle loading problem", which focuses only on the loading problem without routing, making it close to a bin-packing problem.
2. The second application considers a "vehicle assignment problem" (see [Powell, 1988] and [Powell, 1996]), simply selecting the paths to be followed without loading constraints.

This highlights the fact that problems in transportation which sound close might represent drastically different decision environments. So, even though operational routing and loading problems seem to be restricted to a subclass of problems in transportation, the wide range of models, formulations, objective functions, variables and constraints generates an explosion of issues. Adding dynamic and stochastic elements should lead also to a potential tremendous number of various applications. Instead of complicating the model by adding options from the TSP through VRP up to CVRP, it might be interesting to define the most complex problem in transportation and to retrieve features that are not activated in reality. The objective is not to create a complex problem for itself, but to represent the complex situations faced by practitioners with all the required features.

There was a first attempt to define such a global generic model with the "*Pickup and Delivery Problem" (PDP)* that represents the case where goods can be picked-up and delivered along the way between two locations while not exceeding the capacity (see [Berbeglia et al., 2007]). But, even this generic model can be complicated according to situations faced in reality such as temporal constraints, service level requirements from customers and others described below.

In fact, despite their importance, cost or profit are not the only criteria in the relationship between the freight forwarder and the carrier. In practice, the Service Level Agreement (SLA) implies the respect of delivery time. Consequently, models



including time-windows (TW) for the un/loading operations have been introduced. The TW can be "*hard*" meaning that the scheduling of the operations should respect the time limits, or "*soft*" meaning that a penalty is faced in case a violation occurs. These temporal requirements are usually formalized as constraints inside the model.

In the same sense, a variant of the PDP for people transportation called "*Dial-A-Ride Problem (DARP)*" has been defined. Some penalties in the objective function are used to reduce passengers' inconvenience (for an introduction to the DARP, see [Cordeau & Laporte, 2003] and for models and algorithms, see [Cordeau, 2007]).

As one can notice, previous problems mainly focus on finding paths between points or locations. Another kind of problems which tends to find tours on arcs or edges such as the postman problem or garbage collection problem are also investigated in the field.

As another attempt to generalize transportation problems, in [Savelsbergh & Sol, 1995], the authors present a formulation of the "*General Pickup and Delivery Problem (GPDP)*" that can be used to formulate most problems as a sub-case, from the most complex GPDP to the simple TSP (for an introduction to and perspectives on the GPDP, see [Savelsbergh & Sol, 1995] and [Psaraftis, 1995]).

Even if the aim of this article was to present a global formulation for this range of problems, only a few lines mention the dynamic case, which implies a sequence of decisions, and not more for the stochastic version of the problem. In fact, within the model description, as another assumption, many authors assume by default but not explicitly that each decision has no influence on future decisions and is disconnected from previous ones. In most articles and models, this default setting is often hidden within the remark: "*all vehicles start and end at their depot*". As the starting and ending points are fixed, the problem comes back to an original state or at least the final state is defined. Similarly, the fact that only one decision is taken over time is implicitly induced. Therefore, every optimization stage can be disconnected. This represents the local area daily distribution situation or the national distribution in a small size country such as Belgium. However, in other cases such as in long-haul or international transportation, the starting and ending locations of vehicles might differ from a decision period to another. Therefore, the decision process depends on previous decisions and influences the following ones. In [Savelsbergh & Sol, 1995], the authors only mention, "*(...) the objective function for the static subproblem does not necessarily have to be equal to the objective function for the dynamic problem. The objective function for the static subproblem may reflect some knowledge or anticipation of future requests*". (for a recent state of the art review on PDP, see for the static version [Berbeglia et al., 2007] and for the dynamic one [Berbeglia et al., 2010])

Moreover, if the performance of algorithms can be compared over the solution values on benchmark instances for most static problems, a common validation framework for policies performances over dynamic problems is not widely admitted (for a review on dynamic VRP, see [Pillac et al., 2013]). Therefore, dealing with the dynamism of a problem is more than an option, it is a specific decision making area. Yet, these models with successive decision processes are often neglected by the research community. Consequently, even if the road transportation problems investigated inside this thesis, look similar at every time-period to standard models, they are highly different and beyond the limits of the usual models of road transportation. Firstly, because decisions are repeated but also because decisions are related to each other. This makes the problems out of the scope of most models and formulations as problems involve a dynamic process between decision periods.

Furthermore, as a third assumption, when the problem formulation contains forecasts, the information is usually provided as a probability distribution. If the initial plan is performed whatever happens, and only penalty or recourse actions can be taken in a second stage, the problem can be treated by stochastic programming as a static problem (see [Pillac et al., 2013]). This means that a stochastic and dynamic problem can be solved as a static one. On the other side, on-line optimization which regenerates or updates a new solution every time new information comes in for several periods, requires specific evaluation procedure to define what is a policy performance in the long run. Therefore, the term "dynamic" in a model might represent different decision making processes from a single decision to a huge finite sequence of decisions.

Finally, as the size of the model becomes larger (number of vehicles, customers...) the CPU time and the required memory increases. Sometimes, even if solutions techniques can guarantee optimal solutions, time-efficient heuristics have to be used. For problems involving stochastic information, the numerical explosion, called curse of dimensionality, often leads to intractable models in practice. So, finding the optimal solution remains unaffordable.

For all these reasons, less work has been performed in transportation about either dynamic or either stochastic models than about static and deterministic models. Moreover, when both aspects are embedded, the scientific work is rare. Yet, in reality, carriers dispatching centers often face a sequential decision process taking into account commercial forecasts. Whatever the quality of static models can be, they might be rejected by practitioners if they do not take into account the benefit that can be extracted from all the available information and a repetitive decision process. After recognizing the fact that there can be several objective functions in transportation, as a response to the standard paradigm, this thesis tries to propose a methodology to get rid of these two strong hypothesis or assumptions in trans-

portation models: "No forecasts" (stochastic), "No effect on the next decision step" (multi-period).

As a conclusion, transportation optimization problems usually neglect to mention the dynamism of the decision process and the stochastic behavior of real-world applications. Of course, it might be the case in some operational settings, yet for long-haul transportation, it is not the case. For this reason, these aspects are developed in dedicated sections of the introduction. An introduction to dynamism and time in transportation applications is proposed in section 2.3, because both subjects are often mixed and standard definitions are not fixed yet. Consequently, an important part of the section is devoted to expose the various utilizations of the term "dynamism" in transportation models. For that reason, after introducing concepts, details are provided about "time in transportation" in section 2.3.1 and about "dynamic decision process in transportation" in section 2.3.2.

For the second topic, a summary depicts how stochasticity is embedded in transportation models in section 2.4.

Finally, summarizing the introduction, the thesis subject and outlines are exposed in section 2.5.

## **2.3 Dynamism and time in transportation**

The title of the thesis mentions multi-period problems and not dynamic ones. This choice is due to the fact that the term "Dynamic" in operational research might induce confusion as there are several definitions. Inside the thesis, dynamic refers to a repetitive decision process. Yet, the problems are specifically labeled multi-period to highlight a repetitive decision process at fixed decision epochs and because the information is revealed over a rolling horizon. For that reason, when dynamic is used in the present thesis, it refers to a sequential decision process and multi-period to a sub-domain of these dynamic problems.

Nevertheless, to avoid a misunderstanding for the reader, the confusion about "dynamic" must be detailed. First of all, a well-known solution technique and a modeling framework introduced by Bellman (see [Bellman, 1957]) are called "Dynamic Programming". This technique is based on the division of a basic problem into sub-problems easier to solve, so that the global solution of the problem can be found. It was preliminary used for deterministic problems such as the shortest path problem. Therefore, the word "Dynamic" is used to represent a sequential algorithm and not a model to be solved repetitively along time.

The second drawback of this word "Dynamic" comes from the fact that it does not represent the same notion for all authors. The misunderstanding is a bit high-

lighted in [Powell et al., 1995]: *"One question that comes up with surprising frequency is: What constitutes a dynamic model? To answer this, we must first distinguish between a problem, a model, and the application of a model. A problem is dynamic if one or more of its parameters is a function of time. This includes such problems as vehicle routing with time windows or time-varying travel times. Note that two types of dynamic problems are covered here. The first type, which we call problems with dynamic data, are characterized by information that are constantly changing. Dynamic data might include real-time customer demands, traffic conditions, or driver statuses. The second type is problems with time-dependent data which is known in advance. (...) such as VRP-TW (...), but where this data is a known function of time.(...) Similarly, a model is dynamic if it incorporates explicitly the interaction of activities over time. (...) It is useful, however to distinguish between deterministic, dynamic models, and stochastic models which explicitly capture the staging of decisions and the realizations of random variables. (...) Finally, we have a dynamic application if a model is solved repeatedly as new information is received."*

Consequently, if one uses this definition in a broad sense, any problem involving time becomes dynamic. This wide definition might seem to create a confusion with a narrower definition restricted to a repetitive decision process. According to [Pillac et al., 2013], it is better to enlighten the terminology while using "static" and or "dynamic" to reflect the decision process. A static decision process defines a solution and possible recourse actions selected a-priori. This denomination stands even if some data are either known in advance or revealed during the execution. At the opposite, a dynamic process defines a successive decision process taking place while some new information is gradually taken into account.

Therefore, it is more rigorous to use a specific terminology for problems which contains information related to time and also for the dynamism of the decision process as described hereafter in the subsections 2.3.1 and 2.3.2.

The third difficulty may be induced by the word "stages" as it is used in "two-stage" and "multi-stage" expressions. These stages represent typically, two or several decisions steps but conversely to operational problems in transportation which are performed daily, the number of stages is finite. For two stages, a decision is applied after the first stage and then all the unknown information is revealed. So, subsequent actions, called recourse actions, can be applied according to the preliminary ones and the current state of the system in stage 2. For multi-stage problems, a piece of information is revealed after each decision and an extra decision is performed until the last stage is revealed. The confusion might come when the stages do not represent periods but consist of related or interlinked decisions representing levels of decisions or subsequent decision steps. This is the case in *"Material*

*Requirement Planning (MRP)*" when parent items generate the requirements for components. Some models labeled multi-stage in transportation correspond to this definition (e.g. see [Tsilingiris & Psaraftis, 2006]). For these reasons, in the present work, the expression multi-period referring to a consecutive decision process within time is preferred to multi-stage which might be understood as a hierarchical requirements decision process or successive and dependent decision processes. In the latter case for instance, the problem is not dynamic.

### 2.3.1 Time in transportation

For some authors, "dynamic" stands for "information linked to time". There are numerous ways to insert "time" within a transportation problem. Here are some examples:

- **The maximum tour length  $L$  constraint:** which sets, by a constraint in the model, that the sum of selected arcs of a path is inferior to a predefined limit. At first sight and literally, it seems to be a spatial constraint but inside the problem description it usually reflects a limit to avoid extra-hours or to restrict the driving time, which are indeed temporal constraints.
- **Time Windows (TW):** which impose that an action or the start of an action (for example loading, unloading in a PDP, VRP, DARP-TW) must take place during a certain interval of time. In the specific cases of the Traveling Repairman or Delivery Man problem with TW (see [Tsitsiklis, 1992]), the objective function might also be influenced by the time-windows, which leads to a delay minimization or shortest time problem or minimum waiting time problem. These TW can be hard or soft. In the first case, the temporal constraints must be respected. In the second one, a penalty is faced according to the non-fulfillment of the temporal constraint. Within these problems, time can be represented by a continuous or discrete variable and this might change the optimization technique.
- **Time-dependent problems:** due to the increase of traffic jam during peak hours, it becomes valuable to generate solutions that take into account the variability of travel time between points according to the day, weekday or holiday. This influences the temporal parameters of the models. There are two major representations to deal with that aspect: the a-priori one based on statistics that generates a factor that modifies the travel time according to a pattern or function along the day; or the on-line one (see section 2.3.2) that gets the travel time from a mobility control center. These problems get more and more relevant and investigated. They can drastically improve the practical

realization and the efficiency of transportation plans (see for an introduction [Malandraki & Daskin, 1992] and for an application [Ichoua et al., 2003]).

- **Periodic and timetabling problems:** in that case, time is turned into a frequency measure. Public transportation following a regular schedule such as buses, trains, airlines have successfully proposed techniques to gather the customer's frequency level of service. For freight transportation and routing including stock management, some articles propose interesting strategies that require usually a single decision. Anyway, periodic problems are still a developing area of research (e.g. see [Angelelli & Speranza, 2002]).
- **Planning:** some transportation problems are linked with scheduling and generate plans for several weeks or months according to the demand, like maritime vessel scheduling or fleet management for harvest. These plans are often tactical decisions running over several periods. Nevertheless, the initial decision is performed only once and kept. This is not what is considered as a multi-period problem for the present work.
- **Periods or Epochs:** such as in a MRP, data is provided period per period. Therefore, time is discontinuous. To be a relevant dynamic decision process, each decision period requires interactions between past and/or following periods in order to influence the current decisions. If not, every period can be optimized separately and time is not important except in the period itself. The number of periods can be finite or infinite and the information fully revealed or not. If the number of periods is finite and the information fully revealed, the problem becomes a static planning problem. In the other cases, if the information is revealed successively period per period, the decision process might be dynamic such as in a 2-stage model with recourse decision or a rolling horizon procedure described below (see [Cordeau et al., 2014] for an application).

As mentioned, the multi-period problems tackled in this thesis assume the availability of information over several periods and interactions between the decision periods. So, there is a temporal framework for the decision process and one for the information revealed at each decision step. Consequently, the previous notions might be used simultaneously such as a time-dependent vehicle routing problem with time-windows or in chapter 6 a multi-period problem with time-windows.

### 2.3.2 Dynamic decision process in transportation

In this section, the reason for the occurrence of dynamic decision process is depicted and links with other optimization areas are exposed. Secondly, four classes of models are exposed: 2-stage, rolling horizon, on-line and real-time.

Owing to the development of sensors, telecommunication and information systems bringing storage as well as computational power, costs for reporting data from the field have highly decreased since the change from paper-based organizations to wireless ones. For strategic projects or stable environments, a long computing time can be afforded if the decision is only performed once, e.g. for finding the best tour of a fixed set of customers. Moreover, the cost of one-time data acquisition can be high. If this investment in data collection can be recovered over the long run, for real-time operational problems having fluctuations in the parameter values, this data acquisition should be cheap and fast.

Optimization is used as a decision tool for several processes in factories: among others inventory management and production scheduling. These domains often require successive operational decisions linked to each other. Therefore, decision processes are usually dynamic (i.e. repetitive) by nature in these fields. Moreover, there was a small delay between the operations management in manufacturing (called production management) and transportation which take place outdoors. In fact, Global Positioning System (GPS) system and wireless telecommunications (Global System for Mobile Communication GSM) appeared later than Computer Integrated Manufacturing (CIM) based on wired local area networks (LAN). For this reason, machine control has developed its own terminology (states, actions, transition function) related to a production environment. As a consequence, it is possible to find similarities between production and transportation frameworks but this requires a transposition of the theoretical notions. [Chand et al., 2002] provide a survey of the literature on rolling horizon models in operations management including applications in production and transportation. Few of these references deal with stochastic problems. Therefore, we do not further refer to production planning and scheduling inside this thesis.

The dynamic aspect in a sequential decision process lies in the event or the cause that leads to induce a new decision. As a consequence, the amount of time between two decisions can be regular or irregular according to that cause. In the present thesis, problems are multi-period with regular periods of time between subsequent decisions. The frequency of the decision sequence does not depend on any event realization. Yet in some cases, the dynamism of the decision process is induced by the realization of an event, usually a stochastic one. So, dynamism and stochasticity are sometimes embedded together creating confusion. Therefore, the dynamic de-

cision process should be classified by terms explaining the rhythm of the decision process and furthermore if the delays between decisions are irregular, the stochastic events that trigger the decision process should be mentioned. For instance, these causes can be a parameter modification (travel time) or an incoming variable insertion (new customer) or a customer demand revelation along a tour. This second classification related to stochastic event linked to an information is discussed in section 2.4. Hereunder, four of these terms are explained.

- **2-stage** problems correspond to a 2-step decision process where an initial decision is performed and applied. Afterwards, in a second step, all information is revealed and either some predefined actions (such as recourse actions: going back to the depot, rerouting or facing a penalty) or at once a set of corrective actions is performed (see [Secomandi, 2000]). In that case, the initial solution is supposed to be feasible, as in the news-vendor problem or the VRP with stochastic demands. The second stage appears when a breaking point has been reached such as the stock-out of a newspaper in the newsvendor problem or a too high demand in the VRP with stochastic demand. So, the length of the first stage can be variable as the stock-out can occur along the day for the newsvendor or the truck along its tour, while the overstock is revealed at the end of the day or after the tour completion. The second stage represents the realization of unknown data. Decisions at the second stage correct the initial plan if needed. Typically, the problem and the actions of the second-stage are not of the same nature as the first stage part and decisions.
- **In a multi-period rolling horizon process**, the periodicity of the decision process is initially fixed. Consequently, at predefined time-steps, also called epochs, the available information is updated. Then, a decision based on similar information and leading to similar actions as in the previous period is performed. There might be some special periods such as the initial one and the last one. This process is explicitly discussed in chapter 3 as it is the core of the transportation problems investigated in this thesis.
- **On-line**: literally means that the system is connected continuously or at a specific frequency to a system providing information (see [Psaraftis, 1995]). When a specific event occurs, it does not force a decision to be taken but it updates information. If some threshold value is exceeded or particular information comes in, a decision might be generated with fixed or variable times between information collection and decision. So, an on-line decision process might lead to update a present solution, to modify it or to regenerate a new set of decisions. By definition, on-line should be used as a technical term to define the information flow communication process. However,



by extension, it became also the decision process (on-line optimization, on-line control) which might be a bit confusing. As explained in [Van Hentenryck & Bent, 2006], initially on-line decision systems only considered deterministic information. Currently, on-line anticipatory algorithms are developed to embed non-confirmed information issued from probability distribution in order to improve the algorithmic performances. This leads to on-line stochastic optimization. The present work follows that trend of incorporating stochastic information in the model but the updates of information are fixed at predefined epochs.

- **Real-time:** is close to on-line but it is supposed to be more reactive to any new information coming in. This hard constraint assumes that the decision process can be as rapid as the data acquisition process. In that sense, real-time should be considered as an extreme case where the period length is supposed to tend to zero. Moreover, in that specific case, if no extra-piece of information happens, the previous solution is kept, while any new information generates an optimization process. Consequently, the major difference between multi-period and real-time problems is that in general, more information is revealed and more actions are applied in the former case. As a consequence, in the present work, a solution is less frequently generated than the current one is updated in real-time optimization. However, the next decision period solution is fully renewed as a significant set of incoming and outgoing elements have to be taken into account. Notice that in practice, on-line and real-time are often used for each other while on-line should stem for a data acquisition system and real time for the rapidity of a decision process.

The present thesis is dealing with multi-period transportation problems wherein the decision process is induced by fixed periods according to the commercial or operational booking process. This requires to accomplish tasks, typically daily, for the next day as orders have to be planned between partners, such as freight forwarders, carriers and drivers. Updates or new information coming in (mainly loads to be transported) are linked to the business activity and might encompass several days in advance. These data contained within the forthcoming periods can be deterministic or stochastic. In the present cases, some parts of the forecasts are represented by stochastic elements. Therefore, in the present work, the dynamism of the decision process is regular and not enforced by the realization or not of a stochastic event. Decisions are taken at equal intervals in time. The duration of these intervals are not influenced by the information coming in during this period of time. A stochastic customer demand is a good example to understand this difference. In a multi-period decision process, the customer demand is added within the period it occurs. Then, the decision process takes this information into account for optimiza-

tion at each regular decision periods until an action is performed for this demand. Conversely, within a real-time decision process for a taxi or emergency service, the appearance of the customer demand might generate immediately a decision and action process. In this last case, stochastic and dynamic might sound similar (as noticed in [Psaraftis, 1995]). For the multi-period setting, stochastic elements used as forecasts but not leading to a decision process are described in section 2.4.

## 2.4 Stochasticity in transportation

As the dynamism of the multi-period decision process is stated, the information embedded in the periods can be described. This information can contain various data from which a special set can be made of stochastic elements that will realize while time goes by. In this section, the different elements that can be stochastic in transportation models are listed (see section 2.4.1). Then, the focus is put on the probability distributions (see section 2.4.2). In our applications, both the decision variables and the stochastic variables are discrete. Both characteristics are important as they lead to discard some optimization techniques and to select a particular one (see section 4.1.3).

### 2.4.1 Stochastic information in transportation model

Starting from the global model of the GPDP from [Savelsbergh & Sol, 1995], one can understand which elements in the model might vary and generate a stochastic model in transportation.

*"Let  $N$  be the set of transportation requests. For each transportation request  $i \in N$ , a load of size  $\bar{q}_i \in \mathbb{N}$  has to be transported from a set of origins  $N_i^+$  to a set of destinations  $N_i^-$ . Each load is subdivided as follows:  $\bar{q}_i = \sum_{j \in N_i^+} q_j = -\sum_{j \in N_i^-} q_j$ , i.e., positive quantities for pickups and negative quantities for deliveries. Define  $N^+ := \cup_{i \in N} N_i^+$  as the set of all origins and  $N^- := \cup_{i \in N} N_i^-$  as the set of all destinations. Let  $V := N^+ \cup N^-$ .*

*Furthermore, let  $M$  be the set of vehicles. Each vehicle  $k \in M$  has a capacity  $Q_k \in \mathbb{N}$ , a start located in  $k^+$ , and an end location  $k^-$ . Define  $M^+ := \{k^+ | k \in M\}$  as the set of start locations and  $M^- := \{k^- | k \in M\}$  as the set of end locations. Let  $W := M^+ \cup M^-$ .*

*For all  $i, j \in V \cup W$  let  $d_{ij}$  denote the travel distance,  $t_{ij}$  the travel time, and  $c_{ij}$  the travel cost. (...)*

*For each  $i \in V \cup W$  a time window  $[e_i, l_i]$  is introduced denoting the time interval in which service at location  $i$  must take place. Given a pickup and delivery plan and*

departure times of the vehicles, the time windows define for each  $i \in V$  the arrival time  $A_i$  and the departure time  $D_i$ . Note that  $D_i = \max\{A_i, e_i\}$ .

*If  $A_i < e_i$ , then the vehicle has to (\*) wait at location  $i$ .*

Note: (\*) rarely, some models forbid to wait at customer plant.

As many constraints or particular sets can be considered, this model can yield many formulations representing different problems, even in a deterministic environment.

Nevertheless, it is also possible to enrich the model by assessing some stochastic elements over some sets of parameters (see [Gendreau et al., 1996]):

- stochastic customers or transportation request:
  1.  $N$ , the customer or the transportation request may exist or not (see [Thomas, 2007] or chapter 7). In that case, waiting strategies or redirections in promising locations can be developed.
  2.  $N_i^+$  or  $N_i^-$ , the origin or the destination city might be unknown. It is often a consequence of stochastic customers. For example, in emergency services, the patient is transported to the hospital but the origin point is unknown until the demand exists (see [Gendreau et al., 2001] and [Brotcorne et al., 2003]).
- stochastic demands:  $\bar{q}_i$ , the quantity to be transported as in [Secomandi, 2000]. In that case, recourse policies or re-optimization can be useful.
- stochastic travel times:  $t_{ij}$  (see [Pillac et al., 2013] for references)
- stochastic release dates:  $e_i$  (see [Albareda-Sambola et al., 2014] and see chapter 6 for an application). The probability distribution in that case is conditional. If an item does not show up in the first period of the time windows, it increases proportionally the probability distribution of appearance in the remaining periods of the TW. In that sense, this conditional distribution law leads to a "dynamic" stochasticity in the sense that it evolves over time and not according to a decision process.
- stochastic service times: not mentioned in the previous model stochastic service times can occur at the pick-up or delivery places, even if the vehicle was on time at arrival. It refers to the loading or unloading operation durations that might vary according to unpredictable events at customer's plant.

Indeed, from a theoretical point of view, it is possible to assume other stochastic effects in the model: probability of availability of trucks in case of breakage ( $M$ ), probability of cost value ( $c_{ij}$ ), but these issues are not usual in practice and therefore rarely investigated (see [Pillac et al., 2013]).

### 2.4.2 Types of probability distributions

Regarding the stochasticity itself, an important point resides in the probability distribution function that leads to discrete or continuous random variables. In the case of stochastic customers or transportation request, the probability of a transportation request  $i$  to appear is provided by a parameter  $p_i \in [0, 1]$  generating a Bernoulli probability distribution (see chapter 7 for an application). This probability law induces discrete random variable realizations. In a scenario sampling approach (see chapter 4), this would generate various scenarios representing the presence or not of transportation requests. Consequently, this can lead to drastically opposed scenarios.

This mix of discrete or continuous probability distributions and discrete or continuous decision variables allows to use or not some optimization techniques. Inside this thesis, an advantage of the discrete distributions defined over a restricted subset of values is that the total number of possible scenarios is finite. With a continuous probability distribution, the total number of scenarios is infinite. Therefore, the techniques and the number of scenarios required to approximate the distribution can be different. Similarly, with discrete decision variables the total number of decisions is reduced but they might be very different from one scenario to another owing to the combinatorics of the decision framework. Moreover, optimization techniques are usually more complex for discrete decision variables as they are associated with integer programming models. With continuous decisions variables, some efficient techniques may be available like the simplex algorithm.

In the present work, applications deal with discrete probability distributions defined on a restricted set for the random variables. Moreover the decision variables are also discrete as they are binary. This discretization is not mandatory for some parts of the proposed methodology, except for the restricted range of scenario-based algorithms detailed in chapter 4.

## 2.5 Problems in transportation management tackled in the thesis

Transportation management represents a wide range of issues. In the literature, transportation problems are often modeled as richer versions of basic theoretical ones, already hard to solve by optimization techniques. As huge improvements have occurred in the optimization capability of computers, data management, programming languages and algorithms, more realistic problems have been tackled: VRP, GPDP, DARP with TW, real-time decision processes, stochastic issues... From an operational research point of view, this is useful in order to formulate the problem by reckoning up additional constraints. From a practical point of view, for managers, it is also time now to reverse the paradigm to understand what is not usually included within these models such as for example repetitive decision making and stochastic information.

Among these usually removed assumptions, two are investigated in the present work: dynamic decision processes and stochastic information. Therefore, the present thesis aims at proposing a methodology to investigate these two aspects for particular cases as we deal with long-haul transportation:

- the dynamic decision process is a multi-period decision process;
- data are provided over a rolling horizon;
- the probability distribution of the stochastic parameters is defined over a small finite set;
- the decision variables are binary.

These characteristics might seem to reduce the scope of applications to a narrow range of problems in transportation. But, some parts of the methodology can be transposed to other environments such as real-time short-haul problems. Discussing the possibilities of deployment of this methodology in other transportation problems is out of the scope of this thesis. However, some perspectives will be proposed in chapter 8.

**Part II**

**Methodology**



## Chapter 3

# Multi-period optimization

### 3.1 Introduction

According to the introduction, the multi-period optimization problems considered in this part are related to a sequence of decisions performed successively, but not independently. If some data for forthcoming periods are also available, the sequence of updates and decisions is called a rolling horizon process. This section depicts this decision making framework and mostly the data representation within the rolling horizon process. Terminology, information acquisition, repetitive optimization valuation, warm-up and end of horizon biases are also explained. Moreover, theoretical bounds and some concepts related to the value of information are presented.

### 3.2 Multi-period rolling horizon process

A multi-period rolling horizon process in transportation is based on a partition of time into periods:

- The current decision period is referenced as period  $t$ . In the present work, the information is fully revealed, meaning deterministically provided, in this decision period. Hence, decisions lead to actions effectively performed without facing penalties for infeasible decisions. Moreover, actions are only taken in this decision period. This means that the solution over the periods which are not the current decision period  $t$  is fully renewed at the next decision step, so that no decision is frozen in the future.
- Some periods subsequent to period  $t$ , called look-ahead periods. For the applications in our thesis, the number of such periods, containing extra-piece of deterministic or stochastic information, is considered constant. In fact, the



transportation problems investigated here are operational processes that are performed in the long run with a stable number of periods containing forecasts. This number of look-ahead periods is labeled  $H$ , standing for the rolling horizon length. The decision period  $t$  is not included in it (for an application see section 7.8.6).

- There are periods subsequent to period  $t + H$  up to the last period (labeled  $T$ ), i.e. periods in the remote part of the horizon. In this thesis, these periods are not taken into account for the optimization process because no information is supposed to be available for these periods which will enter the rolling horizon while  $t$  rolls over.

The rolling horizon process is led by the indexation of  $t$  that evolves period per period from period 1 up to  $T - H$  where  $T$  represents a finite number of periods or an approximation of an infinite time. The decision period  $t$  becomes  $t + 1$  when the horizon rolls.

At cruising speed, the first step of the rolling horizon process is based on an optimization process at decision period  $t$  taking into account the data belonging to this decision period  $t$  and the periods of the rolling horizon, i.e.  $t + 1, \dots, t + H$ . Therefore, at each decision step  $t = 1, 2, \dots, T - H$ , information belonging to  $1 + H$  periods are taken into account. The obtained solution is analyzed and some actions issued from this solution are performed. The costs or rewards of these actions are recorded into the policy value. This concludes the decision phase for the decision period  $t$ .

Then, data are updated when the horizon rolls from  $t$  to  $t + 1$ :

- some data enters the rolling horizon, typically within the last period  $t + H$
- some data status is modified (e.g. stochastic data becoming deterministic or probability distribution modification)
- some data are discarded from the rolling horizon as they are outdated, typically those from period  $t$ . In the case of stochastic customers, some demands projected in the rolling horizon might vanish completely. Indeed, when such a projection does not lead to a demand, it is removed from the model as the stochastic parameter is set equal to zero.

Finally, a new optimization process takes place in period  $t + 1$  and the sequence of decisions and updates is repeated. This setting can be sketched as follows:

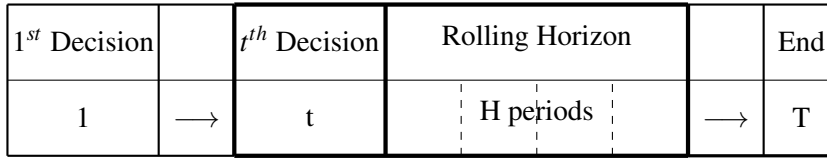


Figure 3.1: Rolling Horizon in period  $t$

In the present thesis, this framework is followed in the theoretical part and in the applications (see section 3.3). However, this rolling horizon framework is not unique in the literature. For some authors, the rolling horizon process can be deployed over a finite time and the look-ahead periods encompass the whole horizon since the beginning of the process. In this case, there are  $T$  decision periods in the set  $\{1, \dots, T\}$  and the rolling horizon length reduces from  $T - 1$  down to zero. For instance, a working day is optimized every hour for requests coming in along the day. In that case, the horizon reduces from 8 to 7, to 6...down to 1, 0 hour.

Moreover, for some authors dealing with on-line optimization of VRP with stochastic demands such as [Van Hentenryck & Bent, 2006] and [Yang et al., 2004], the whole horizon is considered open for decision making from the earliest decision state until the end of the horizon. Therefore, transportation demands taking place during the latest periods of the horizon can be confirmed during the earliest decision periods. This leads to a decreasing rolling horizon length with frozen decisions in the look-ahead periods. This optimization framework is different from the constant rolling horizon length process considered above allowing actions only in the current decision period  $t$ .

In some cases, there might also be a misunderstanding in the literature owing to the fact that authors do not state explicitly the rolling horizon taken into account, the flow of incoming information as well as the possibility to freeze or not decisions.

Note that, the length of  $H$  might influence the performance. At first sight, the longer  $H$  is, the better the performance. Nevertheless in practice, the dispatching process depends for example on the commercial process, called booking process in [Powell et al., 1995]. Therefore, it might be practically impossible to get forecasts at no cost. Consequently, the economic value of increasing  $H$  or the systemic loss should be evaluated (see sections 3.7 and 7.8.6).

### 3.3 Specific parts of the rolling horizon

The  $H$  look-ahead periods following the decision period  $t$ , which contain extra-piece of information, can be refined into several parts:

- The deterministic part of the horizon  $t, \dots, t + RH$  (where  $RH \leq H$ ) is made of subsequent periods following the decision period and wherein the information is fully revealed. The information contained within these periods cannot be modified afterwards.
- The stochastic part of the horizon  $t + RH + 1, \dots, t + H$ , follows the deterministic part up to the end of the rolling horizon. The data included in this part might change as the horizon rolls over (but might also include some piece of deterministic information).
- The action part of the horizon contains the periods where the decisions issued from the solution are effectively applied. Inside this thesis, for the applications, the action part is considered to be equal to the current decision period.
- The frozen zone covers the current decision period  $t$  and some successive periods of the rolling horizon. In these periods, actions previously planned can not be modified and no new actions can be inserted. For example, the decisions taken in day  $t$  might be applied in day  $t + 2$  while actions in  $t + 1$  were selected previously in  $t - 1$  (see e.g. [Tjokroamidjojo et al., 2006]). In Figure 3.2, the frozen zone length equals 2 periods. In other cases, (see [Keskinocak & Tayur, 1998]), some actions can be frozen in a remote period of the horizon, such as mandatory tasks (maintenance, going back home, specific transports) which are scheduled in advance. This action freezing is different from a frozen zone as it allows to modify some decisions variables and related actions in periods ahead of these frozen actions. In the present work, there are neither frozen zones, nor frozen actions ahead of the current decision period  $t$ .

Here is a sketch of the rolling horizon process with the different parts of the rolling horizon:

1 <sup>st</sup>	...	Deci.	Froz.	Act.	Deter.	Stoch.		...	End
1	→	t	t+1	t+2	t+3	t+4	t+5	→	T

Figure 3.2: Generic rolling horizon in period  $t$  with 2 frozen periods

Figure 3.2 shows a rolling horizon process containing  $H = 5$  look-ahead periods. The period  $t$  is the current decision period. The actions are frozen during the present period  $t$  and  $t + 1$  as previous actions have been implemented respectively in  $t - 2$  and  $t - 1$ . Similarly, from the decision generated in  $t$ , actions will be performed in period  $t + 2$ . The deterministic information is available for 3 periods ahead  $RH = 3$ . Stochastic information is in the remote part of the horizon  $t + 4$  and  $t + 5$ . Iteratively, the status of periods is updated according to the dynamic sequence of the rolling horizon process when  $t$  becomes  $t + 1$ .

Owing to the restrictions presented for this thesis, the generic rolling horizon is simplified as follows:

1 <sup>st</sup>	...	Deci.-Act	Deter.	Stoch.	...	End
1	→	t	t+1,...,t+RH	t+RH+1,...,t+H	→	T

Figure 3.3: Thesis simplified rolling horizon in period  $t$

### 3.4 Dynamism of the rolling horizon process

The sequence followed by the process should be detailed as it is a sequence of decisions and information updates. As an example, the dynamic and stochastic knapsack problem (in [Kleywegt & Papastavrou, 1998]) might lead to many sequences depending on when the information on the weight and/or reward of the item is revealed; whether items are inserted one by one or together; whether a discarded item can be selected again afterwards, etc.

The rolling horizon process is part of the problem definition itself and imposes to describe:

- the rolling horizon length  $H$ , the whole horizon length  $T$  and the decision periods  $t = 1, \dots, T - H$
- the information coming in at every time-step, in the deterministic part  $t, \dots, t + RH$  and in the stochastic part  $t + RH + 1, \dots, t + H$  of the rolling horizon
- the frozen zone length if necessary
- the solution found after optimization over the rolling horizon
- the procedure used to determine actions to be implemented in the actions periods
- the effects of these actions on the system
- the modification of the global objective function valuation owing to these actions
- the information becoming outdated after completion of these actions and the information becoming useless when the stochastic variables are replaced by deterministic values.

The sequence of periods, information status, solutions, actions make up the core of the rolling horizon process description. This process is problem dependent and should be detailed accordingly.

### 3.5 Policy versus solution

As mentioned, the actions performed over the long term are based on solutions issued from a sequence of optimization processes over restricted horizons. This is an important step of the multi-period process, as the aim is to value the decision making process in the long run. Depending on the problem, different algorithms may be available and the set of actions performed can vary according to the solution generated by these algorithms. Indeed, each algorithm  $\mathcal{A}_i$  from a set of algorithms  $\{\mathcal{A}_1, \dots, \mathcal{A}_i, \dots, \mathcal{A}_{NA}\}$  generates a set of solutions over the rolling horizon leading to implement the decisions within the action periods. This set of actions, called the policy  $\pi(i)$  of algorithm  $\mathcal{A}_i$ , contains each action labeled  $A^{\pi(i)}(S_t)$  for the system in state  $S_t$  in the decision period  $t$ . The total value (cost or reward) of all these actions

$C(S_t, A^{\pi(i)}(S_t))$  depends on the state of the system  $S_t$  in period  $t$  and the policy  $\pi(i)$ . This expected global value is called the policy value  $\mu_{\pi(i)}$ .

$$\mu_{\pi(i)} = \lim_{T \rightarrow \infty} \frac{E[\sum_{t=0}^T C(S_t, A^{\pi(i)}(S_t))]}{T} \quad (3.1)$$

Thus,  $\mu_{\pi(i)}$  is the average cumulated gain over the successive action periods. Note that the policy value is not the cumulative sum of all objective function values over all rolling horizons. In fact at each decision period, the performed actions are a restricted portion of the decision variables of the solution over the rolling horizon. Consequently, the measuring unit of both values, the policy and the rolling horizon solution might be different. For example in a transportation problem aiming at minimizing the costs, the policy value on the long-run can be valuated as an average transportation cost per ton per period while at each decision period of the rolling horizon the valuation of the objective function might be measured by the global total transportation cost over the rolling horizon.

The aim is to select the best policy  $\pi^*(i) = \pi^*$  among all  $\pi(i)$  and consequently the best algorithm  $\mathcal{A}^*$ . According to the objective function, the best policy  $\pi^*$  might be the one maximizing or minimizing the value  $\mu_{\pi^*}$ . For example, for a maximization problem:

$$\mu_{\pi^*} = \max_{\pi(i)} \mu_{\pi(i)} \quad (3.2)$$

Besides this, the policy requires a solution at each time step to generate a global decision leading to perform actions. This solution might be issued from rules, exact methods, heuristics or meta-heuristics. If the data are fully deterministic inside the rolling horizon, previous research dealing with a single period model can sometimes be of interest. As these models represent an important part of the literature in transportation, it is valuable to adapt them to a multi-period deterministic framework. Indeed, each decision making process might be an instance of the more studied static case of the problem, as in scheduling and routing (see [Pillac et al., 2013]). If the information includes stochastic data within the rolling horizon, dedicated techniques have to be used (see chapter 4).

An important point is how to deal with the time parameter in this framework, as "periods" are considered. These periods discretize time which is continuous by nature. Indeed, the rolling horizon process deals with a sequence of periods  $\{1, \dots, T\}$  but some piece of information linked to time might follow another time unit. For instance, the decision periods might be separated by days or hours, whereas the time-windows for pick-up or delivery of a given load may be provided in minutes (e.g. [Tjokroamidjojo et al., 2006]). Then, such information is allocated to a single

period even if the information arises between the start and the end of the period. In this thesis, every temporal information is given per period. Therefore, there are no reconciliations or rounding procedures between the time scale of the rolling horizon and the time scale of the data.

Another difficulty is due to the fact that models including time constraints, such as problems with time-windows, can be hard to solve, i.e. non-polynomially solvable. Consequently, solving optimally or heuristically a single fixed horizon might already be an issue. For instance, each fixed horizon optimization problem might be NP-Hard.

For the two applications that we consider, a preprocessing step based on the structure of the problem allows us to present two formulations of the deterministic versions of the problems where no index parameter for time exists. This leads to single period formulations computationally tractable by a commercial solver as optimization tool. As less computing time is required, test instances can be diversified and a larger number of periods  $T$  can be considered. More generally, for practitioners and for researchers, developing efficient formulations for each decision step is important because actions issued from the solution at each step make up the policy value. So, the rolling horizon process highly depends on the ability to find a solution at each decision step.

### 3.6 Start and end of horizon biases

When valuating the policy over the rolling horizon, potential biases influencing the results must be depicted. The earliest and the latest decision steps on the whole horizon might induce such phenomena, i.e., start and end of horizon biases.

When performing simulations, one can choose to use historical data to reproduce natural starting points. In practice, one often generates artificial instances to test algorithms over various types of instances. Therefore, starting points have to be randomly regenerated as well.

A way to alleviate these biases is to consider the process at cruising speed or in "stationary state". For example in [Tjokroamidjojo et al., 2006], the process rolls over  $t = 1, \dots, 20 - H$  with a rolling horizon including  $H = 4$  periods at most and authors present results from the decision period 5 to the decision period 15. The major problem of a multi-period optimization framework is that it is difficult to embed past decisions in the preliminary periods to simulate a steady state, because these previous decisions might be the consequences of a policy different from the one tested. Therefore, there would be an initial bias in the policy valuation. A remedy is to remove these initial periods of the whole horizon. This creates a warm-up delay

that allows the policy to take its own decisions up to a representative decision period including policy past effects and representative information inside the rolling horizon. As in [Tjokroamidjojo et al., 2006], a warm-up length equal to the decision period and the rolling horizon length  $(1 + H)$  is often used in practice. Warm-up periods leading to different initial conditions, also induce a bias as the starting points might be different for policies that have already rolled over  $1 + H$  periods.

Also, simulations cannot be run over infinite horizons and must be terminated. As far as the end of the whole horizon is concerned, one has to highlight that multi-period optimization is relevant if there are interactions between periods, meaning that an operation might be postponed, selected or rejected during the rolling horizon optimization. As a consequence, when an infinite horizon case ( $T \cong \infty$ ) is considered, in practice the whole horizon is reduced to a finite one. Due to the truncation and the rolling horizon procedure,  $T - H$  decisions are taken. Therefore, in practice for the applications, equation 3.1 becomes:

$$\mu(\pi) = \frac{E[\sum_{t=0}^{T-H} C(S_t, A^\pi(S_t))]}{T - H} \quad (3.3)$$

Thus, one cannot claim that the same set of actions have been considered by all policies when the truncation arises in  $t = T - H$ . For this reason, the horizon  $T$  should be long enough to reduce this bias. Policy values can also be evaluated per period in order to average the performance over the whole horizon and to be compared with real-life performances. Consequently, it is mandatory to take care of the biases that the start and the end of the whole horizon might induce in the final value of the policy.

For some authors [Hughes & Powell, 1988], the "end of horizon effect" has another meaning, as this effect is related to the last periods of the rolling horizon. Actually, for them, it represents the decreasing value of information in the remote periods of the rolling horizon. As many changes in the data status can occur before reaching the periods at the end of the rolling horizon, at each decision step, it might be interesting to give more importance to data closer to this current decision period than distant ones. To handle this, a decreasing or discount factor (such as  $\gamma$  in equation 3.4), representing a lower weight for the information in the remote periods of the rolling horizon, is inserted in the formulation to improve the policy performance. For example, in the following form of the Bellman equation 3.4 (see [Powell, 2011] and section 4.1.2 for details), the value of a decision  $V_t$  at period  $t$  when the system is in state  $S_t$  is estimated by :

- the cost  $C(S_t, A_t)$  of the decision  $A_t$ ,
- and the discounted ( $\gamma \in [0, 1]$ ) expected ( $\sum_{s' \in S} P$ ) value ( $V_{t+1}$ ) of this decision when it is applied ( $s' | S_t, A_t$ ) over all subsequent realizations ( $s' \in S$ )



$$V_t(S_t) = \min_{A_t \in \mathcal{A}_t} \left[ C(S_t, A_t) + \gamma \sum_{s' \in S} P(s' | S_t, A_t) V_{t+1}(s') \right] \quad (3.4)$$

So, here, the end of the horizon is mitigated ex-ante in order to improve the model and its algorithmic performance, whereas in our previous discussion, the effect appears only in the ex-post evaluation of the policy over a finite horizon.

### 3.7 Rolling horizon length

Obviously, the performance of the decision process might also depend on the rolling horizon length  $H$ . If the rolling horizon is long, its last periods might not influence the solution for the decision period (see [Chand et al., 2002]). For instance, the applied actions in  $t$  might not be affected if the last period  $t + H$  is removed from the rolling horizon and if the decision making process is performed over  $t, \dots, t + H - 1$ . At first sight, if the actions taking into account  $1 + H$  or  $H$  periods are equivalent, both solutions seems to lead to the optimal policy. This is a necessary condition, but not sufficient to prove the optimality of the policy issued from the decisions at each time step. This can be proved easily. Assume a taxi driver which should decide every 10 minutes to relocate his position from the train station to the airport to pick-up a customer for a profit of 2\$. The trip to the airport from the train station takes 60 minutes and its cost is 1\$. A customer is expected within an hour in the airport while the taxi is waiting at the train station. If the taxi driver has got a rolling horizon from 10 to 50 minutes, i.e. from 1 up to 5 periods, he will decide to stay at the train station, because it is more profitable to stay idle than to move. When the demand will be revealed, it will be too late to decide to move and pick-up the customer. If the rolling horizon is 60 minutes or more, the taxi would decide to move and would be on time to pick-up the customer. So, constant actions for several rolling horizon lengths do not prove the optimality of the policy.

Moreover, looking at it closely, increasing the rolling horizon length requires to still be able to solve the model for several periods while increasing the model size. However, the memory requirement and the computing time might make the optimization repellent in practice if many data are concerned. Beside this, getting these forecasts for a long time can be costly leading to a cost vs benefit analysis to decide whether or not some extra-piece of information is valuable.

Conversely, the rolling horizon might be short, so the loss for lacking of information should be valued. This can be done by a test reporting the policy values based on successive rolling horizon length  $H = 0, 1, 2, \dots$  compared to the fully revealed information over  $T$  (see section 7.8.6 for an application).

The influence of the length  $H$  on the policy performances was investigated previously (see [Sethi & Sorger, 1991] for references and [Powell, 1986] for a similar analysis). For both applications presented in the thesis, preliminary tests show improvements if a rolling horizon of at least one period  $\{t, t + 1\}$  is considered instead of a purely myopic decision based on the information available in the decision period  $t$ . Then, the benefit of increasing the rolling horizon length reduces asymptotically to zero in percentage within the few next periods.

From a managerial point of view, as the rolling horizon length  $H$  might be the consequence of the booking process between the freight forwarder and the transportation company, it might be interesting to analyze the process and to re-engineer a new one with a smaller or longer  $H$ , if this is profitable. However, in case of increase,  $H$  should remain commercially relevant, as the amount of time between the booking process and the realization of the order is also part of the service level.

On the other hand, the limitation of the length of the rolling horizon  $H$  might come from the difficulty for developing a dedicated algorithm able to handle longer rolling horizon instances.

As a conclusion, the rolling horizon length  $H$  is an important issue within these multi-period optimization problems and its potential influence should be measured.

### 3.8 Bounds with fully revealed information

Within an optimization process, when the optimal solution cannot be reached, it is important to assess the quality of the approximate solution found. In this section, some bounds based on fully revealed deterministic data are proposed to estimate the quality of the policies using the stochastic information which is the situation faced in reality. In a simulation process, some bounds can be generated using the deterministic information alias fully revealed information alias Oracle data. Three bounds can be generated according to the horizon length considered. Theoretically, the expected values of some policies can be computed over all possible scenarios according to the relative probability of occurrence of these scenarios, also called weight of the scenarios:

1.  $LO$  or  $O_{RH}^*$  the a-priori or myopic bound value based on a policy only using the information from the deterministic part of the horizon  $1 + RH$  and generating a policy called the myopic or a-priori policy.
2.  $O_H^*$  the a-posteriori bound value based on a policy only using the fully revealed information over the rolling horizon  $1 + H$  and generating a policy called the rolling horizon a-posteriori policy.

3.  $O_T^*$  the a-posteriori bound value based on a single solution only using the fully revealed information over the whole horizon  $T$ . In this case, the value of the policy is equal to the value of the solution over the entire horizon and not a cumulated sum of actions issued from the decisions taken in each decision period. In the applications, due to the truncation of the horizon (see equation 3.3), the optimal solution value  $O_T^*$  is estimated as  $O_{T-H}^*$ .

Referring to figure 3.2, one gets the following figures:

1. for  $O_{RH}^*$ , the rolling horizon reduces to 3 periods and the optimization process takes into account the revealed information 3 periods ahead of the present decision period.

1 <sup>st</sup>	...	Deci.	Froz.	Act.	Deter.	...	End
1	→	t	t+1	t+2	t+3	→	T

Figure 3.4: Information for the a-priori or myopic policy in period t

2. for  $O_H^*$ , the deterministic part of the rolling horizon increases up to 5 periods and the optimization process takes into account the revealed information 5 periods ahead of the present decision period.

1 <sup>st</sup>	...	Deci.	Froz.	Act.	Deter.			...	End
1	→	t	t+1	t+2	t+3	t+4	t+5	→	T

Figure 3.5: Information for the rolling horizon a-posteriori policy in period t

3. for  $O_T^*$ , the information is fully revealed over the whole horizon and the optimization process takes into account the revealed information  $T - 1$  periods ahead of the present decision period.

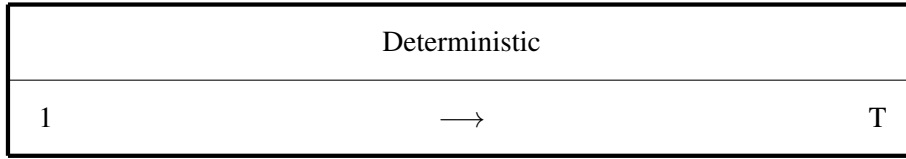


Figure 3.6: Information for the a-posteriori optimal policy

Finding these bounds using deterministic information is usually a bit easier than solving the true problem dealing with stochastic information. Indeed, only a single deterministic problem is solved at every iteration and not a stochastic one which might require a sophisticated algorithm (see chapter 4).

If the values of the policies are issued from optimal solutions of the deterministic problems, one can state that  $O_{RH}^* \leq O_T^*$  for a maximization problem and  $O_{RH}^* \geq O_T^*$  for a minimization one. A-priori solutions  $O_{RH}^*$  cannot outperform the a-posteriori optimal one  $O_T^*$ .

Besides this, the value of the rolling horizon a-posteriori policy  $O_H^*$  using the fully revealed deterministic information over  $1 + H$  periods should be in between these bounds. As a consequence, a lower and an upper bound on the rolling horizon a-posteriori policy exist. They are respectively the a-priori or the a-posteriori solutions for a maximization problem and conversely for a minimization problem. Indeed, one can find a worst solution than the a-priori solution (see [Yang et al., 2004]) such as it might be the case with manual optimization in company. Nevertheless, from an operational research point of view, the value of the myopic solution is supposed to be at least reached, if not the best policy using the a-priori information should be considered as the myopic bound.

In the same sense, the value  $O_H^*$  is assumed to be in between the lower and the upper bound. Yet, this cannot be claimed as sometimes the myopic policy performs better than the rolling horizon a-posteriori one (see [Blackburn & Millen, 1980]). Therefore, the following equations "usually" hold and these symbols are used  $\lesssim$  and  $\gtrsim$ :

- for a maximization problem:

$$O_{RH}^* \lesssim O_H^* \leq O_T^* \tag{3.5}$$

- for a minimization problem:

$$O_{RH}^* \gtrsim O_H^* \geq O_T^* \tag{3.6}$$

If there is a significant gap between the values of the bounds, then the value of the information can be economically valuated (see section 3.9). From a management point of view, similarly to what can be done for the rolling horizon length  $H$ , it is possible to measure the maximum earning that can be made by improving the acquisition of the deterministic information. Conversely, if the bounds are close and the gap economically irrelevant, it is noticeable to know that any improvement is useless and the myopic optimization over deterministic data might be efficient. This annihilates the difficulty to solve the stochastic version of the problem. This situation is also faced when the inequalities  $\lesssim$  and  $\gtrsim$  do not hold in 3.5 and 3.6.

In this introduction, we assume that each scenario, i.e. each deterministic model, is solved to optimality and that the bounds are estimated by the cumulated value of these optimal solution values for all the scenarios taking into account their relative weight. If the values for the bounds are estimated by the cumulated solution values generated at the optimization level of each deterministic model issued from a scenario by a heuristic or a meta-heuristic, proving that the previous inequalities (3.5 and 3.6) still hold is beyond the scope of this thesis (see chapter 8).

Finally, as the number of scenarios can be cumbersome, it cannot be practicable to compute the expected value of the bounds over all scenarios. If all scenarios and their relative weights are considered, the bound value is its expectation. If a restricted set of scenarios (considered as equiprobable) is used, the bound value is an estimate of this expectation. For the applications and for the forthcoming sections, the estimates and the expectations of the bounds are labeled in the same way. Yet, owing to this approximation, the significance of the gap between the estimates of the values should be statistically checked.

### **3.9 Value of information in multi-period optimization**

As introduced in the end of previous section 3.8, it might be profitable to solve the myopic version of the problem if the a-priori and a-posteriori bounds values are not different. Conversely, if gaps exist between the a-priori and the a-posteriori bounds, it might be interesting either to increase the deterministic part of the rolling horizon  $RH$  or to develop a policy using the stochastic information contained in the rolling horizon. Increasing  $RH$  or  $H$  (see section 3.7) requires to adapt the data collection within the booking process from the customer. This process re-engineering is not in the scope of this thesis even if a sensitivity analysis is performed for  $H$ .

Therefore the main focus of the practical part of this thesis lies on finding efficient policies using the stochastic information available in the rolling horizon. Similarly to the previous section, as simulation techniques (see chapter 4) are used to

valuate the policies  $\pi(i)$ , only an estimate of the expectation of the policy value  $\mu_{\pi(i)}$  can be reached. For the sake of simplicity and similarly to the bounds, estimates are labeled like the expectations. However, using estimates requires to assess a statistical significant difference between policy performances.

As deterministic lower and upper bounds are available, the policy values based on stochastic information over the rolling horizon,  $\mu_{\pi(i)}$  can be reported in a chart. This section introduces concepts to measure the values of the information between the deterministic bounds and the value  $\mu_{\pi^*}$  of the best policy found  $\pi^*$  using the stochastic information.

The optimal policy value using the stochastic information is labeled  $E^*$ . Note that, the optimal policy  $E^*$  might remain unknown and different from  $\pi^*$ .

Similarly to the rolling horizon a-posteriori policy  $O_H^*$ , a policy using stochastic information over the rolling horizon  $\pi(i)$  should be between the lower  $O_{RH}^*$  and the upper  $O_T^*$  bounds. If not, the extra-piece of stochastic information or the algorithm generating this policy would have a negative effect. Moreover, the best policy  $\pi^*$  should under-perform when compared to  $O_H^*$  which has the fully revealed information over the rolling horizon and no stochastic elements. Exceptionally, owing to the rolling process and for some instances,  $\mu_{\pi^*}$  might be better than  $O_H^*$  (see chapter 4). Therefore, in the same sense than in equations 3.5 and 3.6, the following symbols are used  $\lesssim$  and  $\gtrsim$ . As mentioned here above, the situation faced when the a-priori policy performs better than the rolling horizon a-posteriori one is profitable, because using a myopic policy is relevant. Similarly when the policy value  $\mu_{\pi^*}$  using the stochastic information outperforms the rolling horizon a-posteriori bounds  $O_H^*$ , the situation is interesting. Indeed, this means that the stochastic information is used in such a way that trying to get the fully revealed deterministic information is useless. Therefore, when these inequalities  $\lesssim$  and  $\gtrsim$  do not hold, the manager faces a beneficial situation. Unfortunately, in most cases, these inequalities remain and lead to:

- for a maximization problem:

$$O_{RH}^* \lesssim \mu_{\pi^*} \lesssim O_H^* \leq O_T^* \quad (3.7)$$

- for a minimization problem:

$$O_{RH}^* \gtrsim \mu_{\pi^*} \gtrsim O_H^* \geq O_T^* \quad (3.8)$$

The gaps between these values provides some information that can be useful to decide which actions can be profitable between data acquisition, algorithmic development or rolling horizon length increase. In the practical part, estimates are

computed, while theoretically expectations are considered. In order to be consistent with the admitted acronyms (see [Birge & Louveaux, 1997]), the expression "Expected" is used instead of "Estimated". Therefore, for a maximization problem, names and values of the intervals between these bounds and the best policy value are proposed hereafter. The **EVPI** is a standard notion (see [Birge & Louveaux, 1997]) while the others are contributions of this thesis.

- **EVPI**: Expected value of the perfect information

$$EVPI = O_T^* - \mu_{\pi^*} \quad (3.9)$$

- **EVMPM**: Expected value of the multi-period model

$$EVMPM = O_{RH}^* - O_H^* \quad (3.10)$$

- **EVAI**: Expected value of the accessible information

$$EVAI = O_H^* - \mu_{\pi^*} \quad (3.11)$$

- **EVTI**: Expected value of the tail information

$$EVTI = O_T^* - O_H^* \quad (3.12)$$

For a minimization problem, terms of the equations are inverted.

For a maximization problem, values can be reported in a chart as follows:

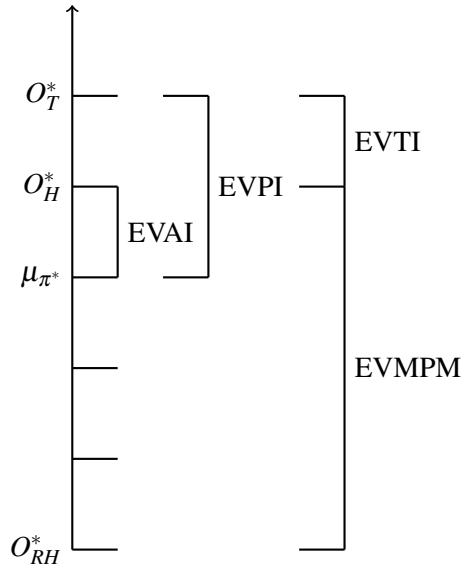


Figure 3.7: Bounds and values of information

- The **EVMPM** shows how interesting it is to treat the problem as if the data available on the rolling horizon were deterministic (i.e. were accessible) and considered within the optimization process. This means evaluating the cost for keeping the process myopic and for not using information from the rolling horizon. If this EVMPM cost is high, the consequence would be to improve the data collection (increasing  $RH$ ) or to develop an algorithm dealing with the stochastic information over the rolling horizon. Conversely, if this EVMPM cost is low, it might be interesting to use only the deterministic part of the horizon  $RH$  or even to reduce this deterministic part of the horizon. Indeed, if the booking process is too rigid by imposing several periods in advance for registration, the transportation company might lose commercial opportunity. Therefore, even reducing  $RH$  might also be an option highlighted by a low EVMPM.
- The **EVPI** shows how interesting it would be to find a solution with the fully revealed information over the whole horizon compared to the best optimization technique using the information presently available over the rolling horizon. The consequence could be to try to improve the performance of the stochastic solution  $\mu_{\pi^*}$  by an algorithmic development or to increase the rolling horizon length  $H$  or by inserting more information at the end of the rolling horizon such as stationary periods (see [Sethi & Sorger, 1991] or [Powell, 1988]).
- The **EVAI** shows how interesting it is to treat the problem as if the data were fully revealed over the rolling horizon compared to the best solution found using the stochastic information also over this rolling horizon. The consequence would be to increase the deterministic part of the rolling horizon (e.g. by improving the booking process) or if possible to develop a more performing algorithm using the stochastic information. So, the **EVAI** helps to decide what to do between the two options provided by the EVMPM:  $RH$  increase or algorithmic development.
- The **EVTI** shows how much is lost due to the length of the rolling horizon  $H$  which might be extended. Therefore, when the EVPI is high, the **EVTI** helps to select which options between  $H$  increase or  $\mu_{\pi^*}$  improvement would be the more profitable.



However, measuring these values shows the potential benefit of actions but do not prove that it is feasible. For instance, it is not always possible to improve the information by transforming forecasts into confirmed orders, so increasing  $RH$  is not obvious. In the same sense, gambling on a performance improvement for an algorithm using the stochastic information of the rolling horizon  $\pi(i)$  can be risky. Actually, the optimal policy for the stochastic problem  $E^*$  remains unknown and can be approximated by  $\pi^*$  but  $E^* = \pi^*$  cannot be usually proved particularly with estimated values.

So, improving  $\pi^*$  is not an obvious option. Inside this thesis, simple optimization techniques to get a performing  $\pi^*$  are proposed (see chapter 4).

Finally, these values of information are often provided normalized as a percentage of the bounds such as in [Pillac et al., 2013] and chapters 5, 6 and 7.

## Chapter 4

# Stochastic optimization problems

### 4.1 Introduction

This chapter depicts in its first section 4.1.1 the standard model used in stochastic programming. This formulation fits best for two-stage problems and is not suited for a multi-period problem whereas the present work aims at minimizing the expected cost (or maximizing the expected reward) in a multi-period framework.

In section 4.1.2, four families of policies to approximate the expected cost of the optimal policy are presented: myopic policies, lookahead policies, policy function approximations and policies based on value function approximation.

Then, linked to these policies, scenario-based models are described in section 4.1.4 leading to related optimization techniques while some discarded optimization techniques are exposed in section 4.1.3.

These algorithms are successively based on mono-scenario approximation (section 4.2.1) and multiple scenario approaches (section 4.2.2).

Then, in section 4.3, the policy values from these algorithms using the stochastic information are inserted in the bounds presented in chapter 3 and values of the stochastic information are introduced.

#### 4.1.1 Stochastic programming

Stochastic programming is a framework for modeling optimization problems with uncertainty. As mentioned in [Kall & Wallace, 1994]: *"Within the mathematical programming community, it is common to split the field into topics such as linear programming, nonlinear programming, network flows, integer and combinatorial optimization, and, finally, stochastic programming. Convenient as that may be, it is*

conceptually inappropriate. It puts forward the idea that stochastic programming is distinct from integer programming the same way that linear programming is distinct from nonlinear programming."

Let us follow this reasoning. The standard form of a linear programming problem, using a matrix-vector notation, is as follows:

$$\left. \begin{array}{l} Z^* = \min (c^T x) \\ s.t. Ax = b \\ x \geq 0 \end{array} \right\} \quad (4.1)$$

where  $c^T$  is the cost vector,  $A$  is a vector of parameters such as weight or distance for a problem in transportation while  $b$  is the corresponding constraining parameter vector such as capacity or tour length. Finally,  $x$  is the vector of decision variables. Values of vectors  $c^T$ ,  $A$ ,  $b$  are assumed to be known deterministically.

If some constraints or the objective function are not linear, but expressed by more general functions  $g_i(x)$ , one can represent the problem by a mathematical programming model such as:

$$\left. \begin{array}{l} Z^* = \min (g_0(x)) \\ s.t. g_i(x) \leq 0, i = 1, \dots, m \\ x \in X \subset \mathbf{R}^n \end{array} \right\} \quad (4.2)$$

So, depending on the set of functions  $g_i$ , the problem is linear or non-linear. Similarly, if some or all  $x \in X$  are required to be integer, the problem is respectively called mixed-integer programming (MIP) or integer programming (IP) problem (see [Kall & Wallace, 1994]).

In order to introduce stochastic programming models, the stochastic information is assumed to be modeled by a random vector  $\omega$  taking its values over a set  $\Omega \subset \mathbf{R}^l$  where  $l$  is the number of dimensions of the random elements. We assume that the probability distribution of  $\omega$  is known. Then, the objective function and the constraint functions can be viewed as functions of  $x$  and of  $\omega$ , to be denoted  $g_0(x, \omega)$  and  $g_i(x, \omega)$ . Therefore, a general formulation of a stochastic program can be formalized as follows:

$$\left. \begin{array}{l} Z^* = \text{"min"} (g_0(x; \Omega)) \\ s.t. g_i(x; \Omega) \leq 0, i = 1, \dots, m \\ x \in X \subset \mathbf{R}^n \end{array} \right\} \quad (4.3)$$

where  $g_0(x; \Omega)$ ,  $g_i(x; \Omega)$  denote "aggregate" values of  $g_0(x, \omega)$ ,  $g_i(x, \omega)$  over  $\Omega$ . In that framework, the constraints can be expressed in a meaningful deterministic way. Typically, we may take  $g_i(x; \Omega) = E_{\Omega} g_i(x, \omega)$  or  $g_i(x; \Omega) = \max_{\omega \in \Omega} g_i(x, \omega)$ .

So, stochastic programming is quite different from the usual mathematical programming model of equation 4.2, because some information materializes after the decision is taken.

For example, a two-stage stochastic program with recourse assuming that the information is fully revealed after the decision is taken can be formulated as follows (see [Birge & Louveaux, 1997]):

$$\left. \begin{aligned} Z^* &= \min (g_0(x) + E_{\Omega} Q(x, \omega)) \\ s.t. & g_i(x; \Omega) \leq 0, \quad i = 1, \dots, m \\ x &\in X \subset \mathbf{R}^n \end{aligned} \right\} \quad (4.4)$$

where  $g_0(x)$  estimates the first stage cost and where  $E_{\Omega} Q(x, \omega)$  is the expectation of the second-stage problem valuating the recourse cost over  $\Omega$ . This recourse cost is valuated as  $q(x, y, \omega)$  where  $q$  is the recourse cost function,  $\omega$  is the observed realization of the random vector, and  $y$  the associated second stage decisions. Thus, the second-stage problem can be formulated as follows:

$$\left. \begin{aligned} Q(x, \omega) &= \min q(x, y, \omega) \\ s.t. & h_i(x, y, \omega) \leq 0, \quad i = 1, \dots, m \\ y &\in Y \subset \mathbf{R}^m \end{aligned} \right\} \quad (4.5)$$

where  $h_i(x, y, \omega) \leq 0$  represents the second-stage constraints taking into account decision  $x$  from the first stage and the value of  $\omega$ .

In case of a discrete and finite distribution, each scenario ( $\omega \in \Omega$ ) is an independent outcome of the distribution and is therefore a potential second-stage. The model can be decomposed per scenario for the second-stage, so that the recourse cost can be estimated independently per scenario. So, the expectation of the global recourse of an action can be computed over all scenarios taking into account their relative probabilities  $p_{\omega}$ . Therefore, equations from models 4.4 and 4.5 become:

$$\left. \begin{aligned} Z^* &= \min (g_0(x) + \sum_{(\omega \in \Omega)} p_{\omega} Q(x, \omega)) \\ s.t. & g_i(x, \omega) \leq 0, \quad i = 1, \dots, m \\ x &\in X \subset \mathbf{R}^n \\ \text{where } Q(x, \omega) &= \min \{q(x, y, \omega), \text{ s.t. } h_i(x, y, \omega) \leq 0\}, \quad \omega \in \Omega \\ y &\in Y \subset \mathbf{R}^m \end{aligned} \right\} \quad (4.6)$$

In the present work, such discrete and finite distributions are investigated. However, the number of scenarios  $\omega \in \Omega$  is so large, that the expectation of the recourse

cost can only be approximated even for each decision period. Moreover, even for a single scenario, the cost function  $q(x, y, \omega)$  can only be approximated as the information is not fully revealed after the current decision period. Actually, some new stochastic information is incorporated in the model at the end of the rolling horizon, while only a part of stochastic information becomes deterministic, i.e., is fully revealed. Indeed, all data are not provided when  $\omega$  is revealed and there is still unknown information that has to be revealed period per period in the remote part of the horizon. Thus, the problems in transportation investigated in the present work are not two-stage ones (as in formulation 4.6), but multi-period ones. Usually, the formulation of this kind of "multi-stage" model is based on a composition of several sequential 2-stage models. This leads to a sequence of decisions and revealed information observations. In this work, for such multi-stage models, similarly to the 2-stage model, estimating the expected recourse cost of an action leads to approximate it, because of the large number of scenarios and because the information is not fully revealed after each stage. So, approximate models should be used to find a solution per period which should be itself embedded into a policy (see section 3.5). This thesis aims at proposing such approximate models or optimization techniques for stochastic problems.

Note that, the segregation, linear vs non-linear, MIP or IP for  $x$  (and  $y$ ) also stands for stochastic programming. This shows that stochastic programming is not a branch, but another trunk in mathematical programming.

Regarding the objective function, the labeling "min" deserves some explanations. In some applications, the objective function based on  $g_0(x; \Omega)$  might include significant penalty costs in case of non-fulfillment of an action. For instance, in emergency services facing uncertain demand, the model attempts to minimize the maximal unserved demand, generating a min-max model. In dial-a-ride problems (DARP), the objective might be to minimize the global or individual maximal inconvenience. In such optimization problems, the solution tends to be immune to the data uncertainty, i.e. robust (see [Bertsimas & Sim, 2004], [Bertsimas et al., 2011] and section 5.3). For instance, in some cases, the optimization takes into account the worst case of the outcomes of the probability distribution as every other outcome would be less damaging. Obviously, if that kind of robust solution minimizes the penalty costs issued from the stochastic information realization in the second stage, it often overestimates the cost for the first stage. This is legitimate when high penalty costs or infeasibility (infinite penalty) have to be faced.

In other contexts, such as in operational models in transportation presented in this thesis where the emphasis is placed on economic performance over the long run, objective functions tend to minimize the expected cost or maximize the expected reward. In these cases, the robustness of the solution is related to the ability to stay

close to the optimal solution even if some data used during the optimization phase were erroneous or imprecise. In that case, extreme scenarios should be sporadic and economically affordable. Consequently, the term "min" may represent several goals in stochastic optimization.

#### 4.1.2 Expected cost minimization approaches

According to [Powell et al., 2012]: "*Stochastic optimization problems arise in many settings, and as a result, a wide range of algorithmic strategies have evolved from communities with names such as Markov decision processes, stochastic programming, stochastic search, simulation optimization, reinforcement learning, approximate dynamic programming and optimal control.*" These authors classify also the policies in four classes wherein each policy approximates the solution for the current period by adapting Bellman's equation (e.g. equation 3.4) with a discount factor  $\gamma = 1$ : Policy function approximation, Myopic cost function approximations, Lookahead policies and Policies based on value function approximations. The differences between these ranges of policies lay in :

- the availability and utilization of the information from the current decision period  $t$ ;
- the availability and utilization of the information in periods subsequent to the current decision period;
- the optimization or not of a decision at each decision step;
- the exact or estimated valuation of the consequences of an action in the next periods.

1. **Policy function approximation:** in this approach, there are rules or functions that automatically generate the decisions without optimization and without taking into account extra pieces of information beyond the structure of the present state itself. In fact, the information in the subsequent periods might not be known explicitly, but globally. For example, in inventory management, the policy can be periodic, meaning that regularly, the inventory is replenished up to a level  $L$  estimated according to an average demand over time. To alleviate the variations in the demand, a security stock level can be taken into account when estimating  $L$ . The computation of the replenishment level is done once for all and replenishment decisions are taken at fixed epochs  $t$ . Quantitatively, each replenishment decision  $A_t$  depends on the inventory level  $S_t$  and  $A_t = L - S_t$ . This decision policy is reduced to a simple function depending on the current state of the system and to an approximation of the action to be performed.

This policy might be optimal in case of a constant demand rate. However, in case of variations of the demand, a safety stock has to be evaluated to avoid stock-outs. Therefore, in each period  $t$ , the replenishment to  $L$  is no more an optimal action, but an approximation that should lead to an efficient policy in the long run.

Similarly, in transportation management, a periodic policy (see [Angelelli & Speranza, 2002]) or a returning to the depot policy (see [Gendreau et al., 1996]) can be used to approximate multi-period or multi-customer delivery processes.

2. **Myopic cost function approximations:** if no information from the periods of the rolling horizon is explicitly revealed or cannot be used efficiently, it is possible to optimize the decision while restricting the data to the information available in the current decision period  $t$ . This leads to a so-called myopic policy. In that case, the policy computes a solution in period  $t$ , called action  $A_t$  belonging to the set of all possible actions in period  $t$ ,  $\mathcal{A}_t$ . This solution is taken according to the current state of the system, called  $S_t$ , while minimizing the cost generated by this action, labeled  $C(S_t, A_t)$ .

$$V_t^{Myopic}(S_t) = \min_{A_t \in \mathcal{A}_t} \left[ C(S_t, A_t) \right] \quad (4.7)$$

The real current state or cost function can be modified if this leads to more efficient results on the long-run: that is,  $S_t$  and  $C(S_t, A_t)$  can be approximated to select a more efficient  $A_t$  over the rolling horizon while using a restricted set of information. The value  $C(S_t, A_t)$  is added to the policy value. In our methodology, this policy can be used to estimate the myopic bound  $O_{RH}^*$ , if the revealed horizon length  $RH$  equals 0 (e.g. application 1).

3. **Lookahead policies:** if data are available for several periods subsequent to the current decision period  $t$ , it is possible to optimize while taking into account the future periods  $t' = t + 1, \dots, t + LA$  including deterministic (see equation 4.8) or stochastic information (see equation 4.9).  $LA$  is the number of lookahead periods taken into account at each decision step and  $LA = H$  in the rolling horizon procedure. The value  $V_t(S_t)$  estimates the policy value over several periods if action  $A_t$  would be applied. However, only the current state decisions  $A_t$  are applied and its related cost  $C(S_t, A_t)$  is added to the policy value. So, there is a gap between the estimate of the policy value over the lookahead periods and the value added to the policy at each decision period  $t$ .

$$V_t^{Lookahead\ deterministic}(S_t) = \min_{A_t \in \mathcal{A}_t} \left[ C(S_t, A_t) + \sum_{t'=t+1}^{t+LA} C(S_{t'}, A_{t'}) \right] \quad (4.8)$$

$$V_t^{Lookahead\ stochastic}(S_t) = \min_{A_t \in \mathcal{A}_t} \left[ C(S_t, A_t) + \sum_{\omega \in \Omega} p(\omega) \sum_{t'=t+1}^{t+LA} C(S_{t'}(\omega), A_{t'}) \right] \quad (4.9)$$

Repeated over the whole horizon, the lookahead deterministic policy (see equation 4.8) can be used to value the bounds (say  $O_{RH}^*$  or  $O_H^*$ ) based on the revealed deterministic information. In these cases, the number of lookahead periods  $LA$  equals respectively  $RH \geq 1$  or  $H$ .

Repeated over the whole horizon, the lookahead stochastic policy (see equation 4.9) should be used to generate the policy value of the problem using the stochastic information over the rolling horizon. However, as explained above, the number of  $\omega \in \Omega$  is too large and the cost  $C(S_{t'}(\omega), A_{t'})$  can only be approximated such as in the following policies.

4. **Policies based on value function approximations:** in that case the decision is optimized in the present state while estimating the expected cost of the decisions taken in future periods. This value function approximation is linked to pricing techniques (see [Pillac et al., 2013]).

$$V_t^{Value\ function}(S_t) = \bar{V}_t(S_t) = \min_{A_t \in \mathcal{A}_t} \left[ C(S_t, A_t) + \mathbb{E} \{ \bar{V}_{t+1}(S_{t+1}) | S_t \} \right] \quad (4.10)$$

Of course, hybrid methods might be set up as mentioned in [Powell et al., 2012]. Within the methodology presented in section 4.2, several classes of policies are used: myopic ones for bounds, policy function approximation for some specific scenarios and lookahead policies including deterministic information trying to approximate the value of a decision over the following periods. As the problems are multi-period, using lookahead strategies seems obvious. However, as stochastic data are embedded in the lookahead periods, these problems are hard to solve to optimality because of the curse of dimensionality. Therefore, instead of introducing the stochastic information, only an approximate deterministic equivalent information is inserted in the lookahead periods. This deterministic information is usually one or some particular realizations of the probability distribution (scenarios) (see 4.1.4) leading to a policy based on value function approximation as in 4.10. In a sense, all models and techniques presented above and in this thesis generate several valuations of  $\mathbb{E} \{ \bar{V}_{t+1}(S_{t+1}) | S_t \}$ , as estimates of  $\sum_{\omega \in \Omega} p(\omega) \sum_{t'=t+1}^{t+LA} C(S_{t'}(\omega), A_{t'})$ .

Testing several realizations at each time step over the rolling horizon to find an efficient solution is called sampling while testing several realizations over the whole horizon to assess the statistical validity of the results is called simulation (see chapter 5). Hence, sampling assumes that one or several scenarios might lead to generate a decision that performs efficiently.



### 4.1.3 Discarded optimization techniques

Some techniques fit better when the probability distribution is continuous such as the normal distribution embedded in a 2-stage decision process while assuming continuous decision variables. The suitability of some algorithms is linked to the convexity or the ability to solve the second stage by linear programming (e.g. the L-shaped algorithm issued from 2-stage stochastic programming see [Kall & Wallace, 1994] and [King & Wallace, 2012]; or for the sample average approximation method (SAA), see [Vermeij et al., 2002] and [Kleywegt et al., 2001]). Consequently, these optimization techniques can be discarded in the present case of multi-period problems with discrete decision variables.

Conversely, some enumeration techniques are more suited to discrete probability distributions and discrete decision variables. For example, in a multi-period setting, models represented by a Markov Decision Process (MDP) can be solved by dynamic programming (DP) (see [Putterman, 1994]). Unfortunately, this approach highly suffers from the size of the problem. This is often the case with the exponential explosion due to the combinatorial effects among uncorrelated probability distributions. Therefore, action and state spaces are so cumbersome that the problem may become intractable for memory reasons or because of the number of iterations required.

Attempts can be made to overcome this difficulty by using an approximate dynamic programming approach (ADP) as described in [Powell, 2011]. Even if this technique seems theoretically promising, it is also not used in the present work. In fact, a target of this thesis is to focus on testing simple and easily implementable algorithms to check the added value of using the multi-period model and stochastic information in operational applications. As, in practice, results show high improvements and robust performances of the best developed algorithms, ADP was kept as a perspective in case of failure of previous approaches. In fact, owing to the huge size of the action and state spaces as well as the variability of the probability distributions over the horizon, the decision rules for each current decision period state is difficult to define by ADP. Actually, this technique tries to evaluate the overall future cost of an action according to the current state space based on the past and simulated future realizations of the stochastic parameters such as in equation 4.10. One difficulty is to define what is the state space in a multi-period environment.

If the current state space  $S_t$  contains only the information available in the current decision period  $t$ , the rule issued from ADP will take into account an average rolling horizon over  $t + RH + 1, \dots, t + H$  and not the information already available in the following periods. Conversely, if the state space includes the decision period and the complete rolling horizon, statistically each state tends to be unique and rare on

the mid or short-term. Therefore, ADP would not be able to anticipate most of the state space and to evaluate the approximate cost of an action. Finally, ADP can approximate the state space itself. This would lead to approximate the cost of an action from an approximate state space which might decrease the performance. Yet, ADP is conceptually exhibited as a strategic tool to define recurrent decision rules over a reduced state space (see [Simao et al., 2009]). For all these reasons, ADP has not been considered in our research.

In this thesis, due to the fact that we are dealing with finite discrete probability distributions, selected optimization techniques are scenario-based.

#### 4.1.4 Scenario-based models

To approximate the stochastic model expressed in equation 4.9, a scenario-based model approach is used over  $H$  lookahead periods. It estimates the value of the solution as in 4.10 on a reduced subset  $\Omega_K \subset \Omega$  where  $K$  is the number of scenarios considered. The value of the policy  $\mu_{\pi^*} \cong E^*$  is made of each decision step such that:

$$V_t^{Scenario-based\ model}(S_t) = \min_{A_t \in \mathcal{A}_t} \left[ C(S_t, A_t) + \sum_{\omega_k \in \Omega_K} \sum_{t'=t+1}^{t+H} C(S_{t'}(\omega_k), A_{t'}) \right] \quad (4.11)$$

Differences reside in

- the computation over a restricted set of scenarios  $\omega_k \in \Omega_K$ ,
- the removal of the relative weight of scenarios  $p(\omega)$ ,
- a family of cost functions approximating  $C$ .

In the present work, problems' characteristics lead us to select :

- a lookahead approach, because of the rolling horizon process
- a scenario-based approach, because of the discrete and finite distribution
- a value function approximation, because of the size of the problem forcing us to use a restricted set of scenarios

These approximate models are based on deterministic scenarios that shadow the stochastic part of the problem within the scenario generation process. These scenarios are particular realizations of  $\omega$ :  $\omega_1, \omega_2, \dots, \omega_k, \dots, \omega_K$ , where  $K$  is the number of deterministic scenarios generated to induce a decision over  $\Omega_K$  approximating  $\Omega$ .

Therefore, getting back to the standard formulation of the stochastic problem, in the present work, scenario-based approaches approximate formulation 4.3 as follows:

$$\left. \begin{aligned} Z^* &= \min (g_0(x; \omega_1, \omega_2, \dots, \omega_k, \dots, \omega_K)) \\ s.t. & \ g_i(x; \omega_1, \omega_2, \dots, \omega_k, \dots, \omega_K) \leq 0, \ i = 1, \dots, \ m \\ x &\in X(\omega_1, \omega_2, \dots, \omega_k, \dots, \omega_K) \subset \mathbf{R}^n \\ k &\in 1, \dots, \ K \end{aligned} \right\} \quad (4.12)$$

In the following sections, algorithmic techniques are related to the cost function  $g_0$ , the number of scenarios  $K$  taken into account and the relations among the constraints.

In section 4.2.1, techniques are based on a single scenario  $K = 1$ , i.e. a single outcome  $\omega'$  among the possible realizations of the events. The solution to the problem  $x$  is taken among  $X(\omega')$ .

$$\left. \begin{aligned} Z^* &= \min (g_0(x, \omega')) \\ s.t. & \ g_i(x, \omega') \leq 0, \ i = 1, \dots, \ m \\ x &\in X(\omega') \subset \mathbf{R}^n \end{aligned} \right\} \quad (4.13)$$

In section 4.2.2, some techniques are based on generating several scenarios  $K > 1$  where the constraints are not linked by the scenario variables  $\omega_k$ , but only by  $x$ . Therefore, some techniques rely on the decomposition of the problem into each scenario  $k$ . So, if each scenario is solved independently, the set of feasible solutions may be restricted to a subset  $X(\omega_1, \dots, \omega_K) \subseteq X$  as a consequence of the selection of scenarios. Actually, a solution  $x_k$  is found for each scenario  $k$ :

$$\left. \begin{aligned} Z_k^* &= \min (g_0(x_k, \omega_k)) \\ s.t. & \ g_i(x_k, \omega_k) \leq 0, \ i = 1, \dots, \ m \\ x_k &\in X(\omega_k) \subset \mathbf{R}^n \\ \forall k &\in K \end{aligned} \right\} \quad (4.14)$$

Afterwards, the cost function  $g_0$  should select the solution of a particular scenario  $x \in X(\omega_1, \dots, \omega_k, \dots, \omega_K)$  or a solution  $x$  generated after processing  $(x_1, \dots, x_k, \dots, x_K)$  in a specific procedure.

Another approach enforces the uniqueness of the decision  $x$  in all scenarios and does not decompose the problem (as in 4.14) by imposing equalities among  $x_k$  variables. These equalities are called non-anticipativity constraints and are detailed in section 4.2.2.6. Typically, they enforce equality constraints like  $x = x_k, \forall k = 1, \dots, K$  as well as other equalities among scenarios showing some similarities. In subsequent sections, we return in more detail to some of these techniques.

## 4.2 Scenario-based techniques

The scenario-based, also called sampling, strategy for stochastic optimization problem is based on the insight that one or several scenarios tackled in a particular way, can provide a decision that will lead to an interesting policy on the long run. If there is an algorithm to solve the deterministic multi-period problem, this algorithm can be employed again for any other single scenario which is a deterministic outcome of the stochastic parameters. These single scenario optimization techniques are depicted in section 4.2.1. Using the same optimization technique, it can also be interesting to solve sequentially a set of individual scenarios. Yet, this requires a meta-framework to select the best decision for the stochastic problem from the set of optimal solutions for each deterministic scenario. This selection procedure is often linked to a pricing technique that tries to estimate the expected value of the solution selected or build from the set of scenario solutions. This is presented in section 4.2.2. If several scenarios are treated simultaneously in parallel, a specific solution technique should be developed (see section 4.2.2.6).

### 4.2.1 Mono-scenario approximation

The mono-scenario approximation has some advantages:

- it is easy to explain to practitioners and to implement,
- usually, it is easier to solve than a stochastic model, as it reduces to a single deterministic optimization,
- the solution speed might be high and practicable,
- the data management is reduced.

Nevertheless, the drawback is "how to find a representative scenario" if such a scenario exists. One can not be sure that a scenario leading to the optimal solution of the stochastic problem exists. Moreover, this scenario might be extremely dependent on the problem structure and difficult to bring to light. So, there are some classical scenarios that should be tested for any problem and afterwards, each problem requires customized attempts according to its structure.

#### 4.2.1.1 Expected value scenario

The expected value solution (EVS) is the optimal policy based on the scenario wherein stochastic parameters are set equal to their expected value [Birge & Louveaux, 1997]. Its value is denoted by EVS. This expected value solution scenario is supposed to be the most representative scenario. But does it lead to the most

interesting decision for the stochastic problem? Yet, when the discrete parameter can only take a small number of values such as in a Bernoulli distribution where the stochastic parameter  $p \in [0, 1]$ , the expected value equals  $p$ . Depending on the structure of the problem, in some cases, it is not possible to build a real scenario out of this non-integer value. Therefore, a choice could be to set this stochastic parameter equal to 0 or 1 if  $p$  is respectively  $< 0.5$  or  $\geq 0.5$  which is the modal value scenario. This induces a Manichean, black and white or 0/1 scenario for the expected value scenario. This scenario taking extreme values might not be close at all to a scenario tempting to be representative of all others, mostly if  $p \cong 0.5$ . In this particular case of discrete probability distribution defined over a small finite set, we propose a new expected value scenario definition that sets all variables to 1 and insert a factor equal to the parameter of the probability distribution in front of the cost or reward of these variables.

Yet, the aim of using this expected value scenario is to generate a scenario that leads to a "representative" scenario. Consequently, a new vision of the expected value scenario is presented within this thesis when the stochastic parameter is drawn from a Bernoulli distribution. This scenario assumes that the stochastic event always occurs ( $x = 1$ ), but the reward or the cost of this event is affected by  $p$ . So, this scenario can be built, while assuming that the stochastic event always occurs, but with an average value.

Surprisingly, the expected value scenario is not a scenario that might realize effectively according to the setting of the original problem. This expected value scenario is a virtual or unrealistic scenario leading to a deterministic model enforcing a decision that might be valid for the stochastic problem. This is not always the case. Yet, in some cases, the optimal decision for the stochastic problem  $E^*$  is encountered by the EVS. So, a complex algorithmic strategy might not be useful.

As introduced in section 3.9, the simulation gives an estimate of the policy value  $\mu_{\pi(i)}$ . Similarly, the estimate of the expected value solution is labeled (EEVS).

#### 4.2.1.2 Modal value scenario

Another classical scenario is the modal value scenario wherein stochastic parameters are set equal to their modal value. It is the likeliest scenario and it should be drawn more often than others in a scenario generation process simulating representative outcomes, as in a Monte-Carlo method. Obviously, one should check that the modal scenario is not equal to the expected value scenario, such as when the stochastic parameters are issued from a normal probability distribution or any symmetric distribution.

As examples of modal scenarios, applications in chapters 6 and 7 show an interesting behavior. In the first application, the distribution law represents the probability of a load to be released in a time-window. In the particular case of a uniform distribution over the periods of the TW, it is not possible to differentiate among periods as they are all modal. So, defining the modal scenario is not always obvious and several ones might exist.

For the second application, the probability distribution is represented by a Bernoulli distribution where the stochastic parameter is  $p_i \in [0, 1]$  representing the probability of realization of a transportation order  $i$ . This leads to a radical modal value scenario when  $p_i \in [0, 0.5[$  or  $p_i \in [0.5, 1]$ .

In both applications, the modal value scenarios under-perform, but as this kind of scenario is easy to set-up, it should be attempted. In some models, this modal scenario might represent the worst case scenario. In these particular cases, the solution of the model generated by the modal value scenario leads to robust optimization and might not reflect the expected value objective.

#### 4.2.1.3 Optimistic value scenario

An optimistic value scenario is not directly induced by the probability distribution, but by the nature of the stochastic parameter. It depends on the problem description and represents a particular extreme case which might be favorable, because all stochastic parameters values are correlated. In the thesis applications, dealing with stochastic release dates and stochastic availability of a transportation order, the optimistic scenarios are respectively, a scenario assuming the earliest release dates for all loads or a case where all projected orders are assumed to be confirmed. The policy issued from the optimization of the first model is efficient while the policy based on the second one under-performs.

#### 4.2.1.4 Customized scenario

Depending on the structure of the model, other single scenarios can be investigated. These scenarios engender heuristic solutions to the stochastic problem which might turn out to be good.

#### 4.2.1.5 Bounds and mono-scenario approximation

Let us assume that algorithm  $\mathcal{A}_1$  outclasses algorithm  $\mathcal{A}_2$  if it is significantly better (details in chapter 5). Conversely, algorithm  $\mathcal{A}_1$  does not outclass algorithm  $\mathcal{A}_2$ , if it is not significantly better. If a policy value  $\mu_{\pi(i)}$  issued from a mono-scenario

approximation is never outclassed by  $O_H^*$ , then this optimization technique is sufficient for the problem. This means that the policy issued from the optimization of a model based on a deterministic outcome of the stochastic information is equivalent to the policy issued from the a-posteriori fully revealed information over the rolling horizon. This means that the solution of a stochastic problem can be approximated easily by the optimization of a deterministic one. This situation jeopardizes the initial assumption that stochastic problems are hard to solve. Indeed, this does not prove that this solution issued from an approximate model is theoretically optimal for the real problem, but statistically equivalent or satisfactory in practice. Therefore, mono-scenario approximation might be relevant while being an easy to implement technique. If not, multiple scenario approaches might be developed and the same test on their relative performances to the bounds can be performed.

#### 4.2.2 Multiple Scenario Approaches (MSA)

A drawback of the mono-scenario approximation resides in the fact that the solution issued for a single deterministic scenario model should suit a wide range of scenarios for the stochastic problem. At first sight, it seems more relevant to explore several outcomes of the stochastic parameters to analyze which solution might arise among others. But in this case, there must be a selection among these optimal solutions or an additional procedure to generate a single decision from several solutions (see [Van Hentenryck & Bent, 2006]). The multiple scenario approaches are multi-step optimization techniques. Depending on the practicability of optimization techniques, a set of scenarios can be solved one by one or simultaneously.

- if several scenarios are solved sequentially:
  1. Scenario generation (see section 4.2.2.1)
  2. Sequential mono-scenario approximations similar to several deterministic multi-period problem optimization (see chapter 3)
  3. Decision selection or generation from a collection of solutions (see sections 4.2.2.2 up to 4.2.2.5)
- if several scenarios are solved simultaneously:
  1. Scenario generation (see section 4.2.2.1)
  2. Unique optimization of a set of scenarios (see section 4.2.2.6)

#### 4.2.2.1 Scenario generation

As the optimization process is supported by a subset of scenarios taken among all possible ones, either this subset should be representative of the global set or it should contain specific scenarios leading to promising solutions. Finding promising solutions assumes that there is some insight on the nature of the stochastic solution. So, it is not obvious according to the structure of the problem to find specific scenarios leading to promising solutions for a selection or aggregation technique. In this thesis, such structural scenario generation process has not been used and only representative subsets are drawn.

However, several ways are proposed to generate a representative subset of scenarios. A usual one is the Monte-Carlo generation method which is based on a random generation process using the probability distributions of the stochastic parameters. The probability to draw a scenario with a stochastic parameter equal to a specific value is equal to the probability of having this parameter value from its probability distribution. If the number of scenarios in the subset becomes larger, the Monte-Carlo generation process tends to reproduce the whole set of scenarios and the relative occurrence of each scenario in the subset tends to its probability. Unfortunately, this convergence is not ensured for small subsets. An analysis on the policy performances in accordance with the number of calibrating scenarios  $K$  can be performed to check the sensitivity of the results to the size of this parameter (see section 7.7.2).

To alleviate some disturbances in the drawing of scenarios, a stratified generation process, or others techniques can be used (see [Mitra, 2006]). An important criterion to validate the scenario generation process is the stability of the solution. This means that the policy performance should not be affected by the outcomes of the scenario generation process.

Another theoretical opportunity might be to use a subset of scenarios that does not fit the probability distribution, but that would lead to an interesting decision in practice, (like the single scenario approximation). However, to the best of the author's knowledge, the description of such attempts and positive results have not been provided yet for multiple scenarios.

In the framework that we discuss here, scenarios are used to *generate* a solution and not to *evaluate* or to *validate* the expected policy performances. So, they are called "*Calibrating scenarios*". Therefore, the statistical validation of the performance should be done on an independently generated set of "*Test scenarios*" representing also outcomes of the stochastic parameter (see chapter 5). Finally, generating scenarios and being able to solve the associated single-scenario models does not lead to a unique optimal solution. Therefore, several algorithms dealing with



scenario sampling as an optimization technique for stochastic problems were introduced and described in [Van Hentenryck & Bent, 2006]. They represent a generic approach, that is a family of algorithms to select or generate a unique solution for the stochastic problem from several optimal solutions for deterministic scenarios.

These multiple scenario approaches are described in the following sections:

- Consensus algorithm in section 4.2.2.2
- Restricted Expectation algorithm in section 4.2.2.3
- Regret algorithm in section 4.2.2.4
- Scenario hedging and branch-and-regret algorithms in section 4.2.2.5

For the Consensus algorithm, parameters have to be customized according to the problem while some others are not problem specific such as the Restricted Expectation algorithm. Besides this, an optimization technique to solve simultaneously several scenarios, the Subtree algorithm, is exposed in section 4.2.2.6.

Note that in our applications, we have only used the Consensus, Restricted Expectation and Subtree algorithms.

#### 4.2.2.2 Consensus algorithm

According to authors in [Van Hentenryck & Bent, 2006], the Consensus algorithm  $C_s$  is based on the simple idea that an action which is common to the optimal solutions of several deterministic scenarios should be efficient in general for the stochastic problem. In a caricatural way, frequently chosen actions should be effective actions. Notice that actions are considered and not decision variables from which they are issued.

For example, in the first application (see chapter 6), the decision variables are binary and represent the decision to send a particular *set of loads*. But, the frequent actions taken into account to generate the consensus actions are based on the decisions to send *a particular load* in the current period whatever the set in which this load is included.

In the second application (see chapter 7), several decision variables might imply for a truck to wait on site. The occurrence of "wait" actions is taken into account in the Consensus algorithm wherever these decisions are issued from.

A generic description of the Consensus algorithm at period  $t$  is as follows:

1. Generate  $K$  calibrating scenarios ( $K$  is an algorithmic parameter).

2. For each  $k = 1, \dots, K$ , solve sequentially the deterministic model associated with scenario  $k$ .
3. For each scenario  $k$ , extract from the solution the set of actions to be performed in period  $t$ , say  $\theta^k$ . Keep track of the number of times, say  $a_i^k$ , that action  $i$  occurs in  $\theta^k$ , where  $i$  ranges over the set of possible actions and  $k = 1, \dots, K$ .
4. Count the total number of occurrences of actions  $i$  over all scenarios, say  $\Theta_i = \sum_{k=1}^K a_i^k$ .
5. Sort the values  $\Theta_i$  by decreasing value and select the actions to be performed, in this order, according to a criterion or a procedure which allows to rebuild a solution. For instance, select action  $i$  if it appears in a majority of scenarios, that is, if  $\Theta_i/K \geq 0.5$ , as long as this is feasible.
6. Implement this consensus solution in period  $t$  and roll over to the next period.

Hence, the Consensus algorithm requires  $K$  mono-scenario optimizations and a counting-aggregation procedure which is customized according to the problem.

A drawback of this algorithm is that the frequency of appearance of an action in the optimal solution of mono-scenario models does not ensure its global relevance for the stochastic problem.

More generally, a weakness of  $Cs$  relies in the fact that it includes no pricing in the solution process. Indeed, the compromise is based on the insight that solutions are more or less similar. This might not be true for some problems with huge differences between solutions and actions consequences. Therefore, customizing the procedure to select the actions is the real issue of this family of algorithms.

#### 4.2.2.3 Restricted Expectation algorithm

Conversely to  $Cs$ , which is mostly based on building a solution from parts of optimal ones, the Restricted Expectation algorithm  $RE^*$  is based on selecting a complete solution among optimal ones.  $RE^*$  stems from the Expectation algorithm described in [Van Hentenryck & Bent, 2006]. In the Expectation algorithm, the procedure is simple and exact, but it is exhaustive as it is an enumerative technique. Indeed, the set of all feasible solutions  $x \in X$  is tested over the whole set of scenarios  $\omega \in \Omega$ . Then, the solution with the best cumulated value,  $E^*$ , is selected on this basis. Theoretically, it is an exact method as the expected value of every solution is calculated. Nevertheless, it is often intractable as already stated in [Van Hentenryck & Bent, 2006], because the set of all feasible solutions might be large and unknown

explicitly, and moreover because the number of scenarios might be huge (or even infinite). Similarly to other methods, the expectation algorithm suffers from the curse of dimensionality. Consequently, the Restricted Expectation algorithm is an approximation of this exact method and we introduce this term "Restricted" in the application of chapter 6 to differentiate the approximate algorithm from the exact one. Actually, this expression "restricted" is owing to a double restriction, regarding the number of solutions tested  $|X|$  and the number of scenarios for valuation  $|\Omega|$ .

The algorithm is based on cross-evaluation of the optimal solution for each calibrating scenario over the same set of calibrating scenarios. Moreover, in our setting, the relative weight of scenarios is not taken into account while cumulating each solution values during the cross-evaluation over scenarios.

A generic version of the Restricted Expectation algorithm works as follows at decision period  $t$ :

1. Generate  $K$  calibrating scenarios.
2. For each  $k = 1, \dots, K$ , solve sequentially the deterministic model associated with scenario  $k$ . Let  $v_k$  denote the optimal value of this model.
3. For each scenario  $k$ , extract from the solution the set of actions to be performed in period  $t$ , say  $\theta^k$ .
4. For each  $k = 1, \dots, K$  and for each  $j = 1, \dots, K$  with  $j \neq k$ , successively:
  - Implement the set of actions  $\theta^k$  in period  $t$  for the calibrating scenario  $j$ .
  - Roll over to period  $t + 1$  and solve the deterministic model associated with this modified scenario  $j$ . Denote by  $v(\theta^k, j)$  the optimal value of this model.
5. For each  $k$ , compute the restricted expected value of the set of actions  $\theta^k$  over all scenarios, defined as  $\Theta^k = v_k + \sum_{j=1, j \neq k}^K v(\theta^k, j)$ .
6. Select the set of actions  $\theta^* = \theta^k$  which provides the best value of  $\Theta^k$ .
7. Implement the set of actions  $\theta^*$  in period  $t$  and roll over to the next period.

Even if this method tends to mimic the exact method, the Restricted Expectation algorithm has also several drawbacks. The set of solutions considered for the cross-evaluations are issued from the optimization of deterministic mono-scenario model. Therefore, the stochastic solution  $E^*$  might not be among the candidates and  $\theta^*$  might be very different from  $E^*$ . Moreover,  $K^2$  mono-scenario model optimization

processes are required. So, the size of the set of calibrating scenarios,  $K$ , might be repellent if every optimization step requires a long computation time. A sensitivity analysis on the number of calibrating scenarios is recommended to keep  $K$  as low as possible in order to reduce the computation time.

#### 4.2.2.4 Regret algorithm

According to the authors in [Van Hentenryck & Bent, 2006], the Regret algorithm is close to the Restricted Expectation algorithm, but it attempts to avoid the drawback of  $K^2$  mono-scenario models optimizations. According to the structure of some problems, when the optimal solution for a scenario  $\theta^k$  is calculated ( $k \in 1, \dots, K$ ), it might be possible to approximate the value of this solution when  $\theta^k$  is implemented in another scenario  $j$ . The value of this approximate solution is labeled  $v(\tilde{\theta}^k, j)$  with  $j \in 1, \dots, K$  and  $j \neq k$ . This is called the suboptimal approximation. Instead of re-optimizing when the solution of scenario  $k$  is implemented in scenario  $j$ , as in  $RE^*$ , the Regret algorithm approximates the worst or the average cost of solution  $\theta^k$  in scenario  $j$ , by taking into account the differences between scenario  $k$  and scenario  $j$ . Initially,  $K$  mono-scenario model optimizations are required for the set of calibrating scenarios. Then,  $K * (K - 1)$  efficient approximation procedures are added.

A generic version of the Regret algorithm works as follows at period  $t$ :

1. Generate  $K$  calibrating scenarios.
2. For each  $k = 1, \dots, K$ , solve sequentially the deterministic model associated with scenario  $k$ . Let  $v_k$  denote the optimal value of this model.
3. For each scenario  $k$ , extract from the solution the set of actions to be performed in period  $t$ , say  $\theta^k$ .
4. For each  $k = 1, \dots, K$  and for each  $j = 1, \dots, K$  with  $j \neq k$ , use  $\theta^k$ , and the differences between scenarios  $k$  and  $j$ , and a suboptimal approximation procedure to generate the value  $v(\tilde{\theta}^k, j)$  as an estimate of the value  $v(\theta^k, j)$ .
5. For each  $k$ , compute the value of the set of actions  $\theta^k$  over all scenarios, defined as  $\Theta^k = v_k + \sum_{j=1, j \neq k}^K v(\tilde{\theta}^k, j)$ .
6. Select the set of actions  $\theta^* = \theta^k$  which provides the best value of  $\Theta^k$ .
7. Implement the set of actions  $\theta^*$  in period  $t$  and roll over to the next period.

This algorithm works faster, but requires a quality warranty on the suboptimal approximation gap. Conversely to  $Cs$  and  $RE^*$ , this process is not obvious to implement even in a simple way as finding the suboptimal approximation algorithm is a problem itself.

#### 4.2.2.5 Scenario hedging and branch-and-regret algorithms

Previous techniques assume that a scenario representing a deterministic outcome of the stochastic parameters can be solved over the rolling horizon  $H$ . In some cases, finding the optimal solution for a single period is already a hard task to fulfill. Therefore, some algorithms based on a decomposition procedure were developed to connect the  $1 + H$  optimal solutions of each period of the rolling horizon and the decision period  $t$ . This means that in a first step, the rolling horizon and the decision period are split into  $1 + H$  periods and then solved independently per period. In a second step, these algorithms connect the solutions between successive periods to generate a coherent solution over all periods of the rolling horizon and the decision period. These are called scenario hedging (SH) algorithms or heuristics. They were first introduced in [Rockafellar & Wets, 1991]. A dynamic and stochastic version (DSSH) for a vehicle routing problem is presented in [Hvattum et al., 2006].

Fortunately in this thesis, the formulations of both deterministic multi-period problems used to solve a scenario-based model are timeless and solvable for all periods at once over the rolling horizon. Therefore, this algorithmic technique is not used.

Another interesting algorithm is the branch-and-regret heuristic (BRH) which is a mix between the Consensus and the Restricted Expectation algorithms (see [Hvattum et al., 2007] and [Tirado et al., 2013]). This algorithm also mimics a branching scheme.

As previously, a set of calibrating scenarios is solved. Then, equal actions among the solution of the mono-scenario model optimization are counted such as in the Consensus algorithm. Next, as in the Restricted Expectation algorithm, the most frequent action is fixed in all mono-scenario models and the cumulated value of the optimal solutions over these mono-scenario models is estimated. Reversely in a second step, the opposite action is fixed and the cumulated value for non-performing that action in the mono-scenario models is computed. Consequently, the expected values for performing or not that action is known. If it is more profitable, the most interesting action is applied or discarded. This choice is definitively fixed. The procedure is repeated in a branching fashion for a second action and similarly until a convergent complete set of actions is selected. One can check that this procedure

is computationally highly demanding. However, one can check that this heuristic includes both the sampling and the pricing issues.

As this thesis focuses on simple techniques and as they perform efficiently in practice, the heuristics from the present section are just mentioned for theoretical purposes and the completeness of the survey.

#### **4.2.2.6 Deterministic equivalent: scenario tree**

Contrary to the Expectation algorithm that performs serially on scenarios and solutions, a technique that works with all scenarios in parallel and on a unique decision can be set up. This exact technique solves the stochastic problem at once by generating all scenarios simultaneously while forcing common actions to be taken for scenarios  $k$  and  $j$  ( $k, j = 1, 2, \dots, K$ ) until both scenarios diverge. To diverge means that at a given period, two scenarios do not contain the same data. So, the solutions  $\theta_k$  for scenario  $k$  and  $\theta_j$  for scenario  $j$  must be identical up to that period. For the remaining periods, decisions for the solution of scenario  $k$  and  $j$  might not be related to each other.

This deterministic equivalent representation is called the scenario tree. It includes all possible combinations of the outcomes of the stochastic parameters according to their probability distributions. In case of continuous probability distributions and of some discrete distributions, the number of scenarios is infinite, so the probability laws should be approximated in order to generate the tree. In this thesis, only problems dealing with discrete distributions defined on a small set are investigated. Therefore, theoretically, the fully equivalent deterministic scenario tree can be generated.

To enforce similar actions for scenarios which share the same initial events, non-anticipativity constraints, as equalities, should be added on common decision variables. This makes the decisions coherent up to the period wherein the differences of the outcomes of the stochastic parameters occur.

Owing to the cumbersome number of scenarios to be considered, for computational and often computer memory reasons, it is not possible in practice to generate and to solve the full tree model. Therefore, a heuristic leading to an approximation by the optimization on subtree model including a reduced subset of scenarios has to be considered. In that process, the relative weights of the scenarios can be removed, leading to another approximation. Actually, the ratio of the occurrence of a stochastic event over the number of scenarios in the subtree should converge to the probability distribution of this event.

Besides this, in a multi-period problem where re-optimization takes place at each decision period, the non-anticipativity constraints could be only inserted among

variables of the current action period and not the entire rolling horizon. Indeed, at each subsequent decision period, the optimization process might change previously optimal decisions over the rolling horizon according to the revealed information. Therefore, if the actions taken can be changed, imposing non-anticipativity constraints in the remote part of the rolling horizon would not represent the flexibility of the decision process along time. Besides this, equalities are only imposed on variables leading to common actions even if they are issued in each scenario from variables representing different decisions (see chapter 7 for an application).

The Subtree algorithm *ST* performs as follows at period  $t$ :

1. Generate  $K$  calibrating scenarios of the subtree,
2. Insert non-anticipativity constraints on variables leading to common actions in period  $t$ ,
3. Solve the subtree model using a dedicated algorithm,
4. Extract the global actions from the decision variables,
5. Implement the solution in period  $t$  and roll over to the next period.

This heuristic is somewhat similar to the branch-and-regret algorithm, while performing the whole computation in a single run. Indeed, it is also a sampling and pricing technique. In fact, the subtree approach is appealing as it tends to avoid previous algorithmic drawbacks and to represent the equivalent deterministic model of the stochastic one.

However instead of estimating the cumulated cost over each scenario when a decision is fixed, this is made simultaneously over all scenarios of the subtree. Therefore, the subtree model optimization is usually computationally demanding. Moreover and unfortunately, even if a single deterministic scenario can be solved efficiently, the non-anticipativity constraints might destroy the structure of the problem when all scenarios are considered simultaneously in the formulation. This might lead to an intractable problem, not only because of the increase in size, but because of the new formulation characteristics.

In [Escudero et al., 2012], an algorithmic framework to solve large-scale multi-stage stochastic mixed 0-1 problems with non-symmetric scenario trees is exposed. This shows that some breakthroughs might arise in the future in scenario tree optimization. For example, in the second application in chapter 7, the subtree formulation can be solved easily. Yet, subtree formulation remains usually hard to optimize, such as in the application of chapter 6.

### 4.3 Bounds and expected value solution

As a first step, our methodology uses bounds issued from the optimizations of specific deterministic scenario models related by equations 3.7 and 3.8:

1. myopic policy value  $O_{RH}^*$ ,
2. value of the policy with fully revealed information over the rolling horizon  $O_H^*$ ,
3. value of the policy with fully revealed information  $O_T^*$ .

They are deterministic bounds assuming fully revealed information over some parts of the horizon. In fact, the optimal decision for the stochastic problem over the whole horizon  $E^*$  might not be equal to the optimal decision for the deterministic one over the whole horizon  $O_T^*$  or over any rolling horizon  $O_H^*$ . Indeed, solutions of the deterministic problem with the fully revealed information  $O_H^*$  or  $O_T^*$  are often unreachable bounds as the stochastic problem with unrevealed information is by nature different from the deterministic one.

The expected value solution (EVS) (or its estimate EEVS) is also considered as a bound, but issued from a particular outcome of the stochastic information and not the fully revealed deterministic one. The EEVS can be introduced in equations 3.7 and 3.8, between the the myopic policy value  $O_{RH}^*$  and  $\mu_{\pi^*}$ .

- for a maximization problem:

$$O_{RH}^* \lesssim EEVS \lesssim \mu_{\pi^*} \lesssim O_H^* \leq O_T^* \quad (4.15)$$

- for a minimization problem:

$$O_{RH}^* \gtrsim EEVS \gtrsim \mu_{\pi^*} \gtrsim O_H^* \geq O_T^* \quad (4.16)$$

Similarly to the explanation from section 4.2.1.5, if it is not possible to assess that  $O_H^*$  is significantly better than the EEVS (see chapter 5), developing a complex algorithms such as in sections 4.2.2 and 4.2.2.6 is useless. So, comparing the EEVS and  $O_H^*$  allows to stop further algorithmic developments. One can state that the stochastic solution approximated by a mono-scenario optimization using the expected value scenario performs as well the fully revealed information optimization over the rolling horizon.

If  $O_H^*$  is significantly better than the EEVS, the maximal gap for developing a dedicated algorithm using the stochastic information in a better way can be estimated. Even if the EVSS is outclassed by  $O_H^*$ , this does not prove that it is outclassed by the optimal solution of the stochastic problem  $E^*$ .



Afterwards, the best policy  $\pi^*$  will be selected among others. If the best policy value  $\mu_{\pi^*}$  is not significantly better than EEVS, for the sake of simplicity of the algorithmic procedure, the computing speed and the data management, it might be profitable to use the expected value scenario optimization instead of a more complex algorithm. The loss due to this choice can be estimated.

Theoretically, the gap between the optimal solution of the stochastic problem  $E^*$  and the EVS gives the Expected Value of the Stochastic Solution, labeled EVSS. **EVSS**: Expected value of the stochastic solution

$$EVSS = E^* - EVS \quad (4.17)$$

As, these values are approximated by the optimization technique ( $\mu_{\pi^*} \cong E^*$ ) and estimated statistically, the EVSS is valued by the following approximation:

$$EVSS \cong \mu_{\pi^*} - EEVS \quad (4.18)$$

This gives a decision criterion on the value for developing a dedicated algorithm for the stochastic problem instead of using one of its simplest deterministic approximation. Similarly to figure 3.9, the EEVS and the EVSS can be reported in a diagram for visualization. For a maximization problem, this leads to :

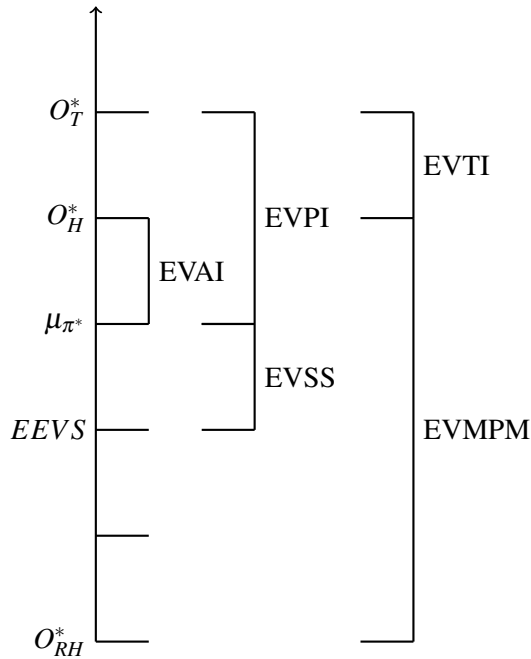


Figure 4.1: Bounds and values of information with expected value scenario

## Chapter 5

# Statistical validation

### 5.1 Introduction

Testing algorithmic performances is not obvious even for mono-period deterministic problems. Even when working with exact algorithms, one might consider the computing time or the number of iterations as relevant performance criteria. Beside this, the computer speed, the available memory, the language, the data management and others features might also be taken into account. When the algorithm is a heuristic or a meta-heuristic, discussions might be diverted to the optimality gap, the rate or speed of convergence, the stopping criterion or the approximation ratio. Moreover, algorithms are often compared on batches of instances where results are presented as averages and percentages (see [Hooker, 1995], [Barr et al., 1995], [Johnson, 2001], [Rardin & Uzsoy, 2001], [Eiben & Jelasity, 2002], [Brownlee, 2007] and [Ruml, 2010]). But these comparisons might lead to opposite conclusions.

There are two additional difficulties in the present work, which deals with multi-period stochastic problems. Owing to the multi-period nature of the problems, the policy value over the horizon  $T$  is an estimate of its expected value on the long-run. Due to its stochastic nature, as mentioned in chapter 4, algorithmic techniques are approximative and often include a random component. Actually, the set of calibrating scenarios  $K$  that are used, for example, in subtree or in restricted expectation algorithms, only represents a subset of all potential scenarios (see chapter 4). Moreover, for related reasons linked to the number of scenarios that might arise along the whole horizon, it is practically not possible in the evaluation phase to test algorithms over all potential scenarios. So, estimates of the policy values can only be evaluated over a reduced sample of so-called *test scenarios* (or *evaluation scenarios*) which are independent of, and usually different from, the calibrating scenarios used by the algorithms in order to generate solutions. These test scenarios represent outcomes

of the stochastic parameters over the whole horizon  $T$ , whereas calibrating scenarios of algorithms range over the rolling horizon  $H$  at each decision step  $t$ .

All these reasons explain that, in computational experiments, policies can only be compared on a statistical basis. In order to reduce the bias due to the generation of scenarios, it is preferable to use the same sample of test scenarios when comparing a pair of algorithms. Doing so removes the variance induced by different samples. Practically, the test scenarios are recorded and the performance of each algorithm is evaluated successively on the same scenarios. In this way, the relative performance over each test scenario can be compared on this reduced subset, rather than just comparing values averaged over distinct sets of scenarios which would require usually to be larger for the same confidence level. Note that, the probability distributions are independent from the decision taken previously, meaning that the scenarios do not evolve according to the algorithmic procedure.

Recording the test scenarios is also important for the methodology as policies are compared with the bounds derived when the information is fully revealed:

- the bound with the fully revealed information over the deterministic part of the rolling horizon  $O_{RH}^*$ , i.e. the myopic bound,
- the bound with the fully revealed information over the rolling horizon  $O_H^*$ , i.e. the a-posteriori bound over the rolling horizon,
- the bound with the fully revealed information over the whole horizon  $O_T^*$ , i.e. the a-posteriori bound over the whole horizon.

When the performance of algorithms has been measured and the results are available, there are several ways to present them. In fact, several methods have been used in the literature to assess the quality of different policies for stochastic problems. Among others, one can notice :

1. Value of information and value of information under heuristic cited in [Mitrović Minić et al., 2004]
2. Means and ratios cited in [Regan et al., 1998]
3. Competitive analysis and Approximation ratio cited in [Angelelli et al., 2007]
4. Percentage to the a-posteriori solution or bounds cited in [Spivey & Powell, 2004]
5. Mean and variance trade-offs cited in [List et al., 2003]
6. Statistical pairwise comparison test cited in [Tjokroamidjojo et al., 2006]

These tests are described in the following sections.

### 5.1.1 Value of information

In [Mitrović Minić et al., 2004], similarly to [Birge & Louveaux, 1997], the authors evaluate the performance of algorithms for dynamic problems. They measure performance as a ratio taking into account the optimal solution of the dynamic problem  $\dot{x}^*$  and the optimal solution of the static problem  $x^*$ . Using the concepts introduced earlier in this thesis, this can be translated into comparing the bounds  $O_H^*$  and  $O_T^*$ . Accordingly, the *Value of information* ratio can be defined as follows:

$$V = \frac{x^* - \dot{x}^*}{\dot{x}^*} = \frac{O_T^* - O_H^*}{O_H^*}. \quad (5.1)$$

For this application, as all requests have to be served, the ratio  $V$  measures the extra cost faced for lacking of information. If all requests were known in advance, another feasible solution, the optimal one for the static problem, would have to be set-up. So, owing to the missing information released dynamically along time, another solution has been performed. This ratio is used to measure the ability to deal with missing information of the algorithm generating the solution of the dynamic problem. Interestingly, the authors also define the *value of information under heuristic* when the exact bounds cannot be computed exactly, but are approximated by a heuristic  $\mathcal{H}$ . In such cases,  $\dot{x}^*$  becomes  $\dot{x}^{\mathcal{H}}$  and  $x^*$  becomes  $x^{\mathcal{H}}$ . Consequently, equation 5.1 can be modified to define the value of information under heuristic:

$$V_{(\mathcal{H})} = \frac{x^{\mathcal{H}} - \dot{x}^{\mathcal{H}}}{\dot{x}^{\mathcal{H}}} \quad (5.2)$$

As explained in chapter 3, this value of information is a good decision tool for the manager to decide whether or not changing the process might be profitable. In the present definition, the ratio  $V$  measures the cost of the rolling horizon length (EVTI). But, in other problems, the value of information is related to the stochasticity (EVPI) such as when equation 3.9 is presented as a ratio.

$$V = \frac{O_T^* - \mu\pi^*}{\mu\pi^*} \quad (5.3)$$

Yet, in general, this value of the information is not a tool to compare two algorithmic policies in a multi-period stochastic problem. Therefore, in the presented methodology, these values are used to measure information gaps instead of being used for the comparison of algorithms. One of the contributions of this thesis is to define and to use various values of the information, instead of using  $V$  for all. We stress the fact that these values are only estimated (by sampling) and we accordingly label them in the format *EXXX* (instead of using the terminology "under heuristic").

### 5.1.2 Means and ratios

In [Regan et al., 1998], the authors present a range of performance measures for algorithms developed for its application of dynamic fleet management such as:

1. *"Mean and standard deviation of the average length of loaded and of empty movements"*
2. *Mean and standard deviation of the average wait time for service*
3. *Ratio of time spent empty to time spent loaded*
4. *An estimate of the operating profit generated per vehicle per week*
5. *Fraction of pickup deadlines missed (...)*
6. *Mean and standard deviation of loaded and empty distances traveled across the fleet"*

Even if all these data are interesting, the objective function is valued as a monetary profit including charges, unladen and loaded travel costs. For their specific application, authors display the results as the average empty distance traveled, the average wait service, the standard deviation for the wait service and the profit per vehicle per week. This results in numerical data from several heuristics from which global conclusions cannot be taken regarding absolute performances. Actually, the decision criteria and their weights are not well defined and evaluation units are different. Consequently, the evaluation of the performance becomes itself a decision problem.

### 5.1.3 Competitive analysis and approximation ratio

The approximation ratio of a heuristic  $\mathcal{H}$  on an instance  $I$  is classically defined as the ratio of the value  $v(\mathcal{H}, I)$  returned by the heuristic to the optimal value  $v^*(I)$  of the instance. More precisely, this ratio is guaranteed to be greater than or equal to 1 both for minimization<sup>(1)</sup> and for maximization<sup>(2)</sup> problems if we define it as follows:

$$r_{\mathcal{H}} = \max_I \left( \frac{v(\mathcal{H}, I)}{v^*(I)} \text{ (1)}, \frac{v^*(I)}{v(\mathcal{H}, I)} \text{ (2)} \right). \quad (5.4)$$

In [Angelelli et al., 2007] for a multi-period deterministic problem, the ratio is transformed to provide the best value of the policy  $\mathcal{H}_H$  taking into account partially revealed information over  $H$ , called on-line algorithm by the authors, and the optimal value of the a-posteriori optimization with the fully revealed information over

the whole horizon  $T$ , called off-line algorithm ( $O_T^*$ ).

$$r_{\mathcal{H}} = \max_I \left( \frac{v(\mathcal{H}_H, I)}{O_T^*} \text{ (1)}, \frac{O_T^*}{v(\mathcal{H}_H, I)} \text{ (2)} \right) \quad (5.5)$$

Equation 5.5 generates a ratio instead of a gap for the value of the information as equation 5.1. Yet, this is different as the approximation ratio is often a theoretical result proved for any  $I$  while the value of information can be estimated on average by simulation.

Even, if it gives an interesting indication on the policy performances, this method leads to two drawbacks for assessing algorithmic performances. Firstly, it is not always possible to evaluate theoretically the approximation ratio of an algorithm. Secondly, the average practical results from algorithm  $\mathcal{A}_1$  and  $\mathcal{A}_2$  might differ from their theoretical approximation ratios, based on a worst case performance. Yet according to authors: "*An on-line algorithm  $\mathcal{A}_1$  is optimal if no other algorithm  $\mathcal{A}_2$  has a competitive ratio  $r_{\mathcal{A}_2} < r_{\mathcal{A}_1}$* ". Finally, in theory, some algorithms might be optimal, but due to the curse of dimensionality, they might be unusable. Therefore, the approximation ratio method might not be practicable to compare policies issued from heuristics while using simulation.

#### 5.1.4 Percentage deviation from the bounds

In [Spivey & Powell, 2004], for a maximization problem, the authors set to 100% the a-posteriori optimal value  $O_T^*$  in order to normalize their results. Therefore, all policies values are below 100% and accordingly their relative difference to this bound provide their deviations in percentage to this upper bound. However, the value of the myopic solution is not set to 0%, but is valued in a range between 10 to 60 %. Therefore, the important managerial information that reflects the closing of the gap between the myopic solution and the a-posteriori solution (by 40% to 90%) is not highlighted. Moreover, for instance, if the myopic policy is valued at 10% or 90%, which conclusion can be drawn for a policy estimated in both cases at 91%, so that the deviation in percentage to the upper bound is 9% in both cases. In the first case, this result seems to be a good performance, but in the second one? Therefore, we think that both deviations to the upper and lower bounds are important.

Beside this, results obtained by a myopic algorithm and some heuristics for several instance classes are reported. For each policy, the results are computed as the global mean over 100 tests and they are provided separately for each instance class. Unfortunately, even if the myopic algorithm can be discarded, the performance of the heuristics evolves erratically according to the instance classes. Therefore, just looking at the figures, it is not possible to select a "best performing" heuristic. In

fact, even though the number of tests is huge, there is no statistical evaluation of the significance of the differences between heuristics. As a conclusion, just looking at the figures even reported as percentage or averaged over many instances is not a sufficient way to analyze policy performances.

### **5.1.5 Mean and variance trade-offs**

In [List et al., 2003], the authors analyze the performance of their algorithms in terms of robustness. According to them, the robustness is the ability of the algorithm to avoid extreme consequences on some scenarios, (as discussed in section 5.3). In particular, [List et al., 2003] can accept a policy which is dominated on average, as long as it avoids to cross a threshold value for some scenarios. This leads to the definition of a mean vs. variance trade-offs: that is, if the variance is below a specific value for some algorithms, their means can be compared.

Unfortunately for this method, algorithms cannot be tested on all scenarios and it is not always possible to identify the worst-case scenarios leading to extreme consequences.

Finally, in the case of long-haul transportation, from an economical point of view, managers would often prefer to minimize their average cost, under the assumption that they are usually facing a normal situation and that they will react specifically in case of extreme events which are supposed to be sporadic. Consequently, this method to compare algorithmic performances is not really appropriate for our purpose.

### **5.1.6 Statistical pairwise comparison test**

In [Tjokroamidjojo et al., 2006], the authors present a two-step method to check statistically the performances of their algorithms. The first test is an ANOVA test to show that the results of the policies are not issued from the same distribution. The second test is a Fisher's statistical pairwise comparison test to indicate whether or not the difference between policies is statistically significant. These tests are interesting to assess the significance of the statistical difference among policy performances.

Moreover, as a contribution from the present work, these tests can be useful to validate the performances of the policy values compared to the estimates of the bounds.

### 5.1.7 Conclusion: Test integration

There are several tests used in the literature to measure the policy performances in dynamic problems. They correspond to different points of view and they answer various questions about policy performances from theoretical worst cases (approximation ratio) through practical worst case, mean and variance trade-offs up to the value of information, the percentage deviation to the a-posteriori solution or statistical validation. In the present case, the aim is to propose a statistical validation method that can take advantage from the previous attempts. Therefore, the methodology presents the policy performances in a blended framework.

- The gap between the estimates of the bounds ( $O_{RH}^*$  and  $O_H^*$ ) is normalized to 100% providing the VMPM, as an opportunity gap (see section 3.9). Therefore,  $O_{RH}^* = 0\%$  and  $O_H^* = 100\%$ .
- Consequently, the fully revealed information policy value  $O_T^*$  is set larger than 100%. Then, various values of the information can be estimated as percentage.
- Policy values using deterministic or stochastic information are reported as percentage between the bounds in order to normalize the results of the algorithms.
- Policies are compared statistically according to a two-step method (see sections 5.2.1 and 5.2.2)
- Robustness is analyzed in a specific way (see section 5.3)

This leads to an understandable presentation framework for managers with bounds, values of information and a statistical validation of policy performances.

## 5.2 Statistical validation: a two-step method

The statistical test used to evaluate policy performances is based on the statistical significance of the difference between two policies or between a policy and a bound. It is a two-step process. Actually, several assumptions should be checked before using the Z-test explained in section 5.2.2 to assess that the difference between two values is statistically relevant (see [Rakotomalala, 2013]):

1. the sample should be large enough  $n \geq 30$ ,
2. tests should be performed over paired samples to reduce the variance due to some random effect in the scenario generation process,



3. the distribution should follow a normal probability distribution.

Complying to the first two requirements is easy. From an instance, 30 test scenarios (see section 5.1) are generated in order to be submitted to all policies and also to estimate the bounds.

The third assumption is treated into the first step of the statistical validation method. It is a normality check of the result distribution which is an hypothesis for the second step consisting in the Z-test.

### 5.2.1 Normality check

There are several tests available to assess that a set of data is not issued from a non-normal distribution (see [Rakotomalala, 2011]). In practice, this double negation leads to use the test as a normality check, but this is not exactly the case. Indeed, the normality can not be checked, but the non-normality characteristic of a distribution can be rejected.

- Graphical tests: histogram, box-plots, QQ-Plot,
- Statistical tests: Shapiro-Wilk, Anderson-Darling, D'Agostino,  $X^2$ , Jarque-Bera, Lilliefors, Kolgomorov-Smirnov, Bontemps Meddahi.

As mentioned in [Razali & Wah, 2011], the Shapiro-Wilk test usually performs better than others and is suitable for a sample size of 30 elements.

This test works as follows:

1. rank the  $n$  observations per increasing value  $y_1 \leq y_2 \leq \dots \leq y_i \leq \dots \leq y_n$
2. evaluate the mean  $\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$
3. evaluate  $T_n$  as  $T_n = \sum_{i=1}^{i=n} (y_i - \bar{y})^2$
4. evaluate the differences  $d_i$  as  $d_i = y_{n-i+1} - y_i$  for  $i = 1 \dots \lceil \frac{n}{2} \rceil$ . If  $n = 2p$  meaning  $n$  is pair, there are  $p$  differences and if  $n = 2p + 1$ , meaning  $n$  is odd, there are also  $p$  differences.
5. evaluate  $W$  defines as  $W = \frac{[\sum_{j=1}^{j=p} (a_j d_j)]}{T_n}$  while coefficient  $a_j$  can be computed or provided by a table (see [Shapiro & Wilk, 1965]).
6. select a critical value  $W_c$  in the Shapiro-Wilk table according to a risk level in percentage (e.g. 1%, 5% or 10%)
7. if  $W > W_c$ , the hypothesis is accepted and the claim that the sample is issued from a non-normal distribution can be rejected. Otherwise if  $W < W_c$ , the hypothesis is rejected and the sample is issued from a non-normal distribution.

## 5.2.2 Statistically significant difference

After checking that the normality assumption is not invalidated by the Shapiro-Wilk test, a  $Z$ -test can be used to differentiate policy performances. Indeed, small gaps between the values obtained for two distinct policies  $\mu_{\pi(1)}$  and  $\mu_{\pi(2)}$  might be related to random effects only. The statistical significance of these differences must be assessed. This method is also useful to validate or not the statistical significance of the difference between a policy value  $\mu_{\pi(i)}$  and the bounds  $O_{RH}^*$ ,  $O_H^*$  or  $O_T^*$ . Therefore in the following formulas,  $\mu_{\pi(2)}$  can be replaced by  $O_x^*$  with  $x \in \{RH, H, T\}$ .

Two types of statistical tests are performed, either two-sided and one-sided:

1. Test A:  $H_0 : \mu_{\pi(1)} = \mu_{\pi(2)}$  vs.  $H_1 : \mu_{\pi(1)} \neq \mu_{\pi(2)}$  to test whether the results provided by two policies are significantly different.
2. Test B:  $H_0 : \mu_{\pi(1)} = \mu_{\pi(2)}$  vs.  $H_1 : \mu_{\pi(1)} > \mu_{\pi(2)}$  (or  $H_1 : \mu_{\pi(1)} < \mu_{\pi(2)}$ ) to test whether the results provided by a policy are significantly better than those provided by another one.

In the applications of this thesis, expressions such as “ $\pi(1)$  outclasses  $\pi(2)$ ” or “ $\pi(1)$  outperforms  $\pi(2)$ ” are used when hypothesis  $H_0$  is rejected in test B. Similarly, we may say that “ $\pi(1)$  not outclassed or not outperformed by  $\pi(2)$ ” when  $H_0$  is not rejected, and so forth.

A classical  $Z$ -test for paired samples can be used to test the hypotheses as follows:

1. denote by  $v(\mu_{\pi(1)}, i)$  and  $v(\mu_{\pi(2)}, i)$ , respectively, the policy values obtained on the  $i$ -th scenario of a paired sample of size  $n$ , for  $i = 1, \dots, n$ ;
2. evaluate the differences  $d_i = v(\mu_{\pi(1)}, i) - v(\mu_{\pi(2)}, i)$  for  $i = 1, \dots, n$ ;
3. evaluate the mean difference  $\bar{d} = \frac{1}{n} \sum_{i=1}^n d_i$ ;
4. evaluate the variance  $s_d^2 = \frac{1}{n} \sum_{i=1}^n (d_i - \bar{d})^2$ ;
5. evaluate the test statistic  $z = \frac{\bar{d}}{s_d / \sqrt{(n-1)}}$
6. fix an uncertainty level  $\alpha$  (e.g., 5% or 1%) and extract the critical value from the tables of  $Z_{(1-\frac{\alpha}{2})}$  for the first test and  $Z_{(1-\alpha)}$  for the second test.
7. for test A, if  $|z| = \frac{|\bar{d}|}{s_d / \sqrt{(n-1)}} > Z_{(1-\frac{\alpha}{2})}$  then at the level of uncertainty  $\alpha$ , the hypothesis  $H_0 : \mu_{\pi(1)} = \mu_{\pi(2)}$  can be rejected and  $H_1 : \mu_{\pi(1)} \neq \mu_{\pi(2)}$  is accepted; conversely, if  $|z| < Z_{(1-\frac{\alpha}{2})}$ , then  $H_0$  cannot be rejected;

8. for test B, if  $|z| = \frac{|\bar{d}|}{s_d/\sqrt{(n-1)}} > Z_{(1-\alpha)}$  then at the level of uncertainty  $\alpha$ , the hypothesis  $H_0 : \mu_{\pi(1)} = \mu_{\pi(2)}$  can be rejected and  $H_1 : \mu_{\pi(1)} > \mu_{\pi(2)}$  or  $H_1 : \mu_{\pi(1)} < \mu_{\pi(2)}$  is accepted; conversely, if  $|z| < Z_{(1-\alpha)}$ , then  $H_0$  cannot be rejected;
9. for test B, when the null hypothesis  $H_0$  is rejected, we conclude:
  - $\mu_{\pi(1)} > \mu_{\pi(2)}$  if  $z > 0$  meaning  $\bar{d} > 0$
  - $\mu_{\pi(1)} < \mu_{\pi(2)}$  if  $z < 0$  meaning  $\bar{d} < 0$ .

The Z-test allows us to draw conclusions on policy outclassment or equivalence with bounds. Practically, in our thesis a 95% confidence level, meaning a 5% uncertainty level  $\alpha$ , has been used as a standard value for all tests.

### 5.3 Robustness analysis

Operationally, it is difficult to know the exact probability distributions for the stochastic parameters. Moreover, the real probability distribution might vary along time while the probability distribution used to calibrate the algorithms is considered stable. Therefore, it is relevant to check that the policies and their relative performances are still valid when the reality (test scenarios) is different from what is projected (calibrating scenarios). In order to prevent this drawback, a robustness analysis is performed over the stochastic parameter of the distribution. Actually, policy performances are tested under false or deviant probability distributions. In practice, the tests can be performed in two ways: either the test scenarios are optimistic or pessimistic in comparison to the calibration scenarios of the algorithms, either conversely the calibration scenarios underestimates or overestimates the outcomes of the probability distributions.

Usually, these final tests are performed over policies that showed good performances under exact valuation of the probability distributions. Therefore, these tests can lead to draw conclusions between policies that were not differentiated up to that stage (e.g. see sections 6.8.2 and 7.8.5).

Finally, as dynamic, the expression "robust" and "robustness analysis" should be used carefully as it is used in the present case for a specific analysis. According to the literature (see [Mulvey et al., 1995]), researchers face problems in case of noisy, incomplete or erroneous information. An a-posteriori or reactive analysis resides in the so-called "sensitivity analysis" that checks the feasibility or the optimality of the previously optimal solution when some data change. As in real-world optimization, these biases on data often occur, it might be profitable to be proactive by generating

a-priori a so-called "robust solution". Authors in [Mulvey et al., 1995] highlight that fact for a stochastic problem which can be represented by scenarios: *"A solution to an optimization model is defined as: solution robust if it remains 'close' to optimal for all scenarios of the input data, and model robust if it remains 'almost' feasible for all data scenarios."* As mentioned in [Bertsimas et al., 2011], *"This assumes that uncertainty is known by a probability distribution."*

These authors also describe "robust optimization" which tends to make the solution immune to the stochastic uncertainty, by replacing it by a solution that is feasible for any realization of the uncertainty in a given set.

As introduced in the present thesis, the robustness analysis is specific and linked to check the robustness of a policy. This means that in case of a stochastic problem, the policy performance is ensured even if the projected probability distribution used in the algorithmic technique generating the policy is different from the real probability distribution revealed afterwards.



**Part III**

**Applications**



## Chapter 6

# Multi-period vehicle loading with stochastic release dates

### Abstract<sup>1</sup>

This paper investigates a multi-period vehicle loading problem with stochastic information regarding the release dates of items to be transported. The deterministic version of the problem can be formulated as a large-scale set covering problem. Several heuristic algorithms are proposed to generate decision policies for the stochastic optimization model over a long rolling horizon. The resulting policies have been extensively tested on instances which display the main characteristics of the industrial case-study that motivated the research. The tests demonstrate the benefits of the multi-period stochastic model over simple myopic strategies. A simple and efficient heuristic is shown to deliver good policies and to be robust against errors in the estimation of the probability distribution of the release dates.

**Keywords :** transportation planning, production-transportation coordination, vehicle loading, stochastic programming, set covering, heuristics, consensus algorithms, robustness

---

<sup>1</sup>This chapter has been written in collaboration with Yasemin Arda, Yves Crama, David Kronus and Pascal Van Hentenryck. It is published in EURO Journal on Transportation and Logistics (see [Arda et al., 2014]).



## 6.1 Introduction

### 6.1.1 Coordination of production and transportation

Production and outbound transportation are consecutive tasks in the supply chain of most manufacturing companies. The associated planning processes, such as production scheduling and transportation planning, have been thoroughly investigated in the operations management literature, and numerous optimization models tackle production and transportation decisions. Some of these models integrate production-transportation decisions, with a view towards better coordination of the supply chain; representative examples of this trend of research can be found, for instance, in [Boudia et al., 2007], [Chen, 2004], [Chen, 2010], [Fumero & Vercellis, 1999], [Melo & Wolsey, 2010], [Sarmiento & Nagi, 1999], [Erengüç et al., 1999], [Stecke & Zhao, 2007], [Tang et al., 2007], [Vidal & Goetschalckx, 1997], [Wang & Cheng, 2009], [Yung et al., 2006], and [Zegordi et al., 2010]. As pointed out in the literature surveys by [Chen, 2004], [Chen, 2010], and [Vidal & Goetschalckx, 1997], most models focus on coordination either at the strategic level or at the tactical level: the objective is then to design efficient supply networks or to take transportation costs into account in the framework of production planning. Some authors, like [Boudia et al., 2007], [Wang & Cheng, 2009], or [Zegordi et al., 2010], deal with operational decisions, but this stream of literature is quite recent and typically places the emphasis on production scheduling issues (see [Chen, 2010]).

Actually, a most common situation in practice is that transportation decisions are largely disconnected from production decisions at the operational level: when production items or batches have been completely processed by the manufacturing plant, they become available for shipping, and they are subsequently dispatched by the transportation department. This may be explained by several factors: low complexity of this uncoordinated approach, “rule-of-thumb” according to which immediate shipment tends to decrease the cost of inventory, and, in many cases, lack of communication between various departments.

From a global supply chain perspective, aiming at better coordination of the product flows and at increased customer satisfaction, this is not an ideal process. It would be by far preferable, indeed, to set up an integrated production-transportation plan taking into account, among other constraints, the capacity of the plants and the customer due-dates. But even when such a plan exists, many events can concur to create significant differences between the provisions of the plan and the actual situation faced by transportation managers on a day-to-day basis. Production delays caused by late deliveries of materials, equipment failures, non-conformities, urgent orders, and other obstacles to lean plant management, are still frequent in many industrial environments. Similarly, on the demand side, customer orders may not

coincide, either in time or in quantity, with the forecasts that have been used to establish the plans.

As a consequence, operational shipping decisions often rely solely on available (deterministic) data about items in physical inventory, and transportation is managed in a purely reactive mode with respect to production. The main objective of this paper is to examine whether and how transportation decisions can be improved when they account for *forecasts* about future releases of items from production. As such, our paper takes an intermediate position between a purely *sequential approach*, whereby production and transportation decisions are independently optimized, and a *fully coordinated approach*, whereby both types of processes are simultaneously handled.

From the methodological point of view, taking forecasts into accounts leads us to consider *stochastic optimization* models of transportation planning. We will show that relatively simple, but efficient and robust methods can be used to improve the quality of transportation plans, without necessarily resorting to advanced methods from stochastic programming, which may be much more demanding in terms of implementation effort and computational requirements.

### **6.1.2 Industrial setting**

Our “production-transportation” problem formulation is inspired by a real-world situation encountered in the steel industry. A large number of different steel coils manufactured on a make-to-order basis must be dispatched everyday to customers. At the operational level, the shipping department is informed of a production plan, including confirmed and projected release dates announced by the plants, and of a delivery schedule based on the delivery time windows requested by the customers.

The shipping department charters trucks according to confirmed coil releases from production and according to customers’ requirements. Each truck typically holds one to four coils, depending on their total weight. The economic objective is to minimize the total cost consisting of transportation costs, penalties for early or late deliveries, and inventory costs incurred when available coils are stored in the warehouse before expedition. Shipping decisions are made and are implemented on a daily basis knowing the coils available to be shipped on the first day and the expected release dates of upcoming coils, meaning that only feasible plans are generated and executed at each period. Then, data such as coil releases from production and coil departures from inventory are updated, and this sequential process is repeated day per day over a rolling horizon (see [Sethi & Sorger, 1991] for the corresponding theoretical framework).

The daily trade-off faced by dispatchers can be expressed as follows: “Is it better to ship a given item today, or to wait in the hope of being able to ship it together with

another item which is expected to be released from production in the near future, as this would reduce transportation costs at the risk of incurring delay penalties?” The optimization issue arises from the large number of potential combinations of coils into truckloads, as well as from the selection of the shipment periods. In view of the uncertainty that surrounds the releases of items from production, we consider both expected cost and robustness to be important criteria when evaluating the quality of shipment decisions.

Although our models have not been fully implemented in an industrial environment, they are meant to reflect the major characteristics (objective function, constraints) of the situation that motivated this work. Similarly, the values of the parameters used in all instances are derived from real-world data, as explained in Section 6.6.

### 6.1.3 Outline

In Section 6.2, we present a stochastic programming formulation of the multi-period vehicle loading problem with stochastic release dates, and we briefly review the relevant literature in Section 6.3. Then, Section 6.4 examines a deterministic version of the problem which underlines its combinatorial aspects. This analysis is put to use in Section 6.5, where several heuristic algorithms are proposed for the solution of the stochastic problem. These algorithms have been tested on numerical instances whose structure is described in Section 6.6. Since we are dealing with stochastic optimization problems, comparing the performance of different algorithms on a same instance is not a trivial task: we explain in Section 6.7 how we have carried out the analysis. The computational results are discussed in Section 6.8. Interestingly, this discussion leads to the development of an additional, rather simple algorithm, which is extensively tested in Section 6.9. Section 6.10 contains the conclusions of our research.

## 6.2 Formulation

We consider the following *multi-period vehicle loading problem with stochastic release dates*, or *MVLSRD* problem for short. A set of  $N$  items must be delivered by trucks to several customers over a discrete (rolling) horizon  $\{1, 2, \dots\}$ . Each item  $i = 1, \dots, N$  has several deterministic attributes:

- its weight  $w_i$ ;
- a delivery time window  $[a_i, b_i] \subseteq \{1, 2, \dots\}$ ;
- the warehouse location where the item must be picked up;
- the customer location where the item must be delivered.

A subset of items are ready to be shipped at the beginning of the first (current) period  $t = 1$ . Moreover, the firm has forecasts about the release dates of items from production for a number of subsequent periods  $t = 2, \dots, L + 1$ , where  $L$  denotes the length of the *look-ahead horizon* (see [Sethi & Sorger, 1991]). We represent this information by probabilistic distributions of release dates: for all  $i = 1, \dots, N$  and  $t = 1, \dots, L + 1$ ,  $p_{it} \in [0, 1]$  denotes the probability that item  $i$  will be released in period  $t$  and hence can be shipped in periods  $t, t + 1, \dots$ . We assume that  $\sum_{t=1}^{L+1} p_{it} \leq 1$  and  $p_{i1} \in \{0, 1\}$  for all  $i$  (information relative to the first period is fully revealed). There is an unlimited number of trucks. The maximum total weight that can be loaded on any truck is equal to  $Cap$ . These attributes, together with a number of auxiliary parameters, allow us to compute the cost generated by a truck picking up a given subset of items at their respective warehouses and transporting them to their respective destinations. In our application, all warehouses are located around the same plant, and we are primarily concerned with long-haul transportation. Therefore, the routing aspects are not of primary interest: each truck only visits a couple of warehouses and customers, so that the optimal route can be easily computed for any truck. The total cost generated by a truckload only depends on:

- the composition of the load;
- the total distance driven by the truck;
- the transportation cost per ton and per kilometer;
- an inventory cost, or opportunity cost, depending on the number of periods that each item spends in the warehouse after it has been released from production;
- penalties linked to the period of delivery of items to customers.

This broad definition allows us to integrate various specific features of the cost function (see Section 6.6.1).

The decisions to be made represent the truckloads to be composed and shipped in period  $t = 1$ . As a general rule, grouping items on a same truck is beneficial, and a good shipping decision is based on the following insights: it may be appropriate to ship an item early (with respect to its delivery time window), or conversely, to wait before shipping it (even though it has been released or its time window will be missed) if this results in a reduction of the expected number of trucks required and, more generally, in smaller expected total logistical costs.

Since the horizon is rolling, we actually want to solve an (infinite) sequence of optimization problems, one for each look-ahead horizon  $\{\ell, \dots, \ell + L\}$ , where  $\ell = 1, 2, \dots$ . This results in successive decisions regarding the shipments to be made in each period. The objective of the MVLSRD problem is to minimize the expected cost per unit amount shipped (say, by ton) over the long term:

$$\min_{\pi} Z(\pi) = \lim_{T \rightarrow \infty} E \left[ \frac{\sum_{t=0}^T C(S_t, A^{\pi}(S_t))}{\sum_{t=0}^T W(S_t, A^{\pi}(S_t))} \right], \quad (6.1)$$

where  $\pi$  denotes the decision policy,  $S_t$  denotes the state of the system at time  $t$  (defined by the collection of items available for shipment at time  $t$ , together with their release date),  $A^\pi(S_t)$  denotes the action taken in state  $S_t$  according to policy  $\pi$ ,  $C(S_t, A^\pi(S_t))$  is the associated cost incurred in period  $t$ , and  $W(S_t, A^\pi(S_t))$  is the total weight shipped in period  $t$ . This problem can be viewed as a Markov decision process with very large state space and action space, [Powell, 2011], although we will not explicitly refer to the MDP framework in this paper.

### 6.3 Literature review

To the best of our knowledge, the multi-period vehicle loading problem with stochastic release dates has not been previously investigated.

If the release dates are deterministic and the length of the horizon is fixed, then the problem essentially boils down to a variant of the bin packing problem (see [Dyckhoff, 1990], [Wäscher et al., 2007], and Section 6.4 hereunder). As such, this version of the problem does not pose any specific difficulty and has not attracted the attention of researchers. Online variants of bin packing are closer to our multi-period framework (see [Coffman et al., 1983] for a survey), but they do not assume any probabilistic information about incoming items, and the bin packing literature mostly focuses on the worst-case analysis of simple heuristics.

Transportation planning under uncertainty has been considered in several papers, from different points of view (see [Crainic, 2003]). The stress is usually placed on uncertainty on the demand side and on routing aspects. For instance, many authors examined *vehicle routing problems with dynamic demands* arising in a single period, as evidenced by the surveys [Cordeau, 2007], [Gendreau & Potvin, 1998], [Pillac et al., 2013] and [Psaraftis, 1995]. More recently, [Angelelli et al., 2007], [Angelelli et al., 2009] and [Wen et al., 2010] describe dynamic *multi-period* routing problems where a set of requests need to be served by a fleet of uncapacitated vehicles over a finite discrete horizon. Some of the requests are known initially but more may arrive over time, and each request has a deadline so that the firm may decide to postpone it or not to a later period. The authors discuss the benefits drawn from the multi-period framework with short look-ahead periods. Here again, as in the online bin packing problem, complete uncertainty is assumed about the requests to be served: this hypothesis makes sense in the case, for instance, of requests placed by private customers of a courier company. But it appears to be inappropriate in the case that we consider, where the production plan provides a fair amount of information about the features of the items that will have to be transported and about their expected release dates. Also, the focus in [Angelelli et al., 2007], [Angelelli et al., 2009] and [Wen et al., 2010] is on routing, rather than optimal vehicle loading (the

items are assumed to be small parcels, so that the capacity of the vehicle is not a binding constraint).

Many papers investigate stochastic optimization models for *fleet management*, see [Crainic, 2003], [Frantzeskakis & Powell, 1990], [Powell & Topaloglu, 2003], and [Powell, 2011]. Here, the uncertainty is mostly due again to customer demands that arise randomly over time, and the models emphasize issues related to the repositioning of empty vehicles and to the acceptance or rejection of incoming orders. Multi-period problems of this nature can be handled as sequences of two-stage problems with recourse, where the second stage accounts for all future periods.

None of the papers cited above, however, simultaneously considers the three defining features of our problem, namely, multiple periods, stochastic release dates, and (small) capacity of the vehicles.

## 6.4 Deterministic optimization

In order to better understand the structure of the problem and to prepare the ground for subsequent developments, we first consider the special case of the vehicle loading problem where all release dates are deterministic:  $p_{it} \in \{0, 1\}$  for all  $i = 1, \dots, N$  and  $t = 1, \dots, L + 1$ .

In this case, the problem shares some similarity with a bin packing problem, since we have to “pack” all items into bounded-capacity vehicles so as to minimize a cost function which heavily depends, in practice, on the number of required vehicles ([Coffman et al., 1983] and [Wäscher et al., 2007]). We can formulate this loading problem as a large set covering problem, where each column, or *pattern*, corresponds to a feasible truckload. Each pattern is represented by binary parameters  $q_{ip}$  with the interpretation that  $q_{ip} = 1$  if pattern  $p$  contains the item  $i$ , and  $q_{ip} = 0$  otherwise. Here, feasibility means that the load does not exceed the capacity of the truck and, possibly, that other relevant constraints are satisfied as well.

If we know the starting time  $t$  of the truck, then the corresponding cost  $c_p^t$  of pattern  $p$  can be simply computed as the sum of transportation costs, inventory costs and auxiliary cost elements. Moreover, for each pattern  $p$ , there exists a “best possible starting time” which minimizes the cost  $c_p^t$ , and we can accordingly define  $c_p = \min_{1 \leq t \leq L+1} c_p^t$ . We call  $c_p$  the *cost of pattern*  $p$ .

If  $\Omega$  denotes the set of feasible patterns and  $\theta_p \in \{0, 1\}$  represents the decision to use or not a pattern, then we obtain the following model for the *multi-period vehicle loading problem with deterministic release dates (MVLDRD)*:

$$\min Z^* = \min \sum_{p \in \Omega} c_p \theta_p \quad (6.2)$$

subject to

$$\sum_{p \in \Omega} q_{ip} \theta_p \geq 1 \quad \forall i = 1, \dots, N, \quad (6.3)$$

$$\theta_p \in \{0, 1\} \quad \forall p \in \Omega. \quad (6.4)$$

This set covering representation is usually more efficient, from a computational point of view, than an ILP model based on assignment variables (where  $y_{ip} = 1$  if item  $i$  is included in load  $p$ ) and explicitly expressing the feasibility constraints (see, e.g., [Vanderbeck, 1999]). It allows multiple parameters (item weights, truck capacity, penalties, complex transportation costs, etc.) and side-constraints (heterogeneous fleet, load-vehicle compatibility, route length, etc.) to be “hidden away” in the definition of the patterns, thus leading to a generic model that is suitable for customized applications.

Generating the set of patterns  $\Omega$  gives rise to an auxiliary task that can be either performed in a preprocessing phase, or embedded in the solution phase by use of a dedicated column generation technique ([Vanderbeck, 1999] and [Vanderbeck & Wolsey, 1995]). The former approach becomes rapidly prohibitive if the size of  $\Omega$  is large. In our case study, however, this size remains manageable: short look-ahead horizons, small numbers of available items per period, and small number of items per truckload concur to limit the set of feasible patterns. Moreover, exhaustive generation of the set of patterns allows us to compute rather complex, nonlinear cost functions  $c_p^t$ , such as those encountered in practice (see Section 6.6.1), and to derive the pattern cost  $c_p$ .

Thus, whenever we rely on model (6.2)–(6.4) in the remainder of the paper, we always assume that the complete set of columns is generated in a preliminary phase. This allows us, in particular, to solve the resulting model to optimality by branch-and-bound. (In our computational experiments, we simply feed the model to a generic IP solver.)

**Procedure: Rolling deterministic**

For each period  $\ell = 1, 2, \dots$ , successively, do

1. Let  $t(i)$  be the release date of item  $i = 1, \dots, N$ .
2. Consider all items such that  $t(i) \in \{\ell, \dots, \ell + L\}$ , and solve the associated set covering problem (6.2)–(6.4). Let  $\Theta$  be the collection of patterns selected in the optimal solution of (6.2)–(6.4) and for which the best possible starting time is the current period  $\ell$ :  $c_p = c_p^\ell$  for  $p \in \Theta$ .
3. Remove all items contained in the patterns of  $\Theta$  (that is, constitute and ship the corresponding truckloads), let  $\ell := \ell + 1$  (increase the time counter to the next period), update all data, and repeat steps 1–2.

Figure 6.1: Procedure *Rolling deterministic*

Putting together the pieces of the previous discussion, we conclude that the deterministic MVLDRD problem over a rolling horizon can be handled by the procedure in Figure 6.1. This sequential process can be used to generate a policy over an indefinitely long rolling horizon. As  $L$  increases, we expect it to provide an increasingly better approximation of the long-term optimal policy, but its computational complexity grows accordingly.

## 6.5 Stochastic optimization

Let us now return to the stochastic version of the MVLSRD problem. We can view a decision policy for this problem as a mapping  $\pi$  which, for every instance described as in Section 6.2, selects the patterns to be loaded and shipped in period 1. The quality of a policy is evaluated by the objective function (6.1). We denote the optimal policy by  $\pi^*$  and its value by  $Z(\pi^*)$ . It is the best expected value that can be achieved in the probabilistic environment. From an algorithmic point of view, the policy  $\pi^*$  cannot be easily computed, due in particular to the very large size of the state space, and to the inherent complexity of the deterministic version of the problem (it is easy to infer from Section 6.4 that the deterministic loading problem is NP-hard, just like set covering and bin packing problems).

In this section, therefore, we propose several heuristic methods that produce “good” policies  $\pi_i$ ; these heuristics are based on generic schemes discussed for instance in [Van Hentenryck & Bent, 2006] and [Powell, 2011]. Several of them rely on the generation of *scenarios*: a scenario is a joint realization of the random release dates of items  $i = 1, \dots, N$ .



### 6.5.1 Deterministic equivalent: scenario tree

For every instance of MVLSRD, the set  $S$  of potential scenarios over the look-ahead horizon  $\{1, \dots, L+1\}$  is finite, since any item  $i$  can only be released in one of the periods  $1, \dots, L+1$ , or not at all. This set  $S$  can be represented by a scenario tree, which can be used in turn to set up an *equivalent deterministic formulation* of the problem as an integer programming problem based on the binary decision variables  $\theta_{p,t,s}$ , where  $\theta_{p,t,s} = 1$  if pattern  $p$  is shipped at time  $t$  in scenario  $s$ . As usual, non-anticipativity constraints ( $\theta_{p,t,s_1} = \theta_{p,t,s_2}$ ) are imposed in this model to enforce the consistency of decisions regarding each pattern  $p$  for all pairs of scenarios  $s_1, s_2 \in S$  which coincide over an initial subhorizon  $\{1, \dots, t\}$  (see, e.g., [Birge & Louveaux, 1997]). Taking into account the whole set of scenarios  $S$  leads to a very large-scale IP model, which turns out to be intractable in practice. An approximation of this model can be obtained by restricting the tree to a subset of scenarios. This approach is called “reduced tree approximation” (see [Heitsch & Romisch, 2007a], [Heitsch & Romisch, 2007b] and [Shapiro, 2003]). We have conducted preliminary experiments with reduced trees built on random samples of scenarios (Monte Carlo and stratified sampling consistent with the probability distributions  $p_{it}$ ,  $t = 1, \dots, L+1$ , have been used). Even with as few as 20 items and 5 look-ahead periods, the resulting models proved very hard to solve to optimality by a commercial software package (IBM ILOG CPLEX 11). Therefore, we have abandoned this approach and we have considered alternative, more efficient heuristics.

### 6.5.2 Local Optimization algorithm ( $LO$ )

This simple myopic heuristic reduces the multi-period stochastic problem to a sequence of mono-period deterministic ones. At the current period  $t = 1$ ,  $LO$  solves the deterministic set covering model (6.2)–(6.4) associated with the set of items that are initially available for shipment, thus disregarding any information about future releases of items in periods  $2, \dots, L+1$ . The same process is repeated in the second period, when the information pertaining to this period is revealed, and so forth, to produce a policy with expected value  $Z(LO)$ .

From the industrial perspective, heuristic  $LO$  mimics the procedure implemented by many companies, whereby transportation decisions are made with the objective to ship all available items as soon as possible, without consideration for combinations of items that may potentially bring additional benefits in the future; see Section 6.1.1. We do not expect  $LO$  to be very effective, but it provides a benchmark against which other approaches can be evaluated.

### 6.5.3 Mean Release Date algorithm (*Mean*)

An intuitive and popular approach to stochastic optimization problems consists in replacing all uncertain parameters by their expected values. In our case, this can be translated into selecting a unique scenario for which the release date of item  $i$  is equal to the mean value  $t(i) = \sum_t p_{it}t$  (rounded to the nearest period). The procedure *Rolling deterministic* defined in Figure 6.1 can be used with these release dates to define an algorithm *Mean* that generates a policy with expected value  $Z(\text{Mean})$ . (Note that when the data is updated in Step 3 of the procedure, the expected value of the release dates may change, since the probability distributions  $p_{it}$  must be conditioned by the occurrence or non-occurrence of certain events.)

When compared with the myopic *LO* heuristic, *Mean* takes into account the look-ahead horizon  $L$ , at least to some extent. Thus, the difference between  $Z(\text{LO})$  and  $Z(\text{Mean})$  provides a measure for the value of using a multi-period model with a rough estimate of the future. Our computational results will demonstrate that this simplistic improvement provides significant benefits.

### 6.5.4 Modal Release Date algorithm (*Mod*)

As an alternative to the previous single scenario heuristic, the scenario based on the modal value of the release dates can be used, together with the procedure *Rolling deterministic*, to define an algorithm *Mod* that generates a decision policy with value  $Z(\text{Mod})$ ; here,  $t(i)$  is the value of  $t$  which maximizes  $p_{it}$ , that is,  $t(i) = \operatorname{argmax}_t p_{it}$  for all  $i$  (ties are arbitrarily broken).

### 6.5.5 Consensus algorithm (*Cons*)

A more complex heuristic to approximate the optimal value  $Z(\pi^*)$  is based on *consensus approaches* described in [Van Hentenryck & Bent, 2006]. This family of algorithms provides a generic framework for multi-period stochastic optimization problems. It relies on a simple intuitive idea: if a same decision is frequently made in the optimal solutions associated with a large number of scenarios, then this decision is presumably “good” and can be adopted more generally. For our specific vehicle loading problem, we implement this principle as follows: First, a sample  $\mathcal{S} = \{1, \dots, K\}$  of scenarios is randomly generated. The scenarios in  $\mathcal{S}$  are called *calibrating scenarios*. Next, the set covering model (6.2)–(6.4) corresponding to each calibrating scenario is independently solved: this yields  $K$  sets of patterns  $\Theta^1, \dots, \Theta^K$ , where  $\Theta^j$  is the optimal set of patterns to be shipped in period 1 for scenario  $j \in \mathcal{S}$ .

Notice that, as compared with the reduced tree approach sketched in Section 6.5.1, we do not impose here any non-anticipativity constraints, so that the decisions

$\Theta^1, \dots, \Theta^K$  are usually different, even though the set of items released in period 1 is identical in all scenarios. We define  $D$  to be the set of those items  $i$  that appear in at least  $\lceil K/2 \rceil$  of the solutions  $\Theta^1, \dots, \Theta^K$  (that is,  $D$  is the set of items that are shipped in period 1 in a majority of the calibrating scenarios). In the Consensus algorithm *Cons*, all items contained in  $D$  will be shipped in period 1.

The reader should observe that, at this point, it is usually not clear how the items in  $D$  should be loaded (i.e., how they should be grouped into truckloads); in fact, the set  $D$  may not have been selected for transportation, as a whole, in any of the independent solutions  $\Theta^1, \dots, \Theta^K$ . Moreover, optimizing the transportation of  $D$  may lead to less-than-full truckloads; therefore, it seems reasonable to complete the set  $D$  with additional items that can be loaded without increasing the total cost.

To account for these observations, in a second phase, we solve the following mono-period optimization model, where  $N_1$  is the set of items available in period 1, and  $\Omega$  stands here for the set of patterns that only involve items from  $N_1$ :

$$\min M \sum_{p \in \Omega} c_p \theta_p - \sum_{i \in N_1 \setminus D} \sum_{p \in \Omega} a_{ip} \theta_p \quad (6.5)$$

subject to

$$\sum_{p \in \Omega} a_{ip} \theta_p \geq 1 \quad \forall i \in D \quad (6.6)$$

$$\theta_p \in \{0, 1\} \quad \forall p \in \Omega. \quad (6.7)$$

Constraint (6.6) expresses that all the items in  $D$  must be included in the truckloads. For  $M$  large enough, the first term of the objective function (6.5) ensures that the total cost of the solution does not exceed the optimal cost incurred for  $D$ , whereas the second term tends to maximize the number of items included in the truckloads.

The solution of model (6.5)–(6.6) defines the consensus decision that comes into effect in period 1. Our Consensus algorithm (*Cons*) is obtained by repeating the same process for each successive period. Note that it requires the solution of  $K + 1$  integer programming problems per period, as compared with one IP for each of the three previous algorithms.

### 6.5.6 Restricted Expectation algorithm (*RE*)

A drawback of the Consensus algorithm *Cons* is that it relies on the frequency of individual shipping decisions for each item whereas, in practice, the quality of a transportation plan depends to a large extent on the composition of the loads, that is, on the efficient combinations of items within trucks. We now propose a so-called *Restricted Expectation algorithm (RE)* which avoids the deconstruction of near-optimal truckloads, which is inspired by the general framework described in [Van Hentenryck & Bent, 2006].

Just like *Cons*, our algorithm *RE* starts by generating a random sample  $\mathcal{S} = \{1, \dots, K\}$  of calibrating scenarios and by computing the corresponding optimal sets of patterns for period 1, say  $\Theta^1, \dots, \Theta^K$ . We now look at  $\Theta^1, \dots, \Theta^K$  as candidate decisions, any of which could be implemented in period 1. By definition,  $\Theta^j$  is optimal for scenario  $j$ , but not necessarily for alternative scenarios. In *RE*, the quality of decision  $\Theta^j$  is evaluated for each scenario  $k \in \mathcal{S} \setminus \{j\}$ , as follows: all items included in patterns of  $\Theta^j$  are discarded and model (6.2)–(6.4) is solved for the remaining set of items with the release dates of scenario  $k$ . This allows us to estimate, the expected cost  $E[\Theta^j]$  generated by decision  $\Theta^j$  over all scenarios in the sample  $\mathcal{S}$  (hence the name “restricted expectation”; this is also akin to estimating the expected value of decision  $\Theta^j$  in the framework of approximate dynamic programming, see [Powell, 2011], or sample average approximation methods, see [Vermeij et al., 2002]). Then, *RE* selects and implements in period 1 the decision  $\Theta^j$  with the smallest value of  $E[\Theta^j]$ . The same process can be subsequently repeated for periods 2, 3, . . . , as in the *Rolling deterministic* procedure.

Note that *RE* requires the solution of  $K^2$  integer programming problems for each decision period, which may turn out to be prohibitive for large-scale instances and/or for large values of  $K$ . In our experiments, we have used  $K = 10$  calibrating scenarios. Larger values of  $K$  did not appear to yield significant improvements for most instances.

## 6.6 Numerical instances

In this section, we describe the numerical instances that we have used in our experiments. Although the instances are randomly generated, their main features are intended to mimic the industrial environment that motivated our study. In the following description, we distinguish between deterministic and stochastic parameters of the instances.

### 6.6.1 Deterministic parameters

We consider four classes of instances involving, respectively,  $N = 80, 120, 160$ , and 200 items. The capacity of each truck is  $Cap = 23.5$  tons. The weight  $w_i$  of each item  $i$  is randomly and uniformly generated, strictly between  $0.2 \times Cap$  and  $0.8 \times Cap$ . This distribution puts an upper-limit of 4 items per truckload, similarly to what happens in real life. (The weight of a coil typically ranges between 4.5 and 19 tons.) The delivery cost per ton varies between 30 and 34 monetary units per ton, depending on the customer.

The data are generated over a “long term” horizon of  $T = 20$  periods, and the look-ahead horizon includes  $L = 4$  periods beyond the current one, so as to re-

flect the weekly operational setting. This results in 16 effective planning periods  $\{1, 2, \dots, 16\}$ , whereas periods  $\{17, \dots, 20\}$  account for the definition of the delivery time windows and for end-of-horizon effects (see Section 6.7.5).

For each item  $i$ , the delivery time window covers four consecutive periods  $a_i, a_i + 1, a_i + 2$ , and  $a_i + 3$ . Delivery before  $a_i$  or after  $a_i + 3$  is infeasible, meaning that it carries an infinite penalty.

Note that, in view of these assumptions, there is no point in considering release dates that would result in deliveries outside of the time window  $[a_i, a_i + 3]$  for item  $i$ . Consequently, the release dates of potential interest for item  $i$  must be four consecutive periods  $r_i, r_i + 1, r_i + 2$ , and  $r_i + 3$ , where  $r_i$  is the first possible release period of item  $i$ . Based on transportation times, these four periods can easily be calibrated to match the delivery interval  $[a_i, a_i + 3]$ , or, equivalently, we can set the transportation times to zero and identify the release intervals with the delivery intervals. Accordingly, in the remainder of the paper, we do not distinguish any longer between  $a_i$  and  $r_i$ . In order to create our instances, we generate  $a_i$  uniformly between 1 and  $T$ .

The ideal target dates for delivery are  $a_i + 1$  and  $a_i + 2$ . Early delivery in period  $a_i$  carries a penalty  $PE$ , late delivery in period  $a_i + 3$  carries a penalty  $PL$ . For each period when an item is available for shipment but the firm decides to postpone its delivery, it incurs an inventory cost  $PI$ . We set  $PI = 15$ ,  $PE = 40$  and  $PL = 70$  monetary units. These values are such that, if the item is available in period  $a_i$  or  $a_i + 1$ , then the cheapest period for delivery is  $a_i + 1$ , and the next cheapest period is  $a_i + 2$ . The most expensive delivery period is  $a_i + 3$ . (These choices reflect a “Just-in-Time” supply chain environment.)

In the industrial setting and, accordingly, in our numerical instances, the computation of the transportation cost involves a number of additional factors which further complicate it, but which do not have a direct impact on the performance of our algorithms. For instance, the transportation cost charged by the transporter is fixed up to a certain minimum load and increases linearly as a function of the load beyond this minimum quantity. Moreover, when a truck picks up items from several warehouses and/or delivers them to several customers, an additional handling fee must be paid to the transporter, and so on.

### 6.6.2 Stochastic release dates

Let us now turn to the stochastic models of uncertain release dates. In view of our previous assumptions, probability distributions of interest for the release date of item  $i$  can be expressed in the form  $[p_{i,t}; p_{i,t+1}; p_{i,t+2}; p_{i,t+3}]$ , where  $t, t + 1, t + 2, t + 3$  are the possible release dates of item  $i$ , and where  $p_{ik}$  is the probability that item  $i$  is released in period  $k$ .

In our experiments, we use four distributions for each item  $i$ , namely:

1. “Early”: [40 %; 30 %; 20 %; 10 %] translating optimistic forecasts for the release dates.
2. “Late”: [10 %; 20 %; 30 %; 40 %] translating pessimistic forecasts.
3. “Uniform”: [25 %; 25 %; 25 %; 25 %] translating maximal uncertainty.
4. “Binomial”: [12.5 %; 37.5 %; 37.5 %; 12.5 %] translating Just-in-Time production targets.

These distributions represent various stochastic profiles in the output of the industrial production system. They are assumed here to be independent and identical for all items. Note that this assumption would fail if the variability of release dates was caused by special causes that simultaneously affect all produced items, such as machine breakdowns or other structural disturbances of the production process. Although the algorithms could still be applied, it would probably be more appropriate to adapt them specifically to deal with such situations.

## 6.7 Algorithmic performance evaluation

When evaluating the performance of optimization algorithms, classical criteria are the quality of the solutions that they deliver (optimality, performance ratio, etc.) and their computing time (see [Barr et al., 1995], [Brownlee, 2007], [Hooker, 1995], [Johnson, 2001], [Rardin & Uzsoy, 2001], [Ruml, 2010]).

The evaluation of algorithms for multi-period stochastic optimization problems, however, presents numerous specific hurdles. We explain in this section how the performance of the algorithms has been assessed in our study.

### 6.7.1 Evaluation scenarios

As explained in the previous section, a numerical instance  $I$  of the problem is completely defined by a selection of values for its deterministic parameters (weight  $w_i$  of each item, time windows  $[a_i, a_i + 3]$ , cost parameters, etc.) and by a choice of the probability distribution of release dates (either Early, or Late, or Uniform, or Binomial). In order to assess the performance of an arbitrary algorithm  $\mathcal{A}$  on instance  $I$ , we should be able to evaluate the expected value  $Z(I; \mathcal{A})$  of the cost function over all possible realizations of the random release dates, i.e., all possible scenarios.

In practice, we estimate this expected value over a random sample consisting of *evaluation scenarios* over the long-term horizon  $\{1, \dots, T\}$ . (These evaluation scenarios are generated independently from the calibrating scenarios used in the Consensus and Restricted Expectation algorithms of Section 6.5.) In our experiments, we have used  $V = 30$  evaluation scenarios for each instance.

Note that the computational burden of this evaluation phase is quite heavy. In particular, with our previous notations, estimating the expected value of the solution generated by the Restricted Expectation algorithm  $RE$  for a single instance requires the solution of  $(T - L) * K^2 * V$  medium scale IP problems. With  $T = 20$ ,  $L = 4$ ,  $K = 10$  and  $V = 30$ , this amounts to 48 thousand IP problems for the estimation of  $Z(I; RE)$ .

### 6.7.2 Statistical analysis of results

Since the cost function  $Z(I; \mathcal{A})$  is estimated on the basis of a random sample of evaluation scenarios for each algorithm, small differences between the values obtained for two distinct algorithms  $\mathcal{A}_1$  and  $\mathcal{A}_2$  might be due to random effects only, and their statistical significance must be assessed.

For each fixed instance, we generate the same sample of evaluation scenarios for all algorithms so as to reduce the variance of the tests. This allows us to apply a paired-sample  $t$ -test to compare the values  $\mu_1 = Z(I; \mathcal{A}_1)$  and  $\mu_2 = Z(I; \mathcal{A}_2)$ . We use both two-sided and one-sided tests:

1. test the hypothesis  $H_0 : \mu_1 = \mu_2$  vs.  $H_1 : \mu_1 \neq \mu_2$ ;
2. test the hypothesis  $H_0 : \mu_1 = \mu_2$  vs.  $H_1 : \mu_1 < \mu_2$  (or  $H_1 : \mu_1 > \mu_2$ ).

We say for short that algorithm  $\mathcal{A}_1$  *outclasses* algorithm  $\mathcal{A}_2$  on a given instance if we can conclude that  $\mu_1 < \mu_2$  with a confidence level fixed at 95 %.

### 6.7.3 Fully or partially revealed information

In general, we are not able to compute exactly the optimal value  $Z(\pi^*)$  of an instance  $I$ . But an optimistic estimate of  $Z(\pi^*)$  can in principle be computed as follows: for each evaluation scenario  $s$ , solve the deterministic optimization problem (6.2)–(6.4) over the long-term horizon  $\{1, \dots, T\}$  to obtain the optimal value  $Z^*(s)$ , and average this value over all evaluation scenarios. Since this approach amounts to solving the problem with fully revealed information, it yields a benchmark value  $O^*$  (as Oracle) which, for practical purposes, can be viewed as a lower bound on the optimal value (except for the fact that we work with a sample of evaluation scenarios).

The value  $O^*$ , however, is overly optimistic since in an industrial setting, it is unrealistic to assume that reliable information about the release dates is available over a long horizon. A more relevant benchmark is obtained by assuming that in period  $\ell$ , the firm has “partially revealed” deterministic information regarding the actual release dates during the look-ahead horizon  $\{\ell, \dots, \ell + L\}$  (see, e.g., [Sethi & Sorger, 1991]). Then, the *Rolling deterministic* procedure of Figure 6.1 can be

used to compute a policy with value  $Z^*(s;L)$  for each evaluation scenario  $s$ . Averaging these values over all evaluation scenarios yields a value  $O^*(L)$ , which gives an indication of the best achievable performance in a production environment “under control” over the look-ahead horizon.

#### 6.7.4 Value of information

In order to reduce its total logistical costs, the firm may want to obtain more precise information about the release dates of items (e.g., by reducing the variability of its production processes, by implementing data collection systems, or by improving its production planning systems), or it may want to improve the performance of its vehicle loading algorithms. Both choices are likely to require additional investments that must be counterbalanced by the benefits that they bring. The following measures bring insights into this trade-off (see, e.g., [Birge & Louveaux, 1997] and [Van Hentenryck & Bent, 2006]).

The difference  $Z(\pi^*) - O^*$  represents the *Value of Perfect Information*, since  $Z(\pi^*)$  is the minimum cost that must be incurred if the firm makes the best possible use of the probabilistic information, whereas  $O^*$  is the cost that would be incurred with perfect information. In our computational reports, we cannot compute  $Z(\pi^*)$ , and we use as a surrogate the best cost value obtained by any of our algorithms: that is, we define  $VPI = \min_{\mathcal{A}} Z(\mathcal{A}) - O^*$ .

As mentioned in Section 6.7.3, the bound  $O^*$  is overly optimistic, and it is more realistic to assume that the firm could collect perfect information for the next  $L$  periods only. Therefore, we also compute the *Value of the Accessible Information*:  $VAI = \min_{\mathcal{A}} Z(\mathcal{A}) - O^*(L)$ .

The difference between  $Z(\pi^*)$  and  $Z(Mean)$  represents the value of using the probability distributions in the most effective way, instead of relying on their mean values; we call this difference *Value of the Stochastic Solution* and we approximate it as  $VSS = Z(Mean) - \min_{\mathcal{A}} Z(\mathcal{A})$ .

Finally, we also propose to compute the *Value of the Multi-Period Model* as  $VMPM = Z(LO) - O^*(L)$ : this measures the performance deficit incurred by the myopic mono-period policy  $LO$  when compared with the benchmark where perfect information is available over the look-ahead periods.



## 6.7.5 Computational limitations and bias

We briefly comment here on some of the design choices that we made in our experiments, and which are directly related to the dynamic, multi-period setting of our problem.

### 6.7.5.1 Start and end of horizon

In a rolling horizon context, the first and last periods of the horizon  $\{1, \dots, T\}$  are somewhat special. In a real-world application, the first, or current period is generally affected by past decisions, such as undelivered items carried over from previous periods. Similarly, since the tests cannot be performed over an infinite horizon, each instance suffers from boundary effects at the end of the horizon, as no further item releases are forecast beyond period  $T$ . In order to alleviate these effects, we work with a “sufficiently long” horizon  $T$ , which reduces the influence of the initial conditions, and we make shipping decisions for periods 1 through  $T - L$  only.

### 6.7.5.2 Objective function

The total logistical costs represent the cost of all performed decisions, meaning that only delivered items are taken into account. Because of the look-ahead periods introduced at the end of horizon, and because it is in the nature of our problem that some shipments may be postponed, all algorithms do not ship exactly the same set of items during the last few periods. This potential bias in the comparison of algorithms is mitigated, however, by the length of the horizon, and, mostly, by the fact that the objective function actually is the *average cost per ton shipped* (see Eq. (6.1)).

Let us also mention that in the industrial application, the total cost incurred for each item is made up of a significant fixed cost (equal to the weight of the item multiplied by a unit cost) which cannot be improved by any algorithm. Therefore, we remove this fixed component from all our estimations, which only include improvable elements such as temporal penalties, multiple pick-up and unloading fees, or opportunity costs for loading the truck under its purchased capacity (see Section 6.6.1).

## 6.7.6 Computing times

Our computational experiments have been performed on a personal laptop computer (Core 2 Duo 2GHz, 2GB of RAM, Windows). The running time of the algorithms is mostly determined by the number of integer programming subproblems that must be solved. In our experiments, these IP subproblems have been solved using IBM ILOG CPLEX 11 with default settings.

For the instance sizes that we consider in this paper, all our algorithms meet the computing time requirements for operational daily use, namely the decision relating to a single period is obtained in a few seconds. This is essentially due to the fact that the number of items that come into consideration over  $L$  look-ahead periods is relatively small so that the number of feasible patterns and the associated set covering problems remain accordingly manageable. (*Evaluating* the quality of the algorithms is a different matter, as we explained in Section 6.7.1, since this requires the solution of a huge number of IPs.) Table 6.1 displays the computing time for solving one instance of problem (6.2)–(6.4) with  $L = 4$  and  $N$  items. For algorithm *RE*, this computing time would be roughly multiplied by a factor  $K^2$ , as mentioned in Section 6.5.6.

Table 6.1: Problem size and computing speed for problem (6.2)–(6.4)

$L = 4$	$N = 50$	$N = 100$	$N = 200$
<b>Average number of patterns</b>	1403	6059	33399
<b>CPU time (seconds)</b>	0.3	1.3	13.0

Therefore, in the following sections, we do not discuss computing times and we concentrate, rather, on a comparison of the quality of the solutions provided by different algorithms.

## 6.8 Computational results

### 6.8.1 First results

Tables 6.2 to 6.5 report the expected value of the objective function (estimated over the evaluation scenarios) for the different algorithms and probability distributions. The best value obtained for each instance is in boldface.

For an easier understanding of the results, all values are expressed as percentages of the lower bound  $O^*$  (which corresponds therefore, to the value 100 %). Note that this standardization of the objective function masks the decrease of the transportation cost per ton when the number of coils increases from 80 to 200. This scale effect is not surprising: indeed, when there is a large number of items with

uniformly distributed weights, we can expect that it is possible to allocate the items to a number of trucks that are loaded to full capacity, or very close to it in each period (see for instance, [Rhee, 1988]). As a consequence, the transportation cost per ton decreases, and the performance of most algorithms tends to improve as the instances become somewhat easier.

Table 6.2: Algorithmic performance – Early distribution

<b>Early</b>	$N = 80$	$N = 120$	$N = 160$	$N = 200$
$O^*$	100	100	100	100
$O^*(4)$	107.2	105.3	103.8	105.6
<b>LO</b>	193.5	172.5	168.7	159.3
<b>Mean</b>	116.5	113.9	111.1	110.4
<b>Mod</b>	112.2	<b>108.5</b>	<b>107.0</b>	<b>107.6</b>
<b>Cons</b>	122.4	119.5	113.1	117.4
<b>RE</b>	<b>111.0</b>	111.7	109.2	111.8
$O^*(4) - O^*$	7.2	5.3	3.8	5.6
<b>VMPM</b>	86.3	67.2	65.0	53.7
<b>VPI</b>	11.0	8.5	7.0	7.6
<b>VAI</b>	3.8	3.2	3.2	2.0
<b>VSS</b>	5.4	5.4	4.1	2.8

Table 6.3: Algorithmic performance – Late distribution

<b>Late</b>	$N = 80$	$N = 120$	$N = 160$	$N = 200$
$O^*$	100	100	100	100
$O^*(4)$	102.7	103.0	102.8	103.8
<b>LO</b>	154.3	144.0	142.3	136.8
<b>Mean</b>	119.6	115.0	112.0	113.2
<b>Mod</b>	120.1	117.0	117.9	115.4
<b>Cons</b>	109.7	<b>110.1</b>	109.8	111.2
<b>RE</b>	<b>109.5</b>	111.0	<b>109.0</b>	<b>109.7</b>
$O^*(4) - O^*$	2.7	3.0	2.8	3.8
<b>VMPM</b>	51.6	41.1	39.5	33.0
<b>VPI</b>	9.5	10.1	9.0	9.7
<b>VAI</b>	6.8	7.1	6.2	5.9
<b>VSS</b>	10.0	4.9	3.0	3.5

Table 6.4: Algorithmic performance – Uniform distribution

<b>Uniform</b>	$N = 80$	$N = 120$	$N = 160$	$N = 200$
$O^*$	100	100	100	100
$O^*(4)$	108.1	104.2	102.9	104.9
<b>LO</b>	179.5	159.2	154.6	147.5
<b>Mean</b>	117.7	112.3	<b>109.6</b>	109.9
<b>Mod</b>	125.1	118.6	113.9	113.3
<b>Cons</b>	115.7	114.7	111.6	113.3
<b>RE</b>	<b>112.1</b>	<b>112.2</b>	110.0	<b>108.7</b>
$O^*(4) - O^*$	8.1	4.2	2.9	4.9
<b>VMPM</b>	71.4	55.1	51.7	42.6
<b>VPI</b>	12.1	12.2	9.6	8.7
<b>VAI</b>	4.0	8.0	6.7	3.8
<b>VSS</b>	5.6	0.1	<b>0</b>	1.2

Table 6.5: Algorithmic performance – Binomial distribution

<b>Binomial</b>	$N = 80$	$N = 120$	$N = 160$	$N = 200$
$O^*$	100	100	100	100
$O^*(4)$	107.2	105.8	104.4	106.1
<b>LO</b>	184.8	179.0	160.9	157.3
<b>Mean</b>	123.4	117.2	114.7	116.7
<b>Mod</b>	123.4	<b>109.9</b>	112.4	115.0
<b>Cons</b>	114.7	114.0	113.5	115.3
<b>RE</b>	<b>113.0</b>	111.6	<b>112.1</b>	<b>111.9</b>
$O^*(4) - O^*$	7.2	5.8	4.4	6.1
<b>VMPM</b>	77.6	73.2	56.5	51.2
<b>VPI</b>	13.0	9.9	12.1	11.9
<b>VAI</b>	5.8	4.1	7.7	5.8
<b>VSS</b>	10.5	7.3	2.6	4.8

Here are some observations issued from these empirical tests:

1. The *Value of the Multi-Period Model*  $VMPM = Z(LO) - O^*(4)$  is quite large: it varies from 30 % to more than 80 %. This underlines the important benefit of taking several periods into account for transportation planning, rather than only the current period as in the myopic policy  $LO$ .
2. On the other hand, although  $O^*(4)$  is significantly larger than  $O^*$  at a confi-

dence level of 99 %,  $O^*(4) - O^*$  is not excessively large from a managerial point of view. (See, e.g., [Rardin & Uzsoy, 2001] for a discussion of statistical versus practical significance.) This suggests that very good policies might be achieved with relatively short look-ahead horizons, such as those that are likely to be implemented in practice. In fact, in our preliminary experiments, we have also computed  $O^*(L)$  for different values of  $L$ ; the tests showed that  $O^*(L)$  decreases rather quickly when  $L$  increases from 1 to 4, and decreases at a slower pace for larger values of  $L$ .

3. *Mean* and *Mod* perform rather poorly in most cases. An intriguing exception is the good performance of *Mod* for the Early distribution of release dates (we return to it later).  $VSS = Z(\text{Mean}) - \min_{\mathcal{A}} Z(\mathcal{A})$  ranges from 0 up to 10 %, meaning that practitioners may want to invest some efforts into the development of algorithms making effective use of the probability distributions.
4.  $VAI = \min_{\mathcal{A}} Z(\mathcal{A}) - O^*(4)$  varies from 4 to 8 %: for some instances, there is considerable value in reducing the uncertainty that surrounds the release dates.
5. The Restricted Expectation algorithm *RE* performs well: it provides the best expected values 10 times out of 16, the second best value 5 times, and the third best value once.

The performance of *RE* is substantiated by the results of hypothesis tests displayed in Table 6.6. Here, we have tested whether there is a significant difference between the mean values obtained by *RE* and by any of three other algorithms, namely, *Mean*, *Mod*, and *Cons*. (*LO* is clearly not in the same league.) The entries in the table indicate the number of instances for which the null hypothesis  $H_0 : \mu_{RE} = \mu_{\mathcal{A}}$  is rejected against the alternative hypothesis  $H_1$ , for each algorithm  $\mathcal{A}$ , at a confidence level of 95 % (see Section 6.7.2).

Table 6.6 shows that *RE* outclasses all other algorithms on most instances. It is statistically tied with *Mean* for a few instances, and with *Cons* for a majority of instances, but it is never outclassed by any of these two algorithms. An interesting exception is *Mod*, which significantly outclasses *RE* on three instances associated with the Early distribution; we refer to Section 6.9 for an analysis of this exception.

Table 6.6: Comparison of means for  $RE$  vs. alternative algorithms

Reject $H_0$ vs. $H_1$	$\mathcal{A} = Mean$		$\mathcal{A} = Mod$		$\mathcal{A} = Cons$	
	Yes	No	Yes	No	Yes	No
$H_1 : \mu_{RE} \neq \mu_{\mathcal{A}}?$	12	4	13	3	7	9
$H_1 : \mu_{RE} < \mu_{\mathcal{A}}?$	12	0	10	0	7	0
$H_1 : \mu_{\mathcal{A}} < \mu_{RE}?$	0	0	3	0	0	0

### 6.8.2 Robustness analysis

*Robustness* may take various meanings in operations research. In broad terms, robust optimization aims at finding the (near)-optimal solution of a mathematical programming problem under a set of constraints that represent all possible realizations of uncertain parameters (see [Bertsimas et al., 2011] and [Kouvelis & Yu, 1997]).

Thus, typically in robustness analysis, the quality (say, the cost) of a solution is measured under adverse realizations of the uncertain events. This focus on worst-case scenarios, however, gives rise to optimization problems that are quite hard to solve. More importantly, it may be seen as overly pessimistic in our planning framework where decisions are repeatedly made over a long-term horizon, and where it is, therefore, unlikely that adverse scenarios will systematically materialize.

Therefore, in our experiments, we have tested the robustness of the Restricted Expectation algorithm  $RE$  in a different way: rather than assuming that the worst-case scenario unfolds for a given policy, we have tried to understand what happens when the decision-maker has poorly estimated the distribution of release dates. (This is akin to “distributional uncertainty” in the theoretical framework of [Bertsimas et al., 2011].) Let us describe more precisely our experimental setting.

Let  $\mathcal{D}$  be a fixed probabilistic distribution of release dates,  $\mathcal{D} \in \{\text{Early, Late, Uniform, Binomial}\}$ , and let  $RE_{\mathcal{D}}$  be the variant of  $RE$  where all calibrating scenarios are drawn according to  $\mathcal{D}$  (see Section 6.5.6). This reflects the situation where the decision-maker believes that  $\mathcal{D}$  is the true distribution of release dates. Then, for a given instance  $I$ , the expected cost of the policy generated by  $RE_{\mathcal{D}}$  is estimated by drawing  $V$  evaluation scenarios (as in Section 6.7.1) according to a different distribution  $\mathcal{R}$ , which is meant to represent the real, unknown distribution of release dates.

Our preliminary experiments were carried out with two “extreme choices”, namely  $RE_{\text{Early}}$  was evaluated over the late distribution  $\mathcal{R} = \text{Late}$ , and conversely,  $RE_{\text{Late}}$  was evaluated over the early distribution  $\mathcal{R} = \text{Early}$ . These experiments revealed that  $RE_{\text{Late}}$  performs rather poorly, whereas  $RE_{\text{Early}}$  seems to resist quite well to a wrong estimation of the distribution. The next section builds on these observations.

## 6.9 An optimistic algorithm

The previous results lead us to the hypothesis that an “optimistic” decision-maker, who always assumes that future items will be released as early as possible, may turn out to achieve a very good cost performance. This hypothesis is motivated, in part, by the results of Section 6.8.2, where  $RE_{Early}$  appeared to be a robust algorithm. But it is also supported by the observations made in Section 6.8.1 regarding the performance of algorithm *Mod* on the instances with an Early distribution (see Table 6.2 and Table 6.6). Indeed, when applied to the Early distribution  $[p_{i,t}; p_{i,t+1}; p_{i,t+2}; p_{i,t+3}]$  where  $p_{i,t} > p_{i,t+1} > p_{i,t+2} > p_{i,t+3}$ , the modal scenario used by *Mod* is exactly the scenario where each item is released at the earliest possible date, namely in period  $t$ .

Thus, we define a simple algorithm *Optimist* as follows: For each instance, *Optimist* selects a unique scenario in which the release date of item  $i$  is equal to its earliest feasible release date:  $t(i) = \min\{t : p_{it} > 0\}$ . Then, the procedure *Rolling deterministic* of Section 6.4 is used with these release dates to define the policy *Optimist*.

### 6.9.1 Comparative performance of algorithm *Optimist*

Procedures  $RE_{Early}$  and *Optimist* have been tested on the 16 instances already used in Section 6.8. Note that each instance  $I$  defines feasible release dates  $\{t, t+1, t+2, t+3\}$  for each item, as well as a (real) probability distribution  $\mathcal{R}$  over these dates (see Section 6.6.2). For  $RE_{Early}$ , the calibrating scenarios are drawn from the Early distribution over the feasible release dates of each item. For both algorithms  $RE_{Early}$  and *Optimist*, the evaluation scenarios are generated from the real distribution  $\mathcal{R}$  associated with the given instance. (So, when  $\mathcal{R}$  is the Early distribution,  $RE_{Early}$  coincides with  $RE$ .)

The results are displayed in Tables 6.7 to 6.10. Quite remarkably, the simple algorithm *Optimist* yields the best solution for 10 instances out of 16, and the second best solution for three additional instances. The algorithm  $RE_{Early}$  provides three times the best value and seven times the second best one (beaten only by *Optimist*, but often close to it in such cases).

Table 6.7: Algorithmic performance – Early distribution

<b>Real <math>\mathcal{R} = \text{Early}</math></b>	$N = 80$	$N = 120$	$N = 160$	$N = 200$
$O^*$	100	100	100	100
$O^*(4)$	107.2	105.3	103.8	105.6
$RE_{Early}$	<b>111.1</b>	111.7	109.2	111.8
<i>Optimist</i>	112.2	<b>108.5</b>	<b>107.0</b>	<b>107.6</b>

Table 6.8: Algorithmic performance – Late distribution

<b>Real <math>\mathcal{R} = \text{Late}</math></b>	$N = 80$	$N = 120$	$N = 160$	$N = 200$
$O^*$	100	100	100	100
$O^*(4)$	102.7	103.0	102.8	103.8
$RE$	<b>109.5</b>	111.0	109.0	109.7
$RE_{\text{Early}}$	111.7	<b>108.6</b>	<b>106.9</b>	108.8
$Optimist$	110.1	109.2	107.3	<b>108.3</b>

Table 6.9: Algorithmic performance – Uniform distribution

<b>Real <math>\mathcal{R} = \text{Uniform}</math></b>	$N = 80$	$N = 120$	$N = 160$	$N = 200$
$O^*$	100	100	100	100
$O^*(4)$	108.1	104.2	102.9	104.9
$RE$	<b>112.1</b>	112.2	110.0	108.7
$RE_{\text{Early}}$	113.7	111.4	107.8	109.5
$Optimist$	113.6	<b>109.7</b>	<b>107.3</b>	<b>108.4</b>

Table 6.10: Algorithmic performance – Binomial distribution

<b>Real <math>\mathcal{R} = \text{Binomial}</math></b>	$N = 80$	$N = 120$	$N = 160$	$N = 200$
$O^*$	100	100	100	100
$O^*(4)$	107.2	105.8	104.4	106.1
$RE$	<b>112.9</b>	111.6	112.1	111.9
$RE_{\text{Early}}$	117.3	113.5	112.1	111.7
$Optimist$	116.0	<b>109.0</b>	<b>109.5</b>	<b>110.3</b>

These observations are confirmed in Table 6.11, where we display the conclusions of several tests of hypotheses regarding the performance of *Optimist*. Here, the algorithms competing with *Optimist* are  $\mathcal{A} = RE_{\text{Early}}$  (evaluated on all instances), and  $\mathcal{A} = RE$  (evaluated on the instances associated with the Late, Uniform, and Binomial distributions). The tests show that *Optimist* outclasses the other algorithms on six instances and is never outclassed.



Table 6.11: Comparison of means for *Optimist* vs. alternative algorithms

Reject $H_0$ vs. $H_1$	$\mathcal{A} = RE_{Early}$		$\mathcal{A} = RE(L-U-B)$	
	Yes	No	Yes	No
$\mu_{Optimist} \neq \mu_{\mathcal{A}}$	6	10	6	6
$\mu_{Optimist} < \mu_{\mathcal{A}}$	6	0	6	0
$\mu_{\mathcal{A}} < \mu_{Optimist}$	0	0	0	0

### 6.9.2 Additional results

The performance of *Optimist* has been further assessed on a larger testbed of 80 additional instances. Namely, for each probability distribution of release dates, we have generated 10 random instances of size  $N = 120$  and 10 instances of size  $N = 160$  with different item characteristics, such as weight and first possible release period. The unit transportation costs are smaller for the instances with 160 items.

The expected value of the solution produced by *Optimist* is reported in Table 6.12 (averaged over 10 instances of the same class), as a percentage of the lower bound  $O^*$ . We see that *Optimist* performs very well in all cases. On average, it comes within 8 % of  $O^*$  for the instances with 120 items, and within 5 % for the instances with 160 items. The standard deviation is around 3 % in all cases. We also observe that, not surprisingly, *Optimist* tends to perform slightly better when the release dates follow the Early probability distribution.

Table 6.12: Algorithmic performance of *Optimist* over 80 instances

$Optimist/O^*$	$N = 120$	$N = 160$
<b>Early</b>	106.7	103.8
<b>Late</b>	108.7	105.1
<b>Uniform</b>	108.6	105.8
<b>Binomial</b>	109.1	106.8

### 6.9.3 Interpretation

The observations in Section 6.9.1 and Section 6.9.2 are quite unexpected and interesting. Indeed,

1. *Optimist* is a much simpler and much faster algorithm than either *RE*, or *RE<sub>Early</sub>*, or even *Cons*. At every period, it only relies on a single scenario and it solves a single set covering problem (6.2)–(6.4) in order to determine the items to be shipped, as opposed to  $K^2$  IP's for *RE* and *RE<sub>Early</sub>*, or  $(K + 1)$  IP's for *Cons*.

2. The complexity of *Optimist* is similar to that of either *Mean* or *Mod*, but the solutions that it produces are significantly better.
3. Most interestingly, perhaps, *Optimist* does not require any information regarding the probabilistic distribution of release dates! It only focuses on the first feasible release date of each item, which essentially depends on its delivery time windows. Therefore, it can easily be implemented in an industrial environment.

A possible explanation for the good performance of this simple algorithm goes as follows. In the current period, say period 1, *Optimist* faces a subset of items that are available for shipment. It establishes a scenario for the upcoming periods based on earliest possible release dates. In comparison with a random scenario, this optimistic scenario foresees relatively many releases over the look-ahead horizon (say, in periods 2 to 5). Hence, this increases the likelihood of postponing the shipment of an item  $i$  that is available in the current period, due to the perspective of shipping  $i$  together with another item in the near future, at a lower cost.

There are now two possibilities for the next periods. Either the observed releases are close to the optimistic forecast, in which case postponing item  $i$  was a good decision, or the releases deviate from the forecast. In the latter case, the transportation cost of  $i$  may increase, but there is still a reasonable chance that  $i$  will be efficiently combined with other items so that the total penalty due to postponement may not be too high (in practice, the inventory costs are small relative to the cost of a truck). In summary, according to this tentative explanation, the algorithm *Optimist* would incite the manager to delay the shipments more frequently than, say, a myopic strategy would do, and this postponement strategy should yield benefits.

Although it is not easy to validate completely the various elements involved in this intuitive explanation, we can exhibit some evidence to support it. First, it appears that shipments are indeed more frequently postponed in the solutions generated by *Optimist* than in solutions generated by other algorithms. This is illustrated by two instances from Table 6.7 on which *Optimist* performs, respectively, worse ( $N = 80$ ), or better ( $N = 200$ ) than *RE* (the other instances yield similar conclusions). For each instance, we have collected the average expeditions dates  $d_{Opt}$  and  $d_{RE}$  of all items over the 30 evaluation scenarios, for the solutions generated by the two algorithms. Table 6.13 shows the average difference  $d_{Opt} - d_{RE}$ , as well as the standard deviation  $\sigma_d$  of these differences. Although the average difference may appear to be rather small, a  $t$  test significantly rejects the hypothesis that  $d_{Opt} = d_{RE}$ , to the benefit of the alternative hypothesis  $d_{Opt} > d_{RE}$  (at high significance levels  $\gg 0.99$ ). We also observe that the differences tend to get larger when *Optimist* performs better.

Table 6.13: Difference in expedition dates for two instances from Table 6.7

	$N = 80$	$N = 200$
$d_{Opt} - d_{RE}$	2.9	8.1
$\sigma_d$	3.6	5.2
$t$ test statistics	4.33	8.37

Furthermore, we note that postponing shipments is unlikely to entail large penalties when the number of items is relatively high in each period, since this increases the number of favorable combinations of items and hence reduces the likelihood that a postponed item may require an additional truck. The results in Tables 6.7 to 6.10 accordingly show that algorithm *Optimist* often performs better than other algorithms when  $N$  gets larger.

These observations, by themselves, cannot fully validate our tentative explanation according to which “postponing decisions tends to produce better outcomes when facing an uncertain future”, but they provide partial support for this plausible interpretation.

## 6.10 Conclusions

In this article, we have investigated an original multi-period vehicle loading problem including stochastic information about releases from production. A main objective of this research was to examine the benefits that the firm can draw from improving the coordination between production and transportation. Improved coordination means, in particular, that transportation of goods to the customers should not be managed on a purely reactive and myopic mode, whereby items are shipped as soon as they are released from the plant, but that it should be optimized on the basis of forecasts derived from the production plan.

We have proposed a stochastic optimization model with rolling-horizon for this problem. The deterministic version of the model, where all release dates are assumed to be exactly known over a fixed horizon, can be formulated as a set covering problem (closely resembling the bin packing problem) and, for realistic problem sizes, can be solved by a commercial IP solver. The stochastic model, however, is much more difficult. We have proposed several efficient heuristic algorithms for its solution.

The corresponding policies have been extensively tested on randomly-generated instances which share the main characteristics of the industrial application that motivated our study. As we are dealing with stochastic optimization problems, particular attention has been paid to the estimation of the objective function (expected cost over a rolling horizon), to the statistical significance of the comparisons, and to the robustness of the results.

Our main conclusions are as follows:

1. The multi-period setting provides a clear benefit over the myopic procedures that are frequently used in practice. In our experiments, the value of the multi-period model (VMPM) is very high and the Local Optimization algorithm (*LO*) is dominated by all heuristics that take future scenarios into account.
2. Even though it does not make much use of the special structure of the problem, the generic *Restricted Expectation algorithm (RE)* yields very good results and is quite robust.
3. The simple heuristic *Optimist*, which combines point forecasts based on earliest release dates with truckload optimization over a short look-ahead horizon, performs surprisingly well. It is at least on a par with and often superior to much more complex algorithms. It allows us to close a large fraction of the gap between the cost of the “optimal policy” (computed ex post under conditions of perfect information) and the cost of the myopic policy *LO*. Moreover, *Optimist* appears to be robust under a variety of assumptions regarding the probability distributions of release dates.
4. The Value of the Accessible Information, i.e.,  $VAI = \min_{\mathcal{A}} Z(\mathcal{A}) - O^*(L)$ , may be relatively large, which suggests that the firm would benefit from reducing the uncertainty that surrounds the release dates, at least over a short look-ahead horizon. In order to attain this ultimate goal of supply chain coordination, the firm may have to rely on improvement practices derived from lean management, total productivity management, or quality management. Such developments are outside the scope of our study.
5. From a methodological point of view, these contributions illustrate the benefits of bridging part of the existing gap between the pragmatic, but suboptimal methods frequently used in the industry, and more sophisticated methods proposed in the scientific literature in order to deal with decision-making situations that involve a significant amount of uncertainty. In particular, we believe that the approaches proposed in this paper provide an interesting response to two of the main limitations encountered in the implementation of stochastic optimization models, namely, their computational hardness, and the difficulty to provide detailed estimates of the relevant probability distributions in an industrial setting. The resulting methods are easily implemented, computationally efficient, robust with respect to the estimation of model parameters, and they deliver considerably better solutions than myopic procedures which do not take forecasts into account.

**Acknowledgments**

We are grateful to the supply chain department of ArcelorMittal Liège for providing us with industrial data, and to Frédéric Semet for his comments during the course of our research. We thank the reviewers for their helpful comments. The project leading to these results was partially supported by the Interuniversity Attraction Poles Programme initiated by the Belgian Science Policy Office, Grant P7/36.

## Chapter 7

# Multi-period vehicle assignment with stochastic load availability

### Abstract<sup>1</sup>

This work investigates optimization techniques for a vehicle-load assignment problem. A company owning a limited fleet of vehicles wants to maximize its operational profit over an infinite horizon divided into periods. The profit stems from revenues for transporting full truckloads and costs derived from waiting idle and moving unladen. The stochastic component of the problem arises from projections on the realization of each transportation order, i.e. load. The methodology is based on optimizing decisions for deterministic scenarios. Several policies are generated in this way, from simple heuristics to more complex approaches, such as consensus and restricted expectation algorithms, up to policies derived from network flow models formulated over subtrees of scenarios. Myopic and a-posteriori deterministic optimizations models are used to compute bounds allowing for performance evaluation. Tests are performed on various instances featuring different number of loads, graph sizes, sparsity, and probability distributions. Performances are compared statistically over paired samples. The robustness of various policies with respect to erroneous evaluations of the probability distributions is also analyzed.

**Keywords:** transportation, multi-period, stochastic, vehicle-load assignment problem

---

<sup>1</sup>This chapter of the thesis is written as an article. It is unpublished yet. It has been written in collaboration with Yves Crama and revised linguistically by Isabelle Peere.

## 7.1 Introduction

### 7.1.1 Industrial problem

The present article investigates a Dynamic Vehicle Allocation problem (DVAP) which is faced by major forwarding companies active in road transportation. A company owning a limited fleet of vehicles wants to maximize its operational profit over an infinite horizon divided into equal periods (days). The profit stems from revenues for transporting full truckloads (FTL) and from costs derived from waiting idle and moving unladen. A decision leading to a set of actions is made at every period of time and is based on the dispatcher's information over a restricted horizon, called rolling horizon, as it evolves subsequently period per period. The data provided by the customers concern their prospective loads or requirements for transportation: locations of departure and destination cities, and a unique pick-up period for each load. Moreover, the dispatcher has data regarding travel times between cities, current location and status (unladen or loaded) of trucks. This is known for sure and represents the deterministic part of the problem.

The stochastic component of the problem arises from the uncertainty on the requirement for the transportation of loads. More precisely, the availability of each load can be either confirmed, or denied, a few periods ahead of the loading period (meaning that clients confirm their order, which the carrier may still decide to fulfill, or not). For projected loads in the remote part of the rolling horizon, the dispatcher only knows the load confirmation probability which represents the stochastic load availability.

The decision problem faced by the dispatcher in each period is to accept or to reject loads, then to assign the selected loads to trucks, taking into account prospective and confirmed loads as well as the availability and current location of trucks. Rejected loads are supposed to be sub-contracted at no cost while bringing no profit. The main objective of this research is to provide efficient algorithmic strategies to tackle this multi-period vehicle-load assignment problem over a rolling horizon including prospective transportation orders.

In practice, as a rule, trucking orders are communicated by the dispatching center to the drivers and to the customers on the eve of the loading period at the latest. The loading decisions are made when all available loads are known for the next day. In the present work, it is slightly different, decisions are supposed to be taken at the early morning of the current decision period when the information for the current day are all known. So, this process is performed with short regular periods, typically equal to days. This results in a planning context which is different from an on-line setting, where the system generates a decision whenever a piece of information changes, or a model with long periods of time, such as in maritime

transportation, wherein the accuracy of the probabilistic forecasts may evolve over several weeks.

This problem is computationally difficult owing to the large number of possible realizations of the random variables, and to the combinatorial nature of the decision space. Our methodology is based on optimizing decisions for deterministic scenarios, so as to alleviate the stochastic aspect of the problem. By solving the assignment problem for a sample of scenarios, by mixing solutions or by evaluating them at each period, we aim at finding actions leading to profitable policies in the long run.

### **7.1.2 Dynamics and stochasticity**

Within the literature on transportation problems, many developments focus on finding one approximate or exact solution to a single period deterministic problem such as the vehicle routing problem (VRP) or the pick-up and delivery problem (PDP), including their variants: limited or unlimited capacity, single vehicle or fleet, time windows, etc. (See [Berbeglia et al., 2007] for a detailed review.)

In such problems, the temporal dimension is usually almost absent: it may be restricted, for instance, to an upper bound on the duration of a tour, or to time windows at customer locations. In other cases, such as when designing bus or train schedules, the decision-maker may restrict his attention to the computation of periodic schedules. Even in this case, where the problem takes several periods into account, a unique decision is made once and for all.

Yet, in other decision frameworks, the time dimension cannot be so easily disregarded. In particular, long-haul transportation firms usually have to cope with a rolling decision process, typically unfolding over days, where deterministic or uncertain forecasts over subsequent periods are available from customers, and where decisions can be possibly postponed or anticipated. This framework leads to a decision process consisting of successive dependent optimization phases. The present research aims at describing a strategy to tackle such problems, and techniques to select a performing algorithmic policy.

We should also stress that the stochastic component of the problems that we handle here is not quite standard. Indeed, in transportation models, the uncertain or stochastic element frequently relates to a parameter value such as customer demand or travel time which can be reasonably modeled by continuous probability distributions (see [Gendreau et al., 1996] for an introduction). More importantly, even though the realization of these random variables may affect the cost or the feasibility of the solutions, it does not modify the set of decision variables.

In contrast with this situation, we consider here a discrete, Bernoulli random variables (representing the availability of loads) whose realization profoundly af-



fects the structure of the available solutions.

In fact, owing to this discrete distribution, the optimization techniques for the model to be solved are scenario-based. Therefore, some decision variables may typically appear in, or disappear from the models associated with these various deterministic scenarios depending on the binary values assumed by these random variables (as in [Thomas, 2007] for an application).

These binary values determine the problem size for every scenario and the set of potential decisions. So, each scenario represents a particular set of loads available for transportation within the rolling horizon. This set obviously contains the loads confirmed in the deterministic part of the horizon, as well as the loads associated with the specific scenario at hand. Consequently, the decision to be made, i.e., the problem to be solved, changes significantly with each scenario, and in every period. This is by far different from a slight variation in the values assumed by numerical parameters, such as those that might result in scenarios associated with realizations of continuous random variables.

Moreover, the classical two-stage formulation from stochastic programming is not really appropriate in our rolling horizon framework. Indeed, the cost of the exact recourse (or lost income) cannot be easily approximated in a rolling horizon procedure wherein period costs are tightly dependent on each other and wherein the information is only released on a sub-part of the whole horizon.

Our contribution in this chapter consists in testing the effectiveness of a generic, practical methodology for tackling such multi-period transportation problems with discrete stochastic forecasts. As an additional contribution, we also provide illustrative numerical estimations and discussions for the expected value of the multi-period model (EVMPM), the expected value of the perfect information (EVPI), the expected value of the accessible information (EVAI), the expected value of the expected value solution (EEVS) and the expected value of the stochastic solution (EVSS). Finally, we analyze the robustness of the algorithm with respect to the accuracy of the estimation of the probability distributions, and we discuss the sensitivity of the performance of the most promising algorithm with respect to the length of the rolling horizon.

## **7.2 Literature review and theoretical framework**

Even if much of the literature on the optimization of transportation problems deals with single period deterministic models, a number of surveys dealing partially or exclusively with dynamic and/or stochastic models can be mentioned: [Psaraftis, 1995], [Savelsbergh & Sol, 1995], [Crainic & Laporte, 1997], [Parragh et al., 2008], [Berbeglia et al., 2010], [Pillac et al., 2013]). In the earliest reviews, models simultaneously dynamic and stochastic are mentioned as perspectives. Indeed, research

on dynamic stochastic models is quite sparse. In papers dealing with such problems, questions are often depicted in words, few formulations are proposed and solution techniques may only be sketched. Therefore, we focus here on a few references which seem to be directly relevant.

In its simplest deterministic version where the horizon is finite and where all loads are known with certainty, the Dynamic Vehicle Allocation problem has been studied for several decades, in particular in connection with the problem of repositioning empty vehicles (mostly, rail cars) in a transportation network; see, e.g., the surveys by [Dejax & Crainic, 1987] and [Crainic, 2003]. Several authors have observed that this version of the problem can be formulated as a min-cost network flow problem and hence, can be solved in polynomial time; see section 7.4 hereunder.

[Tjokroamidjojo et al., 2006] and [Zolfagharinia & Houghton, 2014] have used and solved this formulation in a deterministic rolling-horizon framework which follows the generic description provided in [Sethi & Sorger, 1991]. Their main objective was to examine the influence of the length of the rolling horizon, the load density, the trip length and the fleet size on the quality of the solutions obtained.

The deterministic model can also be viewed as a special case of the generic pick-up and delivery problems (PDP) discussed in [Savelsbergh & Sol, 1995], with a profit maximization objective as in [Maes et al., 2013]. However, our formulation only considers full truckloads, whereas the PDP literature usually allows less-than-full truckloads. Moreover, the DVA problem does not include any typical PDP side-constraints such as driver regulations, forcing the driver to come home, or time windows.

In its dynamic and stochastic version, the Dynamic Vehicle Allocation problem has been almost exclusively studied by Powell in a series of papers [Powell, 1986], [Powell, 1988], [Frantzeskakis & Powell, 1990], [Powell, 1996], [Powell et al., 2000], [Spivey & Powell, 2004], [Topaloglu & Powell, 2007], [Powell, 2011]. Crainic in [Crainic, 2003] provide a nice survey of these contributions. Powell and his coauthors propose several approaches, such as Approximate Dynamic Programming (ADP), for the solution of the DVAP and of its extensions to more abstract resource allocation models. Most of these approaches are based on solving a sequence of multistage (typically, two-stage) problems with recourse, where the recourse accounts for all future periods after the first one(s). As observed by [Crainic, 2003], the resulting models are increasingly complex. A main difficulty lies clearly in the determination of appropriate recourse value functions. In the applications considered by Powell, there are, on average, a few customers to be served (or trucks requested) in each node of the transportation network, and the number of trucks is very large with respect to the number of requests. This makes it possible to rely on concepts like “the marginal value of serving a request”, or “the marginal value of relocating a truck in a node”. Our focus, on the other hand, is on applications where

a small or medium-size carrier faces demand that may be scarce and geographically scattered. Purely combinatorial algorithms appear to be much better suited for this type of instances.

We are interested, also, in evaluating the relative performance of generic heuristic procedures, and in estimating the value of information in such settings. Evaluating the performance of multi-period or stochastic optimization algorithms is difficult, since the value of the optimal policy is usually not known. Therefore, researchers frequently rely on comparisons with (estimates of) lower bounds derived from myopic strategic, as in [Yang et al., 2004] or [Angelelli et al., 2009], or on a posteriori (optimistic) upper bounds, as in [Secomandi, 2000], [Mitrović Minić et al., 2004], or [Spivey & Powell, 2004]. It is unusual to find both myopic and a posteriori bounds in the same article; an example is [Tjokroamidjojo et al., 2006]. Authors in [Berbeglia et al., 2010] observe that, presently, there exists no standard performance analysis framework for dynamic stochastic problems (such as computing time or number of iterations for static and deterministic problems), and they stress the need to compute meaningful bounds.

### 7.3 Problem definition

A transportation company attempts to maximize its profit by delivering a selection of transportation requests, referenced as loads or orders, from a set  $L$  with a fleet of  $I$  trucks. Each load is transported individually (Full-Truck-Load FTL). The fleet is constant, homogeneous and limited. The loads are loaded and unloaded according to their pick-up and delivery periods  $a_j$  and  $b_j$  for  $j \in L$ : each load  $j$  is loaded at the beginning of the pick-up period  $a_j$  and unloaded at the end of the delivery period  $b_j$ . Periods  $a_j, b_j$  belong to the set of periods  $\{1, 2, \dots, WH\}$  where  $WH$  is assumed to be very large, as  $\{1, 2, \dots, WH\}$  stands for the (potentially endless) sequence of decision periods of the company. Loads (origins and destinations) and trucks are located in a restricted set  $C$  of nodes (say, cities) of a network. The position of truck  $i$  at time  $t$  is  $\gamma_i(t)$ . The departure and destination cities of load  $j$  are  $\alpha_j$  and  $\beta_j$ , respectively. The trip duration  $d_j$  for load  $j$  is equal to the number of periods needed to travel from  $\alpha_j$  to  $\beta_j$ . The distance between each pair of cities is expressed by an integer number of time units (i.e., periods). So, the trip duration  $d_j$  is equal to the distance between  $\alpha_j$  and  $\beta_j$ , denoted by  $d(\alpha_j, \beta_j)$ . Therefore, by definition, the following relation holds:  $a_j + d_j - 1 = b_j$ , reflecting the fact that preemption is forbidden.

We assume that the complete problem over  $WH$  periods is handled by solving a sequence of subproblems over rolling horizons of length  $H$ . The periods taken into account in each subproblem are  $\{t, \dots, t + H\}$  where  $t \in \{1, \dots, WH - H\}$  stands

for the current *decision period*. (The last decision period to be considered in the rolling horizon process is period  $WH - H$ .)

At decision period  $t$ , some trucks are unladen (available to be loaded) and some are already loaded (transporting an order). Loaded trucks are unavailable for a new load assignment, and will only become available in a subsequent period, after having delivered their current load. The loads for which  $a_j = t$  are available for loading. If the decision is made to load order  $j$ , then it will be assigned to an unladen truck  $i$  which must be present and available in location  $\alpha_j$  at time  $t$ .

When  $t$  is the current decision period, the information included in periods  $\{t, \dots, t + RH\}$  is fully revealed. So,  $RH$  is the number of periods that contain deterministic information about the availability of loads. Those loads such that  $a_j \in \{t + RH + 1, \dots, t + H\}$  are *projections*, or *forecasts*, meaning that the dispatcher is not sure whether the transportation orders will be confirmed or not by the clients. All decisions regarding loads with  $a_j$  in  $\{t + 1, \dots, t + H\}$  might be re-computed until  $a_j$  becomes the decision period. (See [Tjokroamidjojo et al., 2006] for details.) Also, waiting and unladen moves decisions are reevaluated at each successive decision period.

In summary, at time  $t$ , the dispatcher must decide for each unladen truck  $i$  positioned in node  $\gamma_i(t)$ :

- either to load on truck  $i$  any load  $j$  such that  $a_j = t$  and  $\gamma_i(t) = \alpha_j$ ; then, truck  $i$  starts moving towards node  $\beta_j$ , where it will arrive at time  $b_j$ ; truck  $i$  will be available again for loading at the start of period  $b_j + 1$ ;
- or to get truck  $i$  moving (unladen) towards another node  $c \in C$ ;
- or to keep truck  $i$  waiting at its current location:  $\gamma_i(t + 1) = \gamma_i(t)$  (this can also be viewed as a special case of the previous decision).

As there is a selection process, those loads with  $a_j = t$  that are not allocated to any truck in period  $t$  bring neither profit, nor penalty. As in [Maes et al., 2013], we assume that a competitor, working as a sub-contractor, can be activated for any single load during the decision period  $t$ , at no cost for the carrier. This implies that all plans are feasible and that no recourse or penalty costs are incurred due to past decisions. The total profit results from revenues earned for delivering loads and from costs generated by unladen trips or by waiting on site. Related parameters are:

- $g_j$  revenue per period for transporting load  $j$  (equal for all trucks),
- $e$  unladen trip cost per period (equal for all trucks between any pair of nodes),
- $f$  waiting cost per period (equal for all trucks in all locations).

In a mono-period or myopic setting, decisions might seem greedy. Therefore, no feasible order is ever discarded myopically if the fleet is sufficient and available, since the alternative options would consist in moving unladen or in waiting on site, and since revenues are preferred to costs. In the multi-period model, this conclusion does not necessary hold: it may be more profitable to wait or to move unladen in order to pick up a load that is projected to be released in a future period. In such cases, comparisons have to be made between various combinations of loading actions, waiting times and unladen trips. The combinatorial aspect of the decision making problem comes from this blend of feasible actions.

We assume that moving is more expensive than staying at standstill:  $e > f$ . Sometimes, the decision for a truck  $i$  in the current decision period  $t$  may be to move from its current location  $\gamma_i(t)$  towards a different city  $\alpha_j$ , with a view to loading an order  $j$  in a future period  $a_j = t + k$ . This might imply for the truck to move unladen, but also to spend some time waiting at destination or along the way, in case the travel time to  $\alpha_j$  is shorter than  $k$ . In such a case, we assume that the truck always prefers to wait at  $\gamma_i(t)$  rather than to move, so as to avoid potentially useless unladen trips (e.g., if the initial decision is modified after period  $t$ , the truck might have to go back to its initial position). The reverse option would lead to a repositioning strategy which is not in the scope of this research.

Moreover, still in order to avoid useless moves, we define a parameter  $D \leq RH$  such that unladen trips of length larger than  $D$  are not allowed. This avoids useless trips if a projected load  $j$  does not become available. Both restrictions to avoid useless trips are issued from practice as drivers do not appreciate to drive to be on-time in a place where an expected transportation order is canceled.

Each unladen trip between two cities (departure city  $\gamma_i(t)$ , arrival city  $\alpha_j$ ) consists of a sequence of cities along a path. This path is taken to be the shortest one (ties are arbitrarily broken). In fact, in a rolling horizon framework, only the first intermediate city  $c^1$  on this path, reached after 1 period, is important as the location of the truck is updated at the end of the decision period and a new decision is made in the next period.

The uncertainty of the forecasts, or stochastic availability of a specific load  $j$  is modeled by a Bernoulli distribution:

$$P(q_j = x) = \begin{cases} p_j & \text{if } x = 1 \\ 1 - p_j & \text{if } x = 0 \end{cases} \quad (7.1)$$

where  $q_j$  is a random variable which takes value 1 when order  $j$  is released (i.e., made available for transportation) in period  $a_j$ , and value 0 otherwise, and  $p_j$  is a parameter in  $[0, 1]$ . Our methodology is based on the simulation of scenarios. Generating a single scenario means allocating values to  $q_j$  for all loads  $j$  within

the unrevealed part of the horizon. In the rolling horizon procedure, the availability of load  $j$ , i.e., the value of  $q_j$  is deterministically revealed in period  $a_j - RH$ . In subsequent periods,  $q_j$  remains permanently fixed to that value. So, periods in the horizon  $\{t, \dots, t + RH\}$  contain exact deterministic information. Periods  $\{t + RH + 1, \dots, t + H\}$  (with  $RH < H$ ) contain forecasts, known only through  $p_j$ , for loads that might become available in the future. As a consequence, the combinatorial choice either to load or not to load an order in the deterministic part of the horizon can be influenced by information pertaining to the stochastic part of the horizon, which might not be confirmed a posteriori.

### 7.3.1 Rolling horizon procedure

As explained, in practice, not all information can be known in advance for the whole horizon  $WH$ . So, the problem is transposed to a rolling horizon setting. At each decision period  $t$ , the rolling procedure includes information about load  $j$  if the departure period  $a_j$  is included in the rolling horizon  $\{t, \dots, t + H\}$ . This implies that the set of loads under consideration keeps changing from decision period to decision period. This set is labeled  $J_H$  (the index  $t$  is omitted for short):  $J_H = \{j \in L : t \leq a_j \leq t + H\}$ .

Similarly for the set of trucks  $I$ , previous loading assignments of trucks must be taken into account at every time. A truck-load assignment is represented by a load parameter  $v_j$ : if truck  $i$  is assigned to load  $j$ , then  $v_j = i$ . The truck  $i$  is allocated to load  $j$  temporarily, from loading to unloading. After unloading load  $j$ , the assignment parameter is reset ( $v_j = 0$ ), the current truck location is adapted and the truck status is “free for loading”.

At each period  $t$ , the dynamics of the system is due to:

- New loads  $j$  such that  $a_j = t + H$  entering the rolling horizon.
- Loads  $j$  such that  $a_j = t + RH$  entering the fully revealed horizon (if  $q_j = 0$ , then load  $j$  can be removed from  $J_H$ ).
- Loads such that  $a_j < t$  being removed from  $J_H$ .
- The actions of truck  $i$ :
  1. either to load some load  $j$ ; then,  $v_j = i$ ;
  2. or to move unladen towards  $\alpha_j$ , the loading city of load  $j$ ; then, at the end of period  $t$ , truck  $i$  is repositioned in the first city  $c^1$  on the path from  $\gamma_i(t)$  to  $\alpha_j$ ;
  3. or to wait in city  $\gamma_i(t)$ ;

4. or to unload load  $j$  as a consequence of a previous decision if  $b_j = t$  and  $v_j = i$ ; recall that the destination  $\beta_j$  is reached at the end of the period and the truck is unloaded in the same period; so, the assignment parameter  $v_j$  is reset to 0 before the next period.

The expected profit  $\mu(\pi)$  per period for a decision policy  $\pi$  over the entire horizon is defined as:

$$\mu(\pi) = \lim_{WH \rightarrow \infty} \frac{E[\sum_{t=0}^{WH-H} C(S_t, A^\pi(S_t))]}{WH - H} \quad (7.2)$$

where  $S_t$  denotes the state of the system at time  $t$  (defined by the collection of loads available for shipment at time  $t$ , additional information over the horizon  $\{t + 1, \dots, t + H\}$ , the truck locations and their status - loaded or not),  $A^\pi(S_t)$  denotes the actions taken in state  $S_t$  according to policy  $\pi$ , and  $C(S_t, A^\pi(S_t))$  is the associated profit collected in period  $t$  (revenues minus costs incurred at  $t$ ). The objective is to maximize  $\mu(\pi)$  over all feasible policies  $\pi$ .

### 7.3.2 Time-space example

For a better understanding, let us consider a single-scenario deterministic example (see Figure 7.1) where the current period is  $t = 1$  and there are

- $H = 4$  periods
- $D = 1$  period
- $RH = 1$  period
- 6 cities  $C_1, C_2, C_3, C_4, C_5, C_6$

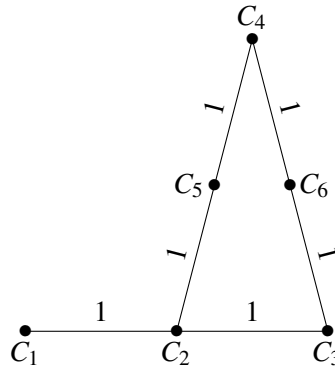


Figure 7.1: Space Graph

- 4 loads to be transported
  1. load  $L1$  from city  $C_4$  to  $C_3$  during periods 0 and 1 ( $a_1 = 0, b_1 = 1$ )
  2. load  $L2$  from city  $C_1$  to  $C_2$  during period 2 ( $a_2 = b_2 = 2$ )
  3. load  $L3$  from city  $C_2$  to  $C_4$  during periods 3 and 4 ( $a_3 = 3, b_3 = 4$ )
  4. load  $L4$  from city  $C_2$  to  $C_3$  during period 5 ( $a_4 = b_4 = 5$ )
- 2 trucks at the beginning of period 1
  1. truck 1 located in city  $C_1$  and idle
  2. truck 2 located in city  $C_6$  and currently on the move from  $C_4$  to  $C_3$ , loaded with  $L1$ .

For the revenue parameters, we assume that  $g_j = g$  for all  $j$ , and that  $e, f$  are arbitrary. It is easy to find manually two equivalent optimal solutions. They are represented by the time-space graphs below (Figure 7.2 and Figure 7.3, where bold solid lines are associated with the loads).

In solution 1 (see Figure 7.2), truck 1 is in standby for 1 period in  $C_1$ . During period 2, truck 1 transports load  $L2$  from  $C_1$  to  $C_2$ , then waits in  $C_2$  for 2 periods and then transports load  $L4$  from  $C_2$  to  $C_3$ . Truck 2 continues to transport load  $L1$  from  $C_4$  to  $C_3$  for one period and drives through city  $C_6$  along the way. Then in period 2, truck 2 moves unladen from  $C_3$  to  $C_2$ , picks up load  $L3$ , transports it from  $C_2$  to  $C_4$ , and finally waits in  $C_4$  for one period.

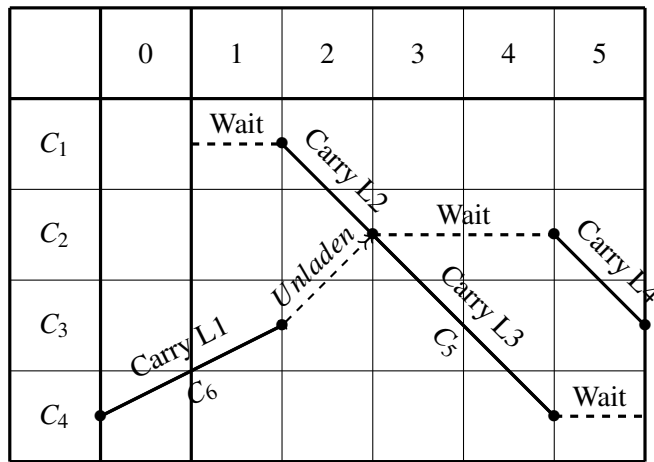


Figure 7.2: Optimal solution 1 in a time-space graph



In solution 2 (see Figure 7.3), truck 1 is in standby for one period in  $C_1$ . During period 2, it transports load  $L2$  from  $C_1$  to  $C_2$ , then transports load  $L3$  from  $C_2$  to  $C_4$  and waits in  $C_4$  for one period. Truck 2 continues to transport load  $L1$  from  $C_4$  to  $C_3$  for one period. Then, it waits in  $C_3$  for two periods, and subsequently moves unladen from  $C_3$  to  $C_2$  during period 4 to be on time to transport load  $L4$  from  $C_2$  to  $C_3$ .

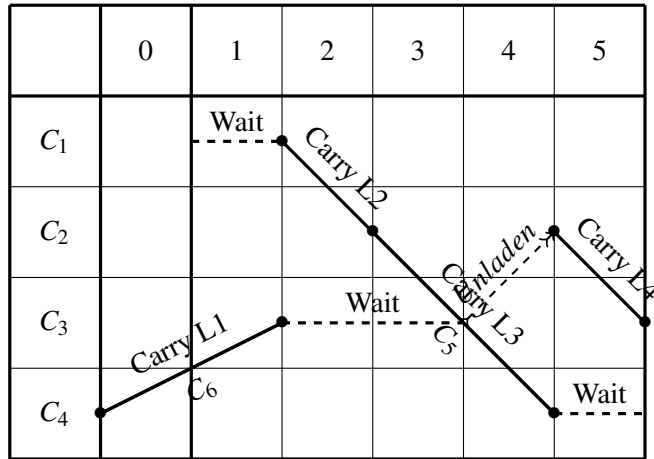


Figure 7.3: Optimal solution 2 in a time-space graph

In both solutions, the profit is equal to the revenues minus the costs :  $5g - e - 4f$ .

The following facts can be observed from this simple example:

1. The optimal solution might not be unique because of the combinatorial situation but also because trucks are identical. Supposing that two trucks are simultaneously in the same city, it is possible to exchange by symmetry the truck locations at the end of the horizon.
2. The rule stating that a truck has to wait before moving unladen might influence the final policy. In fact, when the horizon unrolls, the next starting position of the truck depends on whether the truck has waited or moved.
3. Due to temporal or spatial constraints, some connections from city to city are feasible and others are not. Pre-processing the data can be used to reduce the problem size and the computing time.

## 7.4 Formulation

### 7.4.1 Scenario-based approximate deterministic models

As mentioned previously, for each decision period  $t$ , the loads  $j$  taken into account have  $a_j$  contained in the rolling horizon  $\{t, \dots, t+H\}$ . If  $a_j \leq t+RH$ , the information is deterministic, while in  $\{t+RH+1, \dots, t+H\}$  the load information, its availability, is stochastic with a discrete Bernoulli probability distribution. Consequently, a deterministic equivalent representation could be obtained by a complete scenario tree. The scenario tree is made of all possible scenarios that might arise in the rolling horizon. Unfortunately, due to the huge number of potential scenarios, this would lead to intractable instances as far as computing time and memory are concerned.

Therefore, we will have to resort to heuristics using a deterministic but reduced representation of the future. These heuristics may be based on single or multiple scenarios, which may be viewed as independent of each other or as forming a sub-tree representation of the problem. Each scenario transforms the stochastic information into deterministic information for the uncertain part of the horizon, and leads to an approximate deterministic model. To generate a random scenario, we use a Monte-Carlo simulation based on the probability distribution of the availability variable  $q_j$ , for each load  $j$  with  $a_j \in \{t+RH+1, \dots, t+H\}$ . This process leads to the occurrence or to the absence of load  $j$  in the sets of loads to be considered for this scenario, if  $q_j = 1$  or  $0$ , respectively. The restricted subset of loads  $J$ ,  $J \subseteq J_H \subseteq L$ , only includes load  $j$  if  $q_j = 1$  in the scenario. Note again that all scenarios include the same set of loads in the deterministic part of the horizon  $\{t, \dots, t+RH\}$ , but scenarios are differentiated by the loads that become available in  $\{t+RH+1, \dots, t+H\}$ .

### 7.4.2 Formulation for a mono-scenario model

We now provide a formulation of the mono-scenario deterministic model (similar to, e.g., [Tjokroamidjojo et al., 2006]).

In order to set up the model, two new families of parameters are introduced and denoted by  $TL$  (for Truck to Load) and  $LL$  (for Load to Load) and:  $TL(i, j) = 1$  if city  $\alpha_j$  can be reached from the current location of truck  $i$ , namely  $\gamma_i(t)$ , before the loading period  $a_j$ ; and  $LL(j, k) = 1$  if city  $\alpha_k$  can be reached before period  $a_k$  after unloading load  $j$  at city  $\beta_j$ . More precisely,

$$\begin{aligned} TL(i, j) &= 1 \text{ if } d(\gamma_i(t), \alpha_j) \leq a_j - t \text{ and } d(\gamma_i(t), \alpha_j) \leq D \\ &= 0 \text{ otherwise,} \end{aligned} \tag{7.3}$$

$$\begin{aligned}
LL(j, k) &= 1 \text{ if } d(\beta_j, \alpha_k) \leq a_k - b_j - 1 \text{ and } d(\beta_j, \alpha_k) \leq D \\
&= 0 \text{ otherwise.}
\end{aligned} \tag{7.4}$$

These parameters provide information on feasible connections and will reduce the decision space.

Moreover, a decision concerning a truck might be to wait in its present location for at least one period. Therefore, some nonnegative parameters related to waiting times are introduced, based on trucks and loads information:  $w_{ij}^0$  is the waiting time for truck  $i$  to reach load  $j$  at period  $a_j$ ;  $w_{jk}$  is the waiting time between unloading load  $j$  and loading load  $k$ ;  $w_j^H$  is the elapsed time after delivering load  $j$  until the end of the horizon; and  $w_i^{0H}$  is the waiting time for a truck staying at a standstill over the full horizon.

$$\begin{aligned}
w_{ij}^0 &= a_j - d(\gamma_i(t), \alpha_j) - t \text{ if } TL(i, j) = 1 \\
&= 0 \text{ otherwise}
\end{aligned} \tag{7.5}$$

$$\begin{aligned}
w_{jk} &= a_k - b_j - d(\beta_j, \alpha_k) - 1 \text{ if } LL(j, k) = 1 \\
&= 0 \text{ otherwise}
\end{aligned} \tag{7.6}$$

$$w_j^H = \max(0, t + H - b_j), \tag{7.7}$$

$$w_i^{0H} = 1 + H. \tag{7.8}$$

In a first step, using these parameters leads to a representation based on a time-space graph. The decision variables introduced hereafter represent either the arcs followed by the trucks from city to city (unladen or loaded) or waiting decisions. The connectivity parameters  $TL(i, j)$  and  $LL(j, k)$  provide two benefits by making it possible to:

- remove useless variables associated with infeasible links (e.g., concomitant projected orders),
- switch from a "multi-period" to a "timeless" model as temporal constraints are embedded inside the feasible arcs corresponding to  $TL(i, j) = 1$  and also  $LL(j, k) = 1$  (see below).

In fact, the two dimensional time-space graph can be reduced to a simple oriented graph of feasible links between feasible locations for trucks (see [Desrosiers et al., 1995]). The single dimension "feasibility" of an action is represented by an arc simultaneously replacing the time and space constraints. These arcs exist between to feasible locations for trucks along time. Notice that, as there are no unladen moves for repositioning due to a proactive waiting strategy, feasible locations for a truck along time are related either to its present location for waiting or to the

departure or arrival city of a load. Finally, neither temporal index ( $t$  for time or period) nor positions in the space are required. This allows the multi-period problem to be formulated as a network flow model.

If there are  $I$  trucks and a set  $J$  of loads in the deterministic problem ( $J \subseteq J_H \subseteq L$ ), the oriented graph  $G = (V, E)$  is as follows (see figure 7.4 for an illustration). The set  $V$  consists of vertices  $\{P_i, U_j, V_j, T\}$  for  $i = 1, \dots, I$  and  $j \in J$  where:

1.  $P_i$  is associated with (the initial location of) unladen truck  $i$  at time  $t$ ,
2.  $U_j$  is associated with the pick-up point and time of load  $j$ ,  $U_j = (\alpha_j, a_j)$ ,
3.  $V_j$  is associated with the delivery point and time of load  $j$ ,  $V_j = (\beta_j, b_j)$ ,
4.  $T$  is a sink node associated with the end of the rolling horizon  $t + H$ .

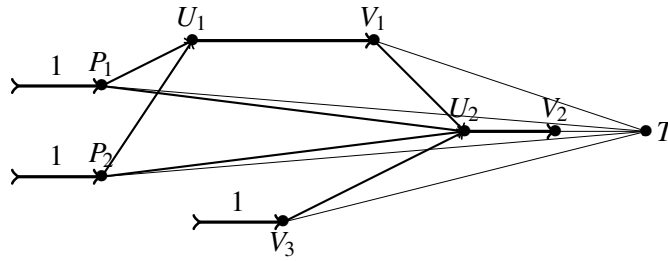


Figure 7.4: Network flow: e.g. 3 trucks (1 loaded) and 2 potential loads

### Arcs and Variables

The set of arcs  $E$  represents the set of feasible links, which can be divided into several categories. Each arc is associated with a single binary variable (here,  $v_j$  denotes the initial assignment of load  $j$ ):

- $PU_{ij}$  with arc  $(P_i, U_j)$  if  $TL(i, j) = 1$  and  $v_j = 0$ , for  $i = 1, \dots, I$  and for  $j \in J$ ;
- $UV_j$  with arc  $(U_j, V_j)$  if  $v_j = 0$ , for  $j \in J$ ;
- $VU_{jk}$  with arc  $(V_j, U_k)$  if  $LL(j, k) = 1$  and  $v_k = 0$ , for  $j, k \in J$  and  $j \neq k$ ;
- $VT_j$  with arc  $(V_j, T)$ , for  $j \in J$
- $PT_i$  with arc  $(P_i, T)$  for  $i = 1, \dots, I$  such that, for all  $j \in J$ ,  $i \neq v_j$ .

We refer to figure 7.4 for an illustration of these arcs.

By analogy with the time-space diagrams of Figure 7.2 and Figure 7.3,

- $PU_{ij}$  represents the decision for an unladen truck to pick-up load  $j$  for its first loaded trip (this might mean to load  $j$  in the present truck location if  $\gamma_i(t) = \alpha_j$  and  $a_j = t$ , or to move unladen in order to pick-up load  $j$  in city  $\alpha_j$  if  $\gamma_i(t) \neq \alpha_j$  and  $a_j > t$ ; in both cases, truck  $i$  may have to wait on site for a few periods);
- $UV_j$  represents the decision to transport load  $j$ ;
- $VU_{jk}$  represents the decision to travel between the destination city of load  $j$ , and the departure city of load  $k$ ;
- $VT_j$  represents the decision to wait on site until the end of the horizon after delivering  $j$ ;
- $PT_i$  represents the decision for a truck  $i$  to stay idle in city  $\gamma_i(t)$  over the rolling horizon

#### Arc capacities

The arcs represent feasible trips for trucks in an aggregated time-space dimension. Each load is transported individually (Full-Truck-Load). So, there can be only one truck per arc representing an action in time and place. Therefore, the maximum (indivisible) capacity is fixed to 1 on all arcs.

#### Arc costs

Let us now turn to a definition of the cost associated with each arc. At time  $t$ , some trucks are already busy transporting a load. Such loaded trucks continue their trip during the current decision period and we assume that their cost has already been taken into account in previous periods.

Now, the variables affecting an unloaded truck  $i$  during the current period are  $PU_{ij}$ ,  $UV_j$  and  $PT_i$ . If  $PU_{ij} = 1$  (and, by the flow conservation constraint,  $UV_j = 1$ ), this can lead to the following actions:

- To load order  $j$  on truck  $i$  if  $\gamma_i(t) = \alpha_j$  and  $a_j = t$  meaning that the truck and the load are in the same city and that the load should be loaded without waiting. In that case, the revenue generated by transporting the load is added to the total value of the solution.
- To move unladen in order to reach load  $j$  if  $\gamma_i(t) \neq \alpha_j$  and  $a_j = t + d(\gamma_i(t), \alpha_j)$ . In this case, the cost of the unladen move is subtracted from the solution value.
- To wait for one or several periods, either if  $\gamma_i(t) \neq \alpha_j$  and  $a_j > t + d(\gamma_i(t), \alpha_j)$  (i.e., the truck waits at least one period before it starts moving), or if  $\gamma_i(t) = \alpha_j$

and  $a_j > t$ , (i.e., the truck waits before it picks up load  $j$ ). In this case, the cost of waiting is subtracted from the solution value.

The second type of decisions is when either  $VT_j = 1$  or  $PT_i = 1$ . They imply to wait until the end of the rolling horizon and to incur the associated cost.

In summary, the arc costs are defined as follows:

- Departure arc to a load:  $C_{PU_{ij}}$  as cost of arc  $(P_i, U_j)$ :  
unladen distance + waiting time =  $-(d(\gamma_i(t), \alpha_j) * e + f * w_{ij}^0)$   
for  $i = 1, \dots, I$  and for  $j \in J$ , if  $TL(i, j) = 1$  and if  $v_j = 0$ .
- Loaded arc:  $C_{UV_j}$  as cost of arc  $(U_j, V_j)$ :  
revenue linked to the distance =  $+d(\alpha_j, \beta_j) * g_j$  for  $j \in J$ , if  $v_j = 0$ .
- Intermediate arc between two loads:  $C_{VU_{jk}}$  as cost of arc  $(V_j, U_k)$ :  
unladen distance + waiting time =  $-(d(\beta_j, \alpha_k) * e + f * w_{jk})$   
for  $j, k \in J$  and  $j \neq k$ , if  $LL(j, k) = 1$  and if  $v_k = 0$ .
- Final arc:  $C_{VT_j}$  as cost of arc  $(V_j, T)$ : waiting time =  $-(f * w_j^H)$  for  $j \in J$ .
- Idle truck arc:  $C_{PT_i}$  as cost of arc  $(P_i, T)$ :  
waiting time over the rolling horizon =  $-f * w_i^{0H}$   
for  $i = 1, \dots, I$  such that, for all  $j \in J$ ,  $i \neq v_j$ .

### Objective function

With the above notations, the objective function can be expressed as

$$\begin{aligned} \max Z = & \sum_{j \in J} \left[ C_{VT_j} * VT_j + C_{UV_j} * UV_j + \sum_{i=1, \dots, I} (C_{PU_{ij}} * PU_{ij}) + \sum_{k \in J; k \neq j} (C_{VU_{jk}} * VU_{jk}) \right] \\ & + \sum_{i=1, \dots, I} C_{PT_i} * PT_i \end{aligned} \quad (7.9)$$

The maximization of  $Z$  is subject to the following constraints:

### Flow conservation for unladen trucks and unloaded orders

$$\text{Node } P_i : \sum_{j \in J} PU_{ij} + PT_i = 1, \text{ for each unladen truck } i = 1, \dots, I \quad (7.10)$$

$$\text{Node } U_j : \sum_{i=1, \dots, I} PU_{ij} + \sum_{k \in J; k \neq j} VU_{kj} = UV_j, \text{ for each unloaded } j \in J \quad (7.11)$$

$$\text{Node } V_j : UV_j = VT_j + \sum_{k \in J; k \neq j} VU_{jk}, \text{ for each unloaded } j \in J \quad (7.12)$$

### Flow conservation for loaded orders

$$\text{Node } V_j : 1 = VT_j + \sum_{k \in J; k \neq j} VU_{jk}, \text{ for each loaded } j \in J \quad (7.13)$$

### Required flow constraint

$$\text{Node } T : I = \sum_{j \in J} VT_j + \sum_{i=1, \dots, I} PT_i \quad (7.14)$$

Constraints (7.10) express that each unloaded truck  $i$  either moves to reach a load or stays idle. Constraints (7.11) ensure that if a load  $j$  is selected, it is because this load has been reached by an idle truck at the beginning or after the unloading of another load  $k$ . Constraints (7.12) show that if load  $j$  is selected on the path, after unloading  $j$  either the truck reaches another load  $k$  or stops. Moreover, if load  $j$  is not selected, none of these actions can take place. While if truck  $i$  is loaded with  $j$  initially, constraints (7.13) imply performing one of the previous actions after the delivery of load  $j$ .

As arc capacities are equal to one and indivisible, decision variables are binary. Thus, the capacity constraints on each arc can be removed. In fact, they are inserted in the flow conservation constraints (7.10) to (7.13). Consequently, the usual source node is not required in the graph because the flow conservation constraints impose a departure for each truck  $i$  either from its initial city  $\gamma_i(t)$  represented by vertex  $P_i$  if truck  $i$  is unladen, or from  $V_j$  representing  $\beta_j$  and  $b_j$  if the truck has been previously loaded with load  $j$ . So, an initial flow conservation constraint from a source node would be redundant.

Finally, constraint (7.14) imposes that  $I$  trucks reach the sink node  $T$ . So, the required total flow is equal to the number of trucks  $I$  originally loaded or unladen.

### 7.4.3 Complexity for a single scenario formulation

This network flow formulation is polynomially solvable (in our experiments, we simply rely on the LP solver of IBM ILOG Cplex 12 to provide an optimal integer solution). It may be worth noting that polynomial solvability is due to some implicit restrictive assumptions of our model. Namely, the ending point at the end of the horizon is not imposed to any truck; no intermediate city has to be mandatorily visited (e.g., for refueling); no specific load has to be mandatorily transported; no waiting periods are imposed for maintenance or to comply with driving regulations; and so forth. Adding such constraints would make the problem theoretically harder to solve (NP-hard), as established by [Keskinocak & Tayur, 1998].

### 7.4.4 Formulation for a deterministic subtree model

From the previous formulation which is valid for a single representation of the future (i.e., scenario), it is easy to move to a formulation associated with a subtree that represents a finite set of  $ST$  possible outcomes of the probability distribution.

Scenarios differ because the outputs of the random generation process might differ for load  $j$  with  $a_j \in \{t+RH+1, \dots, t+H\}$ . For load  $j$ , the stochastic parameter outcome  $q_j$  for scenario  $s = 1, \dots, ST$  is labeled  $q_j^s$ . Considering all possible scenarios would yield an exact deterministic equivalent representation of the stochastic problem, whereas any subtree of scenarios leads to an approximate formulation and hence, to a heuristic solution of the problem. The approximation is due to two factors. First, obviously, the subtree does not contain all scenarios. But also, we do not account in the formulation for the relative weight (probability of occurrence) of each scenario within the subtree: each scenario weight is set equal to one. In the deterministic equivalent representation, each scenario weight in the tree would be equal to the probability  $W_s$  of realization of this scenario  $s$ :

$$W_s = \prod_{a_j \in \{t+RH+1, \dots, t+H\}; q_j^s=1} p_j * \prod_{a_j \in \{t+RH+1, \dots, t+H\}; q_j^s=0} (1 - p_j) \quad (7.15)$$

Disregarding the weights appears to provide a reasonable approximation when the sample of scenarios in the subtree is very small with respect to the total number of scenarios (as in our practical applications). This should be investigated in further tests. To generate the subtree formulation in a natural way, the deterministic formulation of section 7.4.2 is replicated  $ST$  times and scenarios are indexed by  $s$ . In practice for the implementation, all variables get an additional index  $s \in \{1, \dots, ST\}$  and the set of loads under consideration,  $J_H$ , includes all loads in the rolling horizon. So, the graph introduced in section 7.4.2 is replicated  $ST$  times except for the sink node  $T$  which remains unique. Moreover, some variables are useless for specific scenarios when  $q_j^s = 0$ .

In fact, these arcs do not exist in the model, so that they can be removed in a preprocessing phase or the related variables can be set to zero (see equations 7.22 to 7.26).

#### Subtree variables

- $PU_{ij}^s \in \{0, 1\}$  for  $s = 1, \dots, ST$ , if  $TL(i, j) = 1$  and  $v_j = 0$ , for  $i = 1, \dots, I$  and for  $j \in J_H$
- $UV_j^s \in \{0, 1\}$  for  $s = 1, \dots, ST$ , if  $v_j = 0$ , for  $j \in J_H$
- $VU_{jk}^s \in \{0, 1\}$  for  $s = 1, \dots, ST$ , if  $LL(j, k) = 1$  and  $v_k = 0$ , for  $j, k \in J_H$  and  $j \neq k$
- $VT_j^s \in \{0, 1\}$  for  $s = 1, \dots, ST$ , for  $j \in J_H$
- $PT_i^s \in \{0, 1\}$  for  $s = 1, \dots, ST$ , for  $i = 1, \dots, I$  such that, for all  $j \in J_H$ ,  $i \neq v_j$

The subtree formulation is a copy of the single scenario formulation (equations (7.9) to (7.14)) indexed by  $s$ .



**Objective function:**

$$\begin{aligned} \max Z = & \sum_{s=1, \dots, ST} \left[ \sum_{j \in J_H} \left[ (C_{VT_j} * VT_j^s + C_{UV_j} * UV_j^s + \sum_{i=1, \dots, I} (C_{PU_{ij}} * PU_{ij}^s)) \right. \right. \\ & \left. \left. + \sum_{k \in J_H; k \neq j} (C_{VU_{jk}} * VU_{jk}^s) \right] + \sum_{i=1, \dots, I} C_{PT_i} * PT_i^s \right] \end{aligned} \quad (7.16)$$

subject to the following constraints:

**Flow conservation for unladen trucks and unloaded orders**

$$\text{Node } P_i^s : \sum_{j \in J_H} PU_{ij}^s + PT_i^s = 1 \quad (7.17)$$

for  $s = 1, \dots, ST$ , and for each unladen truck  $i = 1, \dots, I$

$$\text{Node } U_j^s : \sum_{i=1, \dots, I} PU_{ij}^s + \sum_{k \in J_H; k \neq j} VU_{kj}^s = UV_j^s \quad (7.18)$$

for  $s = 1, \dots, ST$ , and for each unloaded  $j \in J_H$

$$\text{Node } V_j^s : UV_j^s = VT_j^s + \sum_{k \in J_H; k \neq j} VU_{jk}^s \quad (7.19)$$

for  $s = 1, \dots, ST$ , and for each unloaded  $j \in J_H$

**Flow conservation for loaded orders**

$$\text{Node } V_j^s : 1 = VT_j^s + \sum_{k \in J_H; k \neq j} VU_{jk}^s \quad (7.20)$$

for  $s = 1, \dots, ST$ , and for each loaded  $j \in J_H$

**Required flow**

$$\text{Node } T : I = \sum_{j \in J_H} VT_j^s + \sum_{i=1, \dots, I} PT_i^s \quad (7.21)$$

for  $s = 1, \dots, ST$

Due to the value assumed by some of the random variables, (i.e.,  $q_j^s = 0$ ), some decision variables could not appear in the model or equivalently could be implemented through the following constraints.

**Constraints for missing arcs**

$$PU_{ij}^s = 0, \text{ for } s = 1, \dots, ST \text{ and for } i = 1, \dots, I \text{ and for } j \in J_H \text{ and if } q_j^s = 0 \quad (7.22)$$

$$UV_j^s = 0, \text{ for } s = 1, \dots, ST \text{ and for } j \in J_H \text{ and if } q_j^s = 0 \quad (7.23)$$

$$VU_{jk}^s = 0, \text{ for } s = 1, \dots, ST \text{ and for } j, k \in J_H \text{ and } k \neq j \text{ and if } q_j^s = 0 \quad (7.24)$$

$$VU_{kj}^s = 0, \text{ for } s = 1, \dots, ST \text{ and for } j, k \in J_H \text{ and } k \neq j \text{ and if } q_j^s = 0 \quad (7.25)$$

$$VT_j^s = 0, \text{ for } s = 1, \dots, ST \text{ and for } j \in J_H \text{ and if } q_j^s = 0 \quad (7.26)$$

As a unique set of actions will be performed, some constraints must be added to ensure the consistency of the solutions over the different scenarios. They are called non-anticipativity constraints. In the specific case of a rolling horizon, their role is only to ensure the consistency of actions taken over the current decision period  $t$ . Over the next decision step, in period  $t + 1$ , new actions will be taken and all previous projected ones rescheduled. Hence, fixing common actions within scenarios for periods  $\{t + 1, \dots, t + H\}$  is not mandatory.

So, if a loading is performed in the decision period  $t$  or if a truck  $i$  moves unladen, or else waits in its present location, the same action must be performed in all scenarios. For loading operations of truck  $i$  taking place in period  $t$  (see equation 7.27), replicated decision variables for loading  $j$  and moving unladen are forced to be strictly equal in all scenarios  $s = 1, \dots, ST$ .

#### **Non-anticipativity constraints for period $t$**

For an unladen truck  $i$  in period  $t$ :

$$\begin{aligned} PU_{ij}^1 - PU_{ij}^s &= 0, \text{ for } i = 1, \dots, I, j \in J_H, s = 2, \dots, ST, \\ &\text{if } TL(i, j) = 1, \text{ if } v_j = 0, \text{ and if } (t + d(\gamma(t), \alpha_j) = a_j \leq t + D) \end{aligned} \quad (7.27)$$

If the decision leads neither to load an order, nor to move unladen, the action for this truck is to wait. This action can be issued from decisions variables implying to wait:

- over the whole horizon,
- before loading in a forthcoming period in the present location,
- before moving unladen to another city.

There is no need to explicitly express the equality among these waiting actions in the scenarios of the subtree since they lead to unambiguous actions in the current period. Therefore, a waiting action for a truck can be the aggregation over  $ST$  scenarios of various decisions variables:  $PT_i^s$  or  $PU_{ij}^s$  with  $w_{ij}^0 > 0$ .

After optimizing the problem, actions for the decision period  $t$  have to be extracted to generate the value of the policy for this decision period (see section 7.4.2). Given that loading operations, unladen movements and waiting decisions are consistent in period  $t$ , analyzing one single scenario's variables (e.g., for  $s = 1$ ) is enough

to recover the reward of the policy during this decision period. Consequently, at each decision step, the policy evaluation is similar to what is done for the single scenario representation. Practically, analyzing for each unloaded truck  $i$  whether  $PU_{ij}^1 = 1$ , or  $PT_i^1 = 1$  is enough to deduce the cost or reward of truck  $i$  during period  $t$  for all scenarios of the subtree.

#### 7.4.5 Complexity for a subtree of scenarios

We mentioned earlier that the problem is polynomially solvable for a single scenario if no tasks are preassigned (see section 7.4.3). As non-anticipativity constraints are added, the network flow problem becomes an equal flow problem for some variables (see equation 7.27). Equal flow problems are NP-hard, except in specific cases (see [Meyers & Schulz, 2009]). It might still be possible to take advantage of the network flow structure to develop a specific algorithm but it is beyond the scope of our research. In practice, we observed that the commercial solver IBM ILOG Cplex 12 (using the simplex network solver or the default one) still provides an optimal integer solution equal to the linear relaxation solution of the problem. Moreover, even if the size of the model increases with a factor  $ST$ , we also observed that the computing time does not increase exponentially but rather linearly. For the instances that we handled, it remained within a few seconds, which is practically convenient. This is an important feature of our model since usually, in transportation, subtree formulations including integer variables are computationally intractable. Therefore, we did not develop a specific algorithm to solve this subtree formulation.

### 7.5 Algorithms

In a rolling horizon framework, the sequence of decisions implemented in successive periods results from solving a sequence of optimization models using deterministic and stochastic information over a sequence of restricted horizons of size  $H$ . The sequence of decisions associated with solutions generated by an algorithm  $y$  build up a policy  $\pi(y)$ , where  $y \in Y$  and  $Y$  is the set of algorithms. The value of the policy is denoted  $\mu_{\pi(y)}$ . The aim is to select the best policy  $\pi^* = \pi(y^*)$  maximizing the profit. The value of the best policy selected is  $\mu_{\pi^*} = \max(\mu_{\pi(y)})$  for  $y \in Y$ .

In general, the applied policy suffers from two drawbacks when compared to the single optimal solution that could be computed if we had full deterministic knowledge of the complete scenario over a fully revealed horizon of size  $WH$ . Firstly, it results from a sequential process implementing, at each decision period, actions issued from optimal solution over rolling sub-horizons of length  $H$ ; but such a sequence of optimal short-term solutions does not build the optimal long-term one. Secondly, each rolling horizon solution is based on stochastic information and not

on the fully revealed deterministic information over the rolling horizon. So, the policy might not be optimal even in a short-term perspective.

We denote by  $O^*$  (for Oracle) the value of the (a posteriori) optimal solution over the fully revealed complete horizon of length  $WH$ . This value provides an upper bound over the best possible attainable value:  $\mu_{\pi^*} \leq O^*$ .

Similarly, the value, denoted  $O^*(H)$ , of the policy obtained by solving a sequence of deterministic problems over fully revealed sub-horizons of size  $H$  can be estimated. This value provides a very optimistic estimate of the expected value of policy  $\pi^*$ . Formally, it cannot be proved that this policy issued from the repeated implementation over the whole horizon of the optimal solutions over sub-horizons will provide an upper bound on the optimal value of the stochastic problem (i.e., we cannot be sure that  $\mu_{\pi^*} \leq O^*(H)$ , always). But in practice it is empirically the case except in some pathological instances (as in [Blackburn & Millen, 1980]). Moreover, the difference between  $O^*$  and  $O^*(H)$  gives information on the loss due to the limited size  $H$  of the rolling horizon usually imposed by the booking process and by the forecasting possibilities.

Finally, we consider the value of the myopic policy derived by solving a sequence of deterministic problems over sub-horizons of size  $RH$  (which contain fully revealed information, by definition). This value  $O^*(RH)$  provides an empirical lower bound on the best policy value  $\mu_{\pi^*}$ . Indeed,  $\mu_{\pi^*}$  takes into account the same deterministic information as  $O^*(RH)$ , as well as additional stochastic information over the rolling horizon. If  $\mu_{\pi^*} < O^*(RH)$ , this means that the extra-piece of stochastic information and the algorithm developed to deal with it, are of no use.

Thus, in summary, the objective of our algorithms is to compute the best policy,  $\pi^*$ , such that:

$$O^*(RH) \lesssim \mu_{\pi^*} \lesssim O^*(H) \leq O^*. \quad (7.28)$$

### 7.5.1 Statistical validation of policy performance

As the aim is to evaluate the expected values  $\mu_{\pi(y)}$  of several policies  $\pi(y)$ , each optimization process should ideally be performed for each potential realization of the stochastic parameters, i.e., for each potential scenario. In the present case, the number of scenarios is finite as the stochastic variables can only take binary values. Yet, this number is so huge that the expected value of each policy  $\mu_{\pi(y)}$  can only be approximated on a restricted subset of scenarios. If  $F$  is a random sample of test scenarios (see chapter 3), this leads to an estimate of  $\mu_{\pi(y)}$  labeled  $\mu'_{\pi(y)}$ :

$$\mu'_{\pi(y)} = (1/|F|) * \sum_{j \in F} C(y, j), \quad (7.29)$$

where  $C(y, j)$  is the value of the policy generated by algorithm  $y$  over scenario  $j$ .

The bounds presented in section 7.5 can be similarly approximated, leading to the estimates of  $O^*(RH)'$ ,  $O^*(H)'$  and  $O^*$ . So, inequalities above (7.28) can only be assessed statistically. They become:

$$O^*(RH)' \lesssim \mu'_{\pi^*} \lesssim O^*(H)' \leq O^* \quad (7.30)$$

For the sake of simplicity, the bounds are written like their estimates in the rest of the document, i.e.,  $O^*$  represents its estimate  $O^*$ , and so forth.

As small gaps between the estimated values  $\mu_{\pi(1)}$  and  $\mu_{\pi(2)}$  obtained for two distinct policies  $\pi(1)$  and  $\pi(2)$  might be due to random effects, their statistical significance must be assessed. For each fixed instance, we use the same set of test scenarios  $F$  in order to reduce the variance of the observed values between policies. After checking normality of the distribution of results (using a Shapiro-Wilk non-normality test), we can apply a paired-sample  $Z$ -test to compare the values  $\mu_{\pi(1)}$  and  $\mu_{\pi(2)}$ . Two-sided and one-sided tests are available. In the discussions of our results, only the second ones were used in order to test whether a policy outclasses another or not:

$$H_0 : \mu_{\pi(1)} = \mu_{\pi(2)} \text{ vs. } H_1 : \mu_{\pi(1)} < \mu_{\pi(2)} \text{ (or vs. } H_1 : \mu_{\pi(1)} > \mu_{\pi(2)}) \quad (7.31)$$

As the objective function is profit maximization, we say for short that policy  $\pi(1)$  *outclasses* policy  $\pi(2)$  on a given instance, if we can conclude that  $\mu_{\pi(1)} > \mu_{\pi(2)}$  with a predefined confidence level.

## 7.5.2 Value of the stochastic information

If  $O^*(RH)$  is approximatively equal to  $O^*(H)$ , i.e., if  $O^*(RH)$  is not statistically smaller than  $O^*(H)$ , it means that any reasonable algorithm  $y$  should generate a profit  $\mu_{\pi(y)}$  which is close to both the lower bound and the upper bound. Therefore, the stochastic information contained in the horizon  $\{t + RH + 1, \dots, t + H\}$  appears to be useless. Consequently, the optimization process can be based on the deterministic information contained in the horizon  $\{t, \dots, t + RH\}$ , and the policy issued from  $O^*(RH)$  approximates the optimal  $\mu_{\pi^*}$ .

Therefore, a preliminary check in this research framework is to analyze whether  $O^*(H)$  outclasses  $O^*(RH)$ . If it is not the case, investigating other optimization policies or algorithms might be useless for some instance classes. As a contribution, we stress the fact that a multi-period stochastic problem can be reduced to an approximate timeless deterministic scenario-based model which provides an efficient policy for the complete problem. For instance, when the value of the stochastic information is low. So, when one can prove that the stochastic information in the rolling horizon  $H$  is statistically useless, the a-priori myopic policy with value  $O^*(RH)$  is as efficient as the a-posteriori one over the rolling horizon, with value  $O^*(H)$ .

Finally, from these bounds and using a reduced set of scenarios  $F$  for validation, we can estimate the expected values of some concepts:

- **EVMPM**: Expected value of the multi-period model =  $O^*(H) - O^*(RH)$
- **EVPI**: Expected value of the perfect information =  $O^* - \mu_{\pi^*}$
- **EVAI**: Expected value of the accessible information =  $O^*(H) - \mu_{\pi^*}$
- **EVTI**: Expected value of the tail information =  $O^* - O^*(H)$

If these differences between estimates of the theoretical expectations are statistically significant, economically relevant and technically reducible, it might be interesting for a company to modify the information collection by:

1. investing in its information system or changing its processes so as to collect more deterministic data over a longer horizon  $RH$ ;
2. or changing the booking process or forecasting tool in order to increase  $H$ .

Instead of improving the data collection, another option resides in developing optimization techniques leading to the best policy  $\pi^*$ . These techniques are described hereafter and are the core of this article.

### 7.5.3 Single scenario algorithms

As described in section 7.4.2, our problem can be tackled with a formulation based on a single scenario, such as what is performed for  $O^*(RH)$  and  $O^*(H)$  where the perfect information is revealed either over the deterministically revealed part of the horizon or over the rolling horizon. Any single scenario in between these two, contains the deterministically revealed part of the horizon and one realization of what might happen on the stochastic part of the rolling horizon. This realization might be issued among other arbitrary choices from specific scenarios such as the three detailed hereafter : Optimistic, Modal and Expected value scenarios. The main advantage of these heuristic choices is that they are computationally efficient.

#### 7.5.3.1 Optimistic scenario algorithm: Opt

In this scenario, all loads  $j$  are supposed to become available for transportation, i.e., all  $q_j$  are considered equal to 1 for all  $a_j \in \{t + RH + 1, \dots, t + H\}$ . The policy issued from this scenario is called *Optimistic*. This scenario is at the opposite of the pessimistic scenario where all  $q_j$  are considered equal to 0 for  $a_j \in$

$\{t + RH + 1, \dots, t + H\}$  which yields the value  $O^*(RH)$ . These extreme cases represent scenarios wherein parameter values are correlated while they are in fact independent. It is clear that the optimistic scenario is extreme in the sense that its occurrence probability is low or not representative except if the stochastic availability is nearly deterministic. So, this scenario can highly distort the representativity of the information in case of low value of  $p_j$  within the probability distributions.

### 7.5.3.2 Modal value scenario algorithm: Mod

This scenario generates a policy based on the modal value of the probability distribution. At the opposite of the above optimistic scenario, it is supposed to reduce the scenario to interesting loads instead of the exhaustive set. So, if  $p_j \geq 0.5$ , the scenario includes  $j$  ( $q_j = 1$ ). Conversely, if  $p_j < 0.5$ , load  $j$  is not inserted in the scenario ( $q_j = 0$ ). As for the previous scenario, the drawback of this one is that it leads to two extremes decisions: discarding or keeping orders even if their availability might be close to 50% or if there are numerous loads  $j$  with a low  $p_j$ . Moreover, the probabilistic gain of each load is not taken into account but only its availability probability.

### 7.5.3.3 Expected value scenario algorithm: EG

To take the advantages and to avoid the drawbacks of previous scenarios, a policy can be derived from a scenario that includes all  $j \in J_H$  as if  $q_j$  was equal to 1 (as in *Opt*). Yet, in this scenario, the reward of the projected orders is modified according to their availability probability  $p_j$  (at the opposite of *Mod*). So, in the formulation of section 7.4.2 the cost  $C_{UV_j} = dist(\alpha_j, \beta_j) * g_j$  becomes  $C_{UV_j} = dist(\alpha_j, \beta_j) * g_j * p_j$  for all loads with  $a_j \in \{t + RH + 1, \dots, t + H\}$ . This scenario is called the expected value scenario: it is based on the expected value of the gain for each load in the uncertain part of the horizon. So, here, all orders are supposed to be available but their reward is modified. Therefore, a slight variation of the formulation from section 7.4.2 gives rise to a single-scenario heuristic denoted EG. The value of the resulting policy  $\pi(EG)$  is called the "Expected value of the Expected Value Solution" (EEVS):  $EEVS = \mu_{\pi(EG)}$ . As in the previous section, this value can be estimated on a sample of scenarios. It can also be used to derive the "Expected Value of the Stochastic Solution" (EVSS) (see [Birge & Louveaux, 1997]), which is defined as the difference between the value of the best policy  $\pi^*$  and the EEVS:  $EVSS = \mu_{\pi^*} - EEVS$ .

By analogy with section 7.5.1, if the upper bound  $O^*(H)$  or any more complex heuristic  $y$  generating a policy  $\pi(y)$  do not statistically outclass  $\pi(EG)$ ,  $\pi(Opt)$  or  $\pi(Mod)$ , then single scenario heuristics (i.e., *EG*, *Opt* or *Mod*) can be sufficient and efficient optimization techniques for the problem.

## 7.5.4 Multiple-scenario approximations

To find a single scenario leading to the best policy requires some luck. Another strategy might be to solve  $K$  independent scenarios, called calibration scenarios, then to select or generate the best policy from the  $K$  solutions. After optimizing a set of  $K$  scenarios generated randomly and independently in a Monte-Carlo fashion, two particular optimization techniques based on multiple-scenario approximations are used to generate some policies  $\pi(y)$ : a Consensus algorithm  $Cs$  and a Restricted Expectation algorithm  $RE^*$  ([Van Hentenryck & Bent, 2006]).

### 7.5.4.1 Consensus algorithm: $Cs$

Once the  $K$  deterministic models issued from calibration scenarios have been solved independently,  $K$  solutions are found for the current decision period  $t$ . These solutions are labeled  $\theta_k$  for  $k = 1, \dots, K$ . Then, the aim of the consensus algorithm  $Cs$  resides in trying to generate a new "compromise" solution based on the most frequent decisions taken among these solutions. At first sight, a common decision means that corresponding variable values have to be the same in different solutions. Yet, identical decisions can also be issued from various variables in models issued from several scenarios, such as for example waiting decisions. For instance, in a solution  $\theta_j$ , the decision for a truck to wait in city  $c$  can be issued from a variable representing a complete standstill decision over the rolling horizon while in another solution  $\theta_k$ ,  $k \neq j$ , the waiting decision for this truck might be issued from the expectation to load in city  $c$  in the next period. Conversely, locally opposite, but globally equivalent decisions can be taken within two solutions  $\theta_j$  and  $\theta_k$ . Consider for instance, two idle trucks located at the same time in the same city. In solution  $\theta_j$ , truck  $A$  waits, truck  $B$  loads and conversely in solution  $\theta_k$ . These decisions are equivalent, even if the replicated variables are not indexed in the same way. So, there should be an aggregation phase of variables representing identical or equivalent decisions. These equivalent decisions can then be used to build a consensus solution. Notice that from the selection of scenarios included in the calibration set to the collection of variables according to their frequency in the solutions and to the construction of the compromise solution, the consensus algorithm represents a family of heuristics or an optimization framework. It is not a unique standardized optimization algorithm. Therefore, this procedure must be more specifically described for our specific case. The aggregation and the assignment parts contain two phases.

Initially, for **the aggregation phase**, we generate several counters for each city  $c \in C$  for each kind of decision: loading, moving unladen, or waiting. At the current period  $t$ , the counters are:

- $CL_{(c,j)}$ , number of solutions  $\theta_1, \dots, \theta_K$  such that load  $j \in J_H$  is loaded in city



$c \in C$ ;

- $CE_{(c,c^1)}$ , total number of trucks planned to move unladen from city  $c$  to city  $c^1$  in period  $t$  over all solutions  $\theta_1, \dots, \theta_K$ ;
- $CW_c$ , total number of trucks planned to wait in period  $t$  in city  $c \in C$  over all solutions  $\theta_1, \dots, \theta_K$ .

So, during this preliminary phase, the number of decisions per city is estimated. However, for the loading decisions, the counters take into account each load  $j$  and for the unladen moves each destination city  $c^1$ . Therefore, in a second phase, all loading operations and all unladen moves are aggregated per city  $c$ , so that the most frequent decisions among loading, unladen move or waiting can be ranked per city. Summing up over all loads  $j$  for  $CL_{(c,j)}$  and over all cities  $c^1$  for  $CE_{(c,c^1)}$ , three counters representing the frequency of decisions per city  $c$  are obtained:  $CL_c$ ,  $CE_c$  and  $CW_c$ .

These counters are divided by  $K$  and rounded to the nearest integer to get the number of decisions of each type per city regardless of parameter  $K$ . These integer values provide the number of similar decisions to be assigned to trucks located in each city:

- $NL_c$  number of trucks that should load in city  $c$ ,
- $NE_c$  number of trucks that should move unladen out of city  $c$ ,
- $NW_c$  number of trucks that should wait in city  $c$ .

These indicator computations end the aggregation phase of decisions.

**The assignment phase** is also split into two parts.

For each idle truck currently located in city  $c$ , we allocate an operation according to the values of the indicators  $NL_c$ ,  $NE_c$  or  $NW_c$ , representing the most frequent decision in that city.

Thus, for each city  $c$ , decisions are assigned truck per truck according to an iterative procedure starting from the largest of  $NL_c$ ,  $NE_c$  or  $NW_c$ . For instance, if  $NL_c \geq NE_c$  and  $NL_c \geq NW_c$ , then the first truck should load an item. Each time an action is allocated to a truck, the value of the current indicator  $NL_c$ ,  $NE_c$  or  $NW_c$  is reduced by one. Therefore, for the next truck an other kind of decision might be assigned. This first phase of the assignment procedure is repeated iteratively for each truck in each city.

The second phase of the assignment procedure aims at defining which load  $j$  should be loaded when a loading decision has been assigned to a truck and which destination city  $c^1$  should be selected when an unladen move has been assigned to a

truck. To select the load  $j$  to be loaded, loading counters from city  $c$  are ranked by non-increasing values of  $CL_{(c,j)}$  and each load is assigned once subsequently. Similarly, unladen moves are specified according to non-increasing values of  $CE_{(c,c^1)}$ . The current largest counter value is reduced by one after each assignment and counters are re-sorted accordingly.

Note that, in case of equal values between  $NL_c$  or  $NE_c$  or  $NW_c$ , preference is given first to loading, then to unladen moves. Finally, if some trucks remain unallocated after all counter values ( $NL_c$ ,  $NE_c$ ,  $NW_c$ ) are reduced to zero, the action assigned to these trucks is to wait. As a consequence, any plan issued from this heuristic is feasible and a consensus solution can always be built.

At first sight, it seems promising to solve  $K$  scenarios rather than a single one in order to generate a decision that is more resilient to the diversity of the exact realization of the future. Yet, the consensus procedure has its own drawbacks. In fact, the final plan is created from the cumulated frequency of single decisions according to an aggregate-disaggregate method. This process might destroy the structure or consistency of the combinatorial solution which is optimal for each scenario  $k = 1, \dots, K$ . This revised solution might not retrieve a better combination of decisions than any single scenario-based one, i.e.  $\theta_1, \dots, \theta_K$ .

#### 7.5.4.2 Restricted Expectation algorithm: RE\*

In order to avoid the drawback of  $Cs$ , the Restricted Expectation algorithm  $RE^*$  is based on the selection of one single solution  $\theta_k$  associated with a scenario  $k$  where  $k \in \{1, \dots, K\}$  and  $K$  is the number of calibrating scenarios. In our setting, the  $K$  scenarios are independently and randomly generated using a Monte-Carlo method. Then, the model associated with each scenario  $k$  is solved and actions (load, move unladen, wait) from  $\theta_k$  for the current decision period  $t$  are recorded as  $\theta_k^t$ , as well as the objective value  $v_k^t$  over the rolling horizon. Afterwards, each decision  $\theta_k^t$  is applied in every remaining scenario  $k' = 1, \dots, K$ ,  $k' \neq k$ . This is feasible as every solution in the current period  $t$  is performed on the deterministic part of the horizon  $\{t, \dots, t + RH\}$ . Doing so makes it possible to compute the optimal solution value,  $v_{(k,k')}^t$ , of a  $(H - 1)$  period scenario defined over  $\{t + 1, \dots, t + H\}$  and issued from scenario  $k'$  but constrained by the applied actions  $\theta_k^t$ . Repeated over all scenarios  $k, k' = 1, \dots, K$  and  $k \neq k'$ , we get the cumulated objective value  $\Theta_k^t$  of action  $\theta_k^t$  in a so called cross-evaluation procedure.

$$\Theta_k^t = v_k^t + \sum_{k'=1, k' \neq k}^{k'=K} v_{(k,k')}^t \quad (7.32)$$

The solution  $\theta_k$  with the highest cumulated value  $\Theta_k^t = \Theta_k^{t*}$  is selected and actions  $\theta_k^t$  are applied.

The name "Expectation" comes from the cross-evaluation procedure that is an attempt to evaluate the expected value of each solution, if it should be effectively applied. The qualifier "Restricted" is due to the fact that the number of calibrating scenarios  $K$  is much lower than the complete set of possible scenarios. Notice that even if  $K$  is equal to the number of possible scenarios, this does not ensure that the solution with the highest expected value,  $\Theta_k^{t*}$ , computed from one single scenario model  $k$  is equal to the decision issued from the best policy  $\pi^*$ . In other words, the optimal solution  $\pi^*$  might not be among the set of optimal solutions  $\theta_k^t$  issued from each single scenario optimization. Conversely to  $Cs$  that tries to generate a consensus, the policy  $\theta_k^{t*}$  issued from  $RE^*$  selects the best solution among a set that contains only elitist solutions. Indeed,  $\theta_k^t$  is the optimal solution for scenario  $k$ , but none of these optimal solutions might be efficient on average.  $\theta_k^{t*}$  might be the least damaging solution rather than a good one. Moreover, when  $K$  increases, the confidence level based on the fact that the cross-evaluation statistically tends to represent the expected value of any solution  $\theta_k^t$  increases. Nevertheless, this does not ensure that  $\pi^*$  belongs to the set of solution  $\theta_k^t$ , even when  $K$  becomes large. Finally, while  $Cs$  requires  $K$  optimizations,  $RE^*$  needs  $K^2$ . It might become prohibitive in practice if  $K$  is large even if each single optimization is polynomially solvable.

### 7.5.5 Subtree algorithm: TR

In order to avoid the previous drawbacks of  $Cs$  (solution deconstruction) and  $RE^*$  (elitist solution selection), it is possible somehow to combine both algorithms  $Cs$  and  $RE^*$  in the formulation presented in section 7.4.4 which provides the best compromise solution over a subset of scenarios. Non-anticipativity constraints act like the cross-evaluation procedure from  $RE^*$  which fixes  $\theta_k^t$  for the current period while allowing different decisions in the remaining part of the rolling horizon according to each scenario (see [Ruszczynski & Shapiro, 2003]). Whereas these constraints cannot ensure that the best decision has been taken for any single scenario, a compromise solution based on the highest reward on average over the subtree can be selected. As in our problem, there are no extreme scenarios in terms of rewards or any penalties for failure, the compromise might be linked to the most frequent "good" decisions, as in  $Cs$ . Fortunately, the subtree formulation turns out to be easy to solve even with the equal flow constraints. Therefore, we have not developed any dedicated optimization techniques (such as Bender's decomposition) to solve it.

As explained in section 7.4.5, the computing time increases with  $ST$  (in our tests, linearly with the model size). The influence of parameter  $ST$  for the numerical performance of the subtree is analyzed in section 7.8.1.

## 7.6 Instances

### 7.6.1 Temporal data

In the experiments, to fit with practice and similarly to [Tjokroamidjojo et al., 2006], the rolling horizon size  $H$  is set to 4 periods representing weekly forecasts. The deterministically revealed part of the horizon  $RH$  equals 1 day, so there are  $H - RH = 3$  periods including stochastic forecasts. The whole horizon  $WH$  involves 20 effective periods with load departures  $a_j \in \{1, \dots, 20\}$ , allowing delivery outside the horizon. This is manageable according to our timeless formulation. The rolling horizon process generates 16 successive decision steps including loading from period 1 to period 16 so that the last decision period includes the same rolling horizon as the other decision periods. As a warm-up period, an initial "0" period is added to start with a fleet of unladen trucks that might be moving unladen or waiting during the starting phase. Therefore, there are  $1 + 16 = 17$  decision steps. This initial bias is similar for all algorithms and just slightly modifies the expected reward value per period. Finally, the maximum unladen distance  $D$  was set to its maximum  $RH = 1$ . Thus, no shortest path problems need to be solved in a preliminary phase (see section 7.3).

a	b	Rolling horizons	c	d	e		
-1	0	...	16	17	18	19	20

Figure 7.5: Planning Horizon

- a : initialization period for locations of the unladen fleet
- b : preliminary repositioning period for unladen fleet
- c : decision period of the last rolling horizon
- c & d : deterministically revealed information periods of the last rolling horizon
- e : stochastic information periods of the last rolling horizon

## 7.6.2 Spatial data

Different sets  $C$  of cities are positioned on three grids of size 10 ( $3*3+1$ ), 20 ( $4*5$ ), 25 ( $5*5$ ), (Figures 7.6, 7.7 and 7.8). They represent a permanent network of logistical platforms aggregating the local clients' orders. All edges are of length 1 and the speed is fixed so that it takes one period to cross an edge. The maximal distance between cities is 4. Tests were performed with complete graphs where each node is a city and also with sub-graphs where 15 or 20 nodes out of figure 7.8 are selected among the 25 possible nodes to create grids representing three different (A, B or C) spatial distribution of cities. It is usual (see Solomon or Li and Lim Instances on <http://www.sintef.no/Projectweb/TOP>) to generate such clustering of clients and to introduce diversity to avoid a uniform spatial distribution.

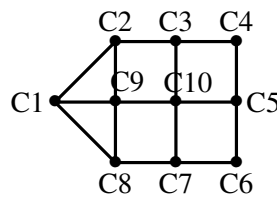


Figure 7.6: Grid of 10 cities

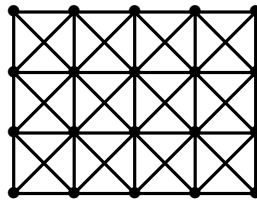


Figure 7.7: Grid of 20 cities

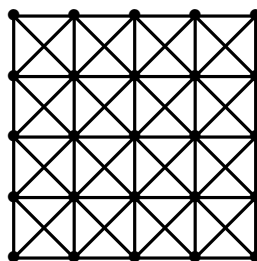


Figure 7.8: Grid of 25 cities

For the first set of instances, all cities are considered equivalent as departure or arrival cities. For the second ones, cities are weighted. These city weights are linked to the city ranges: large city (weight= 4), medium city (weight= 3), small city (weight= 2). The larger the weight, the more loads are issued from or delivered to a city. In table 7.1, the number of cities per range is provided for each sub-graph.

City range	Large	Medium	Small
City weight	4	3	2
Sub-graph 15/25 A: number of cities per range	4	6	5
Sub-graph 15/25 B: number of cities per range	5	5	5
Sub-graph 15/25 C: number of cities per range	2	4	9
Sub-graph 20/25 A: number of cities per range	5	8	7
Sub-graph 20/25 B: number of cities per range	6	7	7
Sub-graph 20/25 C: number of cities per range	3	5	12

Table 7.1: Number of cities per range in each sub-graph

The subgraphs are generated such that respectively the medium cities are more abundant (A), then the larger ones (B) and finally the small ones (C). For example in the sub-graph 15/25 A, there are 4 large cities of weight 4, 6 medium cities of weight 3 and 5 small cities of weight 2.

### 7.6.3 Economic data

- $g_j$  : reward per period for load  $j$  is set randomly between [80, 120] monetary units (MU)
- $e$  : unladen trip cost per period is set to 100 MU
- $f$  : waiting cost per period is set to 75 MU as fuel cost usually represents approximatively 25% of the total cost  $e$

(<http://www.fntr.fr/sites/default/files/conjoncturetrm0212.pdf>)  
 Consequently, the difference between  $f$  and  $e$  (25 MU) is lower than  $g_j$ , so that unladen moves towards a load are profitable compared to staying at standstill.

### 7.6.4 Trucks

Instances include 10 trucks uniformly randomly assigned to any of the cities.

### 7.6.5 Loads

Instances include either 150 or 200 loads. For a first set of instances, loads are randomly and uniformly assigned over the horizon  $\{1, \dots, WH\}$  for departure date  $a_j$ , for departure city  $\alpha_j$  and final destination  $\beta_j$ . For the second set of instances,  $\alpha_j$  and  $\beta_j$  are randomly generated while taking the city weight into account. This means that the probability to send or receive a load is proportional to the departure or arrival city weight. This allows us to compute the percentage of loads issued from every range of cities according to the city weights and the number of cities per range.

City range	Large	Medium	Small
City weight	4	3	2
Sub-graph 15/25 A: number of cities per range	4	6	5
Theoretical percentage of loads	36	41	23
Sub-graph 15/25 B: number of cities per range	5	5	5
Theoretical percentage of loads	45	33	22
Sub-graph 15/25 C: number of cities per range	2	4	9
Theoretical percentage of loads	21	32	47
Sub-graph 20/25 A: number of cities per range	5	8	7
Theoretical percentage of loads	35	41	24
Sub-graph 20/25 B: number of cities per range	6	7	7
Theoretical percentage of loads	41	35	24
Sub-graph 20/25 C: number of cities per range	3	5	12
Theoretical percentage of loads	24	29	47

Table 7.2: Number of cities and percentage of loads per range in each sub-graph

In the sub-graph 15/25 A, the percentage of loads equals  $\frac{(4*4)}{(4*4+6*3+5*2)} = 36\%$  for the large cities,  $\frac{(6*3)}{(4*4+6*3+5*2)} = 41\%$  for the medium and  $23\%$  for the small ones.

### 7.6.6 Stochastic parameters

To model the probability distributions, i.e. the probability that an order becomes available for transportation ( $p_j$ ), it is assumed that this probability depends on the trip duration (see table 7.3) or is related to the city range (see table 7.4). The choice of these parameters comes from the gravity model transposed in Reilly's law of retail gravitation (see [Reilly, 1931]). Consequently, in a first set of instances where all cities are equal, the probability distributions are as follows:

<b>Duration</b>	1	2	3	4
<b>Distribution 1</b>	1	0.5	0.33	0.25
<b>Distribution 2</b>	0.25	0.33	0.5	1
<b>Distribution 3</b>	0.75	0.5	0.33	0.25
<b>Distribution 4</b>	0.5	0.5	0.5	0.5

Table 7.3: Probability  $p_j$  of load availability according to the trip duration: distributions 1 to 4

For instance, according to distribution 3, for a load which has a trip duration equal to 3, the probability to become available is 0.33. Notice that  $p_j = 1$  implies  $q_j = 1$ . So, some deterministic information can be included in  $\{t + RH + 1, \dots, t + H\}$ .

Likewise, according to the city range, the following distributions are generated:

<b>City range</b>	Large	Medium	Small
<b>City weight</b>	4	3	2
<b>Distribution 5</b>	0.75	0.5	0.33
<b>Distribution 6</b>	1	0.5	0.25
<b>Distribution 7</b>	0.25	0.33	0.75
<b>Distribution 8</b>	0.5	0.5	0.5

Table 7.4: Probability  $p_j$  of load availability according to the city range: distributions 5 to 8

For instance, according to distribution 5, for a load which is issued from a large city, the probability to become available is 0.75.

The average probability parameter  $Av p_j$  can be estimated over all loads  $j$ . For the probability distributions based on the trip duration, it is possible also to estimate the average trip duration  $Av d_j$  over all loads according to the graph type. Hence, the average load availability per graph  $Av p_j$  is usually around 0.5 [min 0.37; max 0.67], meaning that the average probability to have a transportation order that becomes available is around 50%. The average trip duration  $Av d_j$  is around 2 periods [min 1.62; max 3.04]. As a consequence, the probability distributions and the graph types lead to various, though not drastically atypical instances.



	$Av p_j$	$Av d_j$	$Av p_j$	$Av d_j$	$Av p_j$	$Av d_j$
<b>City/Graph</b>	10/10		20/20		25/25	
<b>Distribution 1</b>	0.61	1.62	0.58	1.73	0.53	1.87
<b>Distribution 2</b>	0.37	2.34	0.41	2.56	0.47	2.86
<b>Distribution 3</b>	0.53	1.72	0.51	1.83	0.47	1.98
<b>Distribution 4</b>	0.5	2	0.5	2.15	0.5	2.36
<b>City/Graph</b>	15/25 A		15/25 B		15/25 C	
<b>Distribution 1</b>	0.55	1.83	0.55	1.81	0.55	1.82
<b>Distribution 2</b>	0.48	2.92	0.45	2.78	0.45	2.77
<b>Distribution 3</b>	0.48	1.94	0.49	1.92	0.48	1.93
<b>Distribution 4</b>	0.5	2.35	0.5	2.29	0.5	2.30
<b>Distribution 5</b>	0.55	-	0.57	-	0.47	-
<b>Distribution 6</b>	0.63	-	0.67	-	0.49	-
<b>Distribution 7</b>	0.4	-	0.39	-	0.51	-
<b>Distribution 8</b>	0.5	-	0.5	-	0.5	-
<b>City/Graph</b>	20/25 A		20/25 B		20/25 C	
<b>Distribution 1</b>	0.54	1.85	0.55	1.80	0.49	2.02
<b>Distribution 2</b>	0.45	2.80	0.43	2.70	0.44	3.04
<b>Distribution 3</b>	0.48	1.96	0.49	1.91	0.43	2.17
<b>Distribution 4</b>	0.5	2.32	0.5	2.25	0.46	2.57
<b>Distribution 5</b>	0.55	-	0.56	-	0.48	-
<b>Distribution 6</b>	0.61	-	0.64	-	0.5	-
<b>Distribution 7</b>	0.4	-	0.4	-	0.51	-
<b>Distribution 8</b>	0.5	-	0.5	-	0.5	-

Table 7.5: Average load availability and trip duration according to graph

## 7.7 Statistical validation and algorithmic parameters

### 7.7.1 Statistical validation parameters

To validate statistically the algorithmic performances, the number of scenarios per instance  $|F|$  was fixed at 30. This enables us to check the normality of the results and to use the tests explained in section 7.5.1 to determine whether an algorithm outclasses another. The confidence level for the outclassement results is fixed at 95%, i.e., policy  $\pi(1)$  *outclasses* policy  $\pi(2)$  if the  $Z$  statistic is larger than 1.65 (one-sided test).

### 7.7.2 Algorithmic parameters

After preliminary testing, the algorithmic parameter  $K$  representing the number of calibrating scenarios for algorithms  $C_s$  and  $RE^*$  was set to 10. Indeed, initial analysis showed that for the consensus algorithm  $C_s$ , most of the decisions over the  $K$  calibrating scenarios are similar. This leads to identical decisions for loading, moving unladen or waiting over most of the scenarios. Decision frequencies expressed as percentages lie either between 0 and 10 % or conversely between 90 and 100% when considering all calibrating scenarios. Therefore, increasing the number of scenarios is of no use as also demonstrated by a sensitivity analysis on the algorithmic parameter  $K$ .

For the restricted expectation algorithm  $RE^*$ , the computing speed becomes an issue as  $K^2$  optimizations have to be performed. A sensitivity analysis also showed that increasing  $K$  beyond 10 calibrating scenarios does not statistically improve the values obtained. For the subtree  $TR$ , the number of calibrating scenarios  $ST$  was set either to 10 or 30 leading to subtree models labeled  $TR_{10}$  and  $TR_{30}$ .

## 7.8 Numerical results

Results are provided hereafter in table 7.6 where instances are referenced in the form "Distribution- $x$ -Graph  $y$ "; so, for example, instance  $I-20-25C$  is associated with the graph 20-25C and with the probability distribution 1 from table 7.3. For each set of instances, the first table contains the value of the objective function produced by various algorithms. These values are expressed in percentage points, with the lower bound  $O^*(RH) = O^*1$  being set to 0% and the upper bound  $O^*(H) = O^*4$  being set to 100%. Algorithms that make effective use of the stochastic information should perform somewhere between these bounds (note that  $O^*16$  approximates the optimal value  $O^*$  obtained when perfect information is revealed over the whole horizon, and hence  $O^*16$  is usually larger than 100%). This presentation allows us to measure how each algorithm fills in the gap between myopic optimization and the optimal policy that would benefit from deterministic information over each rolling horizon. Table 7.7 delivers the value of the Z-test statistic which allows us to check the pairwise relative performance of algorithms.

### 7.8.1 Results for availability linked to duration (150 loads)

Info		LB			EEVS					UB
Inst./Alg.	$O^*16$	$O^*1$	$Opt$	$Mod$	$EG$	$C_s$	$RE^*$	$TR_{10}$	$TR_{30}$	$O^*4$
1-10	120.4	0	22.8	17.3	37.3	31.2	12.4	48.6	<b>58.2</b>	100
1-15-25A	153.0	0	12.9	38.8	38.4	51.4	43.1	65.7	<b>70.2</b>	100
1-15-25B	153.8	0	13.7	44.7	49.2	45.5	26.7	66.5	<b>75.5</b>	100
1-15-25C	176.0	0	32.8	43.5	67.1	45.2	36.9	72.8	<b>85.2</b>	100
1-20	135.0	0	14.8	41.3	52.5	38.9	46.5	69.8	<b>71.0</b>	100
1-20-25A	167.8	0	6.8	32.5	62.4	21.3	44.9	73.1	<b>78.1</b>	100
1-20-25B	149.6	0	23.3	41.0	46.2	42.0	31.8	<b>70.6</b>	60.0	100
1-20-25C	199.8	0	-22.1	30.1	24.1	27.0	-24.7	61.5	<b>67.9</b>	100
1-25	164.9	0	-83.6	6.5	7.9	12.6	-32.1	<b>54.9</b>	50.6	100
2-10	163.7	0	18.4	38.9	44.7	37.7	26.8	67.4	<b>74.3</b>	100
2-15-25A	221.3	0	69.2	70.2	65.8	70.9	63.7	<b>77.2</b>	76.4	100
2-15-25B	186.1	0	65.1	66.3	70.3	51.0	62.4	83.2	<b>87.4</b>	100
2-15-25C	136.6	0	36.7	60.4	67.5	73.1	42.5	78.3	<b>82.4</b>	100
2-20	204.6	0	59.6	<b>74.5</b>	57.7	53.0	39.3	71.6	70.1	100
2-20-25A	190.1	0	51.6	71.1	81.1	69.4	60.2	82.7	<b>83.3</b>	100
2-20-25B	150.9	0	30.4	40.0	54.5	57.4	53.2	<b>77.5</b>	74.2	100
2-20-25C	180.9	0	65.2	86.5	87.1	79.6	62.0	86.3	<b>89.2</b>	100
2-25	167.3	0	11.4	50.0	65.0	64.2	42.6	<b>69.8</b>	61.0	100
3-10	142.1	0	-7.5	47.5	32.1	37.9	20.7	62.9	<b>63.9</b>	100
3-15-25A	198.2	0	-8.4	50.7	33.3	57.1	29.3	68.0	<b>83.0</b>	100
3-15-25B	249.2	0	27.9	28.7	39.5	30.8	-13.2	91.8	<b>92.6</b>	100
3-15-25C	155.4	0	-5.3	17.3	24.8	31.2	6.6	51.1	<b>61.6</b>	100
3-20	154.1	0	9.0	31.5	33.1	38.9	32.7	56.0	<b>60.6</b>	100
3-20-25A	180.7	0	-37.6	11.2	25.8	11.8	-22.6	39.0	<b>41.7</b>	100
3-20-25B	199.2	0	-33.2	-9.5	26.7	16.8	2.9	54.8	<b>69.1</b>	100
3-20-25C	164.1	0	20.4	54.4	40.2	37.1	42.3	<b>61.8</b>	53.0	100
3-25	177.4	0	-1.3	43.7	22.8	28.7	15.7	61.5	<b>63.4</b>	100
4-10	121.5	0	51.4	51.4	60.4	<b>68.0</b>	45.5	64.8	63.9	100
4-15-25A	201.6	0	24.2	24.2	49.1	38.0	26.7	60.1	<b>79.1</b>	100
4-15-25B	155.8	0	25.5	25.5	48.5	53.3	25.5	64.2	<b>69.0</b>	100
4-15-25C	162.2	0	-19.0	-19.0	27.8	28.7	-9.4	52.1	<b>55.1</b>	100
4-20	165.0	0	-14.5	-14.5	27.6	30.2	7.2	55.6	<b>63.1</b>	100
4-20-25A	145.1	0	10.0	10.0	34.8	41.0	12.9	56.6	<b>66.0</b>	100
4-20-25B	142.8	0	53.9	53.9	52.0	62.7	42.3	68.7	<b>72.4</b>	100
4-20-25C	177.8	0	7.6	7.6	69.5	48.6	14.5	65.4	<b>70.1</b>	100
4-25	148.0	0	15.1	15.1	36.3	23.1	10.0	48.6	<b>52.1</b>	100
<b>Average</b>	<b>168.4</b>	0	15.2	35.6	<b>46.2</b>	43.2	25.8	65.6	<b>69.3</b>	100

Table 7.6: 150 loads and availability linked to duration: performance in percentage

From this table, we can observe that:

- Neither the probability distribution, nor the graph types or ranges appear to be discriminating parameters.

- Algorithms *Opt*, *Mod* and *RE\** give poor results and sometimes under-perform when compared to the myopic bound  $O^*1$ .
- The subtree algorithm with 30 calibrating scenarios  $TR_{30}$  is best 28 times out of 36 instances.
- The subtree algorithm with 10 calibrating scenarios  $TR_{10}$  is best 6 times out of 36 instances and close to  $TR_{30}$ .
- The expected value of the perfect information (EVPI) over all instances is high: 97.9% on average ( $O^*16 - \mu_{\pi^*}$  where  $\mu_{\pi^*}$  is the best value obtained and shown in bold).
- The expected value of the tail information (EVTI) (68.4%) is high, meaning that the information after the rolling horizon is relevant and that increasing the rolling horizon size might be interesting.
- $TR_{30}$  closes the gap between the myopic algorithm  $O^*1$  and the algorithm  $O^*4$  based on the fully revealed information over the rolling horizon by 69.3% on average.
- The expected value scenario algorithm *EG*, which gives the expected value of the expected value solution (EEVS), yields the second best policy after  $TR_{ST}$ .
- Yet, the expected value solution performance (EEVS) varies a lot.
- The expected value of the stochastic solution (EVSS) is 24.4%.
- The consensus algorithm *Cs* is the third best policy, but is far behind by  $TR_{ST}$  and sometimes under-performs.

Consequently, we decided to restrict our tests to  $TR_{ST}$ , *EG* and *Cs*. We provide the outclassement results for the two first policies and for *Cs* only when it provides the best policy value for a particular instance.

Table 7.7 shows that  $TR_{30}$  outclasses *EG* 24 times ( $Z > 1.65$  in bold), meaning that using the stochastic information and a dedicated algorithms is relevant. Moreover,  $TR_{30}$  also outclasses  $TR_{10}$  twice, while  $TR_{30}$  is never outclassed by its challenger  $TR_{10}$ .

Moreover, the subtree algorithm  $TR_{30}$  is never outclassed by any other algorithm, not even by the best policy  $\mu_{\pi^*}$  found for any instance.

For the subtree algorithm *TR*, the speed increases almost linearly with the number of scenarios according to a factor 3 between  $ST = 10$  and  $ST = 30$ .

A deeper performance analysis on the algorithmic parameter  $ST$  was also performed by setting  $ST = 50$ , leading to  $TR_{50}$ . This allowed us to conclude that the larger the subtree, the better  $TR_{ST}$  numerically performs on average while reducing the standard deviation. Similarly, the computing time increases by a factor 5 between  $ST = 10$  and  $ST = 50$ . Yet, as  $TR_{30}$  is never statistically outclassed by  $TR_{50}$ , numerical experiments were eventually restricted to  $ST = 30$  calibrating scenarios to reduce computing time. However in practice, as each optimization step is performed only daily,  $TR_{50}$  might be used for each decision period.

Finally, an important observation is that, for six instances,  $TR_{30}$  is not outclassed by  $O^*4$ , which means that it produces a profit that is not significantly lower than the upper bound. Consequently, stochastic optimization techniques sometimes perform just as well as the algorithm based on available deterministic information. (See table 7.7)

Table 7.7: 150 loads and availability linked to duration: test ( $Z > 1.65$ )

$\mu_{\pi(1)} > \mu_{\pi(2)}$	$O^*4 > O^*1$	$O^*16 > O^*4$	$O^*4 > EG$	$O^*4 > TR_{10}$	$TR_{30} > TR_{10}$	$O^*4 > TR_{30}$	$TR_{30} > EG$	$TR_{30} > \pi^*$
1-10	9.48	4.39	9.71	5.74	1.63	5.55	3.52	
1-15-25A	9.52	7.91	8.48	3.88	0.64	3.92	4.12	
1-15-25B	9.24	8.36	4.52	3.21	1.71	2.50	3.10	
1-15-25C	6.12	8.15	2.80	2.16	1.04	1.23	1.64	
1-20	9.64	7.13	7.96	6.40	0.17	5.16	1.98	
1-20-25A	9.99	7.82	3.39	3.71	4.13	3.35	1.71	
1-20-25B	7.46	5.49	6.39	3.38	-1.54	4.48	1.43	-1.54 (TR10)
1-20-25C	7.07	6.05	5.43	2.29	0.78	1.99	4.25	
1-25	5.89	5.70	7.47	3.84	-0.50	4.39	4.88	-0.50 (TR10)
2-10	6.33	8.89	4.87	3.17	0.78	2.26	2.16	
2-15-25A	9.39	12.52	3.53	2.69	-0.20	2.42	1.56	-0.20 (TR10)
2-15-25B	5.35	6.21	1.94	1.86	0.37	1.08	1.27	
2-15-25C	9.49	4.90	5.85	4.91	0.72	3.12	2.50	
2-20-25A	8.41	9.02	2.10	2.40	-0.16	2.69	0.99	-0.36 (Mod)
2-20-25B	7.86	8.36	4.92	3.46	-0.65	3.72	1.86	-0.65 (TR10)
2-20-25C	6.59	6.94	1.31	1.46	0.43	1.26	0.25	
2-25	7.36	5.86	3.54	2.37	-0.92	3.43	-0.45	-0.92 (TR10)
3-10	9.01	7.46	5.30	3.76	0.09	3.21	2.35	
3-15-25A	8.23	8.56	5.60	2.59	1.60	1.30	3.71	
3-15-25B	3.60	6.49	2.59	0.35	0.06	0.39	2.99	
3-15-25C	8.52	6.01	6.17	4.21	1.24	4.18	3.04	
3-20	7.89	5.61	3.95	3.37	0.59	3.25	2.47	
3-20-25A	7.96	6.88	6.04	5.95	0.23	4.98	1.41	
3-20-25B	6.15	10.85	4.79	2.87	1.38	2.07	2.22	
3-20-25C	9.56	8.62	6.84	5.01	-1.24	5.75	1.35	-1.24 (TR10)
3-25	7.76	8.96	5.72	3.28	0.20	4.00	4.06	
4-10	8.54	5.67	7.06	6.97	-0.19	5.32	0.60	-0.60 (Cs)
4-15-25A	5.19	6.28	3.71	2.65	2.08	1.54	2.34	
4-15-25B	8.43	7.70	5.39	4.87	0.55	3.39	2.56	
4-15-25C	7.72	6.67	6.24	5.18	0.32	5.11	2.54	
4-20	7.67	7.68	9.98	4.26	0.79	4.65	4.42	
4-20-25A	6.81	6.27	7.72	4.41	1.45	3.88	3.64	
4-20-25B	10.75	6.98	6.39	3.49	0.82	3.73	3.38	
4-20-25C	6.89	8.32	3.07	3.66	0.54	3.15	0.06	
4-25	9.19	6.99	6.83	6.41	0.48	5.13	1.85	

## 7.8.2 Results for availability linked to duration (200 loads)

Info	LB	EEVS			UB	
Inst./Alg.	$O^*16$	$O^*1$	$EG$	$Cs$	$TR_{30}$	$O^*4$
1-10	119.1	0	46.5	55.1	<b>69.8</b>	100
1-15-25A	164.2	0	47.0	41.7	<b>63.1</b>	100
1-15-25B	149.9	0	29.6	23.1	<b>47.5</b>	100
1-15-25C	156.5	0	26.7	31.2	<b>56.5</b>	100
1-20	165.4	0	29.6	56.1	<b>79.2</b>	100
1-20-25A	138.4	0	33.1	28.0	<b>53.8</b>	100
1-20-25B	147.8	0	17.3	18.2	<b>50.1</b>	100
1-20-25C	139.5	0	42.6	24.2	<b>54.8</b>	100
1-25	162.1	0	34.7	29.3	<b>62.3</b>	100
2-10	136.4	0	46.0	44.5	<b>70.7</b>	100
2-15-25A	197.1	0	54.1	<b>87.8</b>	79.9	100
2-15-25B	174.7	0	65.0	47.1	<b>68.5</b>	100
2-15-25C	190.3	0	48.3	44.5	<b>68.0</b>	100
2-20	145.7	0	70.7	83.1	<b>90.0</b>	100
2-20-25A	165.6	0	43.3	40.4	<b>62.5</b>	100
2-20-25B	188.1	0	53.0	70.9	<b>84.3</b>	100
2-20-25C	204.9	0	69.3	64.1	<b>86.8</b>	100
2-25	164.4	0	36.1	36.7	<b>48.6</b>	100
3-10	140.5	0	48.6	43.8	<b>66.4</b>	100
3-15-25A	164.1	0	32.5	50.3	<b>73.9</b>	100
3-15-25B	160.6	0	53.2	55.6	<b>68.9</b>	100
3-15-25C	172.1	0	43.0	<i>10.8</i>	<b>59.0</b>	100
3-20	127.9	0	30.4	26.4	<b>52.2</b>	100
3-20-25A	157.9	0	21.1	26.9	<b>43.4</b>	100
3-20-25B	153.7	0	29.9	36.9	<b>52.7</b>	100
3-20-25C	172.5	0	64.8	68.1	<b>82.6</b>	100
3-25	167.8	0	25.4	16.8	<b>47.3</b>	100
4-10	135.4	0	61.6	67.1	<b>79.0</b>	100
4-15-25A	204.3	0	3.9	<i>-10.4</i>	<b>29.6</b>	100
4-15-25B	204.6	0	57.5	<b>68.5</b>	65.8	100
4-15-25C	136.5	0	43.9	55.7	<b>63.3</b>	100
4-20	193.6	0	54.3	52.8	<b>71.1</b>	100
4-20-25A	175.7	0	46.8	50.0	<b>70.3</b>	100
4-20-25B	171.7	0	43.3	43.6	<b>78.2</b>	100
4-20-25C	180.3	0	36.8	<i>5.6</i>	<b>62.4</b>	100
4-25	220.1	0	<b>59.9</b>	11.9	51.4	100
<b>Average</b>	165.3	0	<b>43.0</b>	41.8	<b>64.3</b>	100

Table 7.8: 200 loads and availability linked to duration: performance in percentage

From this collection of figures, one can observe that:

- Neither the probability distribution, nor the graph types or ranges appear to be discriminating parameters.
- The subtree algorithm  $TR_{30}$  is still the best policy, it closes the gap between  $O^*1$  the myopic policy and  $O^*4$  the policy based on the fully revealed information over the rolling horizon by 64.3% on average.
- The expected value of the perfect and tail information (EVPI, EVTI) (100.5%, 65.3%) and the expected value of the stochastic solution (EVSS) (21.8%) are similar and remain still high on average.
- The subtree algorithm  $TR_{30}$  is beaten by the consensus  $Cs$  twice and by the expected value scenario algorithm  $EG$  once.

In the outclassment test (see table 7.9),  $TR_{30}$  outclasses  $EG$  27 times, i.e., the developed stochastic policy is useful. Unfortunately,  $TR_{30}$  turns out to be as effective as  $O^*4$  only twice. This suggests that the expected value of the accessible information (EVAI) is relatively high.

$\mu_{\pi(1)} > \mu_{\pi(2)}$	$O^*4 > O^*1$	$O^*16 > O^*4$	$O^*4 > EG$	$O^*4 > TR_{30}$	$TR_{30} > EG$	$TR_{30} > \pi^*$
1-10	15.47	6.47	10.25	6.23	<b>3.86</b>	
1-15-25A	7.12	9.09	5.24	5.04	<b>1.91</b>	
1-15-25B	8.70	5.53	7.51	3.75	1.18	
1-15-25C	8.07	9.39	6.92	4.13	<b>2.83</b>	
1-20	6.13	8.73	5.68	1.98	<b>3.95</b>	
1-20-25A	11.05	8.62	7.83	8.11	<b>2.57</b>	
1-20-25B	7.89	8.15	7.77	5.16	<b>3.09</b>	
1-20-25C	6.32	7.45	4.87	4.50	1.23	
1-25	5.88	7.06	3.79	4.32	<b>1.89</b>	
2-10	7.06	4.88	4.86	3.24	<b>2.78</b>	
2-15-25A	7.78	10.85	4.55	2.62	<b>2.61</b>	-1.06 ( $C_s$ )
2-15-25B	11.13	11.95	2.92	3.10	0.37	
2-15-25C	6.54	8.78	3.71	2.42	<b>2.01</b>	
2-20	9.04	5.95	3.69	<b>1.35</b>	<b>3.39</b>	
2-20-25A	6.65	7.14	5.61	3.12	<b>2.08</b>	
2-20-25B	8.17	9.00	4.31	2.34	<b>4.10</b>	
2-20-25C	7.03	10.72	3.17	<b>1.26</b>	1.64	
2-25	7.04	7.43	6.97	5.37	1.50	
3-10	9.11	7.63	4.98	4.23	<b>1.81</b>	
3-15-25A	7.06	6.22	5.87	3.73	<b>3.64</b>	
3-15-25B	8.47	9.17	5.28	3.61	<b>2.13</b>	
3-15-25C	8.17	7.58	5.57	4.66	1.61	
3-20	8.89	4.59	6.59	7.26	<b>2.34</b>	
3-20-25A	8.60	6.26	9.18	5.94	<b>1.90</b>	
3-20-25B	9.21	6.44	7.87	5.24	<b>1.89</b>	
3-20-25C	7.99	6.10	3.24	2.03	<b>2.32</b>	
3-25	6.01	7.08	7.09	5.43	<b>2.11</b>	
4-10	8.95	5.58	4.05	2.34	<b>2.54</b>	
4-15-25A	4.82	6.84	5.10	3.76	<b>1.88</b>	
4-15-25B	10.93	9.58	4.93	3.35	0.98	-0.30 ( $C_s$ )
4-15-25C	9.94	6.37	6.53	4.21	<b>2.38</b>	
4-20	6.86	9.76	3.45	2.91	1.46	
4-20-25A	8.03	8.61	4.62	3.16	<b>2.54</b>	
4-20-25B	8.68	6.62	5.89	2.27	<b>3.48</b>	
4-20-25C	6.17	7.62	4.43	3.24	<b>1.70</b>	
4-25	3.64	11.19	2.30	3.02	-0.73	<b>-0.73 (EG)</b>

Table 7.9: 200 loads and availability linked to duration: test ( $Z > 1.65$ )



### 7.8.3 Results for the availability linked to departure city range

Info		LB	EEVS			UB
Inst./Alg.	$O^*16$	$O^*1$	$EG$	$C_s$	$TR_{30}$	$O^*4$
5-15-25 A	222.0	0	73.6	<b>80.0</b>	79.2	100
6-15-25 A	156.1	0	78.6	<b>90.8</b>	89.7	100
7-15-25 A	171.0	0	57.2	68.0	<b>70.7</b>	100
8-15-25 A	187.3	0	<b>54.3</b>	13.8	53.4	100
5-15-25 B	153.1	0	57.7	61.2	<b>81.6</b>	100
6-15-25 B	165.7	0	55.8	42.8	<b>60.3</b>	100
7-15-25 B	194.7	0	56.5	60.4	<b>61.0</b>	100
8-15-25 B	201.4	0	86.7	60.8	<b>100.0</b>	100
5-15-25 C	192.4	0	64.1	53.8	<b>78.8</b>	100
6-15-25 C	125.9	0	62.7	78.3	<b>88.0</b>	100
7-15-25 C	179.2	0	63.9	49.6	<b>70.4</b>	100
8-15-25 C	192.0	0	47.0	20.0	<b>63.5</b>	100
5-20-25 A	195.1	0	63.9	45.2	<b>65.9</b>	100
6-20-25 A	153.8	0	52.1	54.4	<b>74.3</b>	100
7-20-25 A	253.9	0	38.6	32.1	<b>44.5</b>	100
8-20-25 A	225.7	0	7.3	-36.5	<b>21.9</b>	100
5-20-25 B	141.9	0	62.9	33.2	<b>68.4</b>	100
6-20-25 B	147.4	0	62.7	53.4	<b>74.2</b>	100
7-20-25 B	176.7	0	52.1	52.7	<b>66.1</b>	100
8-20-25 B	165.1	0	49.8	25.6	<b>54.2</b>	100
5-20-25 C	171.7	0	51.4	61.2	<b>67.7</b>	100
6-20-25 C	215.3	0	39.1	23.6	<b>56.1</b>	100
7-20-25 C	142.9	0	53.6	54.0	<b>61.3</b>	100
8-20-25 C	150.3	0	67.3	41.7	<b>71.3</b>	100
<b>Average</b>	<b>178.4</b>	0	<b>56.6</b>	46.7	<b>67.6</b>	100

Table 7.10: 150 loads and availability linked to city range: performance in percentage

From these results, one can notice that:

- Neither the probability distribution, nor the graph types or ranges appear to be discriminating parameters.
- The subtree algorithm  $TR_{30}$  provides the best value except twice for  $C_s$  and once for the expected value scenario  $EG$ , but it never under-performs.
- The average performance of  $TR_{30}$  remains around 2/3 of the gap between  $O^*1$  the myopic policy and  $O^*4$  the policy based on the fully revealed information.
- The subtree algorithm  $TR_{30}$  is equal to the upper bound  $O^*4$  once.
- The expected value of the perfect and tail information (EVPI, EVTI) remain high on average 110.7%, 78.4%.

- The expected value scenario algorithm *EG* (EEVS) performs better than previously, only 11% behind  $TR_{30}$ .
- So, the expected value of the stochastic solution (EVSS) decreases (11.1%).

$\mu_{\pi(1)} > \mu_{\pi(2)}$	$O^*4 > O^*1$	$O^*16 > O^*5$	$O^*4 > EG$	$O^*4 > TR_{30}$	$TR_{30} > EG$	$TR_{30} > \pi^*$
5-15-25 A	8.91	12.72	3.14	2.40	0.78	-0.10 ( <i>Cs</i> )
6-15-25 A	9.29	9.65	3.98	<b>1.87</b>	<b>2.41</b>	-0.27 ( <i>Cs</i> )
7-15-25 A	6.00	5.25	3.61	3.09	1.39	
8-15-25 A	4.57	5.02	3.01	3.79	-0.08	<b>-0.08</b> ( <i>EG</i> )
5-15-25 B	10.06	5.54	4.40	2.15	<b>3.74</b>	
6-15-25 B	6.78	7.34	5.00	3.96	0.50	
7-15-25 B	6.25	7.70	3.77	3.18	0.49	
8-15-25 B	8.37	9.84	1.15	<b>0.00</b>	<b>1.81</b>	
5-15-25 C	7.87	7.37	3.58	1.65	1.39	
6-15-25 C	10.18	7.16	6.40	3.80	<b>4.26</b>	
7-15-25 C	6.95	9.22	3.98	2.97	0.73	
8-15-25 C	5.99	9.12	4.27	3.59	1.44	
5-20-25 A	7.69	10.11	3.87	3.30	0.22	
6-20-25 A	10.46	8.99	4.33	3.32	<b>2.77</b>	
7-20-25 A	4.47	6.82	3.61	3.11	0.48	
8-20-25 A	4.00	5.74	4.46	3.57	0.62	
5-20-25 B	6.39	4.27	3.13	2.78	0.75	
6-20-25 B	8.12	7.11	5.54	3.05	1.43	
7-20-25 B	7.98	9.37	4.64	3.16	1.32	
8-20-25 B	9.60	8.88	5.90	5.42	0.51	
5-20-25 C	7.00	7.68	4.92	4.10	<b>1.99</b>	
6-20-25 C	7.53	9.85	5.67	3.52	<b>1.66</b>	
7-20-25 C	5.80	5.13	3.72	3.51	0.97	
8-20-25 C	7.91	7.64	4.42	4.04	0.59	

Table 7.11: 150 loads and availability linked to city range: test ( $Z > 1.65$ )

One can check that:

- The subtree algorithm  $TR_{30}$  is never outclassed but outclasses 7 times the expected value scenario algorithm *EG*.
- The fully revealed information algorithm over the rolling horizon  $O^*4$  always outclasses  $TR_{30}$  except once, i.e. the expected value of the accessible information (EVAI) remains relevant.

The subtree algorithm  $TR_{30}$  still outperforms *EG* (EEVS), even if the relative performances and also the expected value of the stochastic solution (EVSS) decrease when the probability distribution is linked to the departure city range instead of the trip duration.

#### 7.8.4 Preliminary conclusions

From the previous results, we can conclude that the subtree algorithm including 30 calibrating scenarios  $TR_{30}$  is usually the best performing algorithm. It outclasses all other algorithms ( $Opt$ ,  $Mod$ ,  $Cs$ ,  $RE^*$ ) while it is not outclassed itself. It closes approximatively by  $2/3$  the gap between the lower bound provided by the myopic algorithm  $O^*1$  and the upper bound issued from the algorithm using the fully revealed information over the rolling horizon  $O^*4$ . This proves that the expected value of the multi-period model (EVMPM), which measures the gap between the two bounds, is important and that information about the projected orders might be useful to close this gap. Other tests show also that  $TR_{30}$  always outclasses  $O^*1$  proving the interest of using the information from the rolling horizon. Unfortunately, it statistically reaches the upper bound  $O^*4$  only a few times. This means that the stochastic information is useful but not sufficient to equal the performances of an a-posteriori solution assessing the knowledge of the deterministic information over the rolling horizon, the accessible information. So, the expected value of the accessible information (EVAI) remains significant. In the same sense, by observing the gap between algorithms based either on fully revealed information over the rolling horizon  $O^*4$ , or on fully revealed information over the whole horizon  $O^*16$ , we see that the tail information (EVTI) remains high. One can conclude that increasing the rolling horizon length might be useful. But in practice, this length may be constrained by the commercial process and may not be increased on demand. The expected value scenario algorithm  $EG$  that provides the expected value of the expected value solution (EEVS) performs well. Nevertheless,  $EG$  is usually left behind the subtree algorithm  $TR$ . So, the expected value of the stochastic solution (EVSS) is high, meaning that using the stochastic information based on the probability distribution is profitable. Finally, the results show that the expected value of the perfect information (EVPI) is high. One cannot conclude that this gap leads to some room for improvement for the algorithms as the value of the best policy of the stochastic problem over the rolling horizon  $\mu_{\pi^*}$  could be by far different from the policy for the fully revealed deterministic problem. Hence, we insist on highlighting that EVPI, EVTI, EVAI do not measure the quality of the algorithms but the interest to increase the rolling horizon length or to acquire deterministic data over the rolling horizon.

### 7.8.5 Subtree algorithm robustness analysis

In practice, it may be hard to determine  $p_j$  (i.e., the exact probability distribution of a specific load), as it can only be derived from the aggregation of historical data or from expert opinions. Therefore, it is important to analyze the sensitivity of the results to the exactitude of the valuation of the distribution. We call this process “robustness analysis” in the present framework. Except  $Opt$ , which radically rounds up to 1 any outcomes of the probability distribution, meaning that  $Opt$  does not depend on the probability distribution at all, all other algorithms are based on a presupposed knowledge of the probability distribution  $p_j$  which is a strong assumption. If we do not know the exact probability distribution, a strategy should be set up to still perform efficiently with an approximate probability distribution. Consequently, tests were performed when the probability  $p_j$  used to generate the tests scenarios were set to 20% or 80% while the probabilities  $p'_j$  used to generate the subtrees were issued from distributions assuming respectively 20%, 50% or 80% of load availability. Consequently, the robustness in this case is linked to the ability of the algorithm to still perform when under or over-estimating the load availability. Moreover, the reverse situation of an over (80%) or under (20%) estimation of the distribution  $p'_j$  in the generation of the subtree compared to an average real distribution of 50% for the load parameter  $p_j$  was also tested (see table 7.12). The following results from table 7.12 show the performances of various algorithms in percentage between the myopic planing  $O^*1$  considered as 0% and  $O^*4$  representing the fully revealed information over the rolling horizon. Algorithms  $EG^{50}$  represents the expected value scenario algorithm calibrated with a 50% distribution. The subtree algorithm  $TR_{30}^X$  represents the subtree algorithm  $TR_{30}$  calibrated with a  $X\%$  distribution. So, the calibrating probability  $p'_j$  equals  $X$ . Instances are based on 2 graphs 15-25 A and 20-25 A. A label  $d - Y$  or  $w - Y$  is added for the distribution which depends either on the duration  $d$  or the city range  $w$ .  $Y\%$  represents the distribution parameter of the effective distribution, i.e. the real probability  $Y = p_j$ .

Info		LB	EEVS				UB
Inst./Alg.	$O^*16$	$O^*1$	$EG^{50}$	$TR_{30}^{20}$	$TR_{30}^{50}$	$TR_{30}^{80}$	$O^*4$
d-20-15-25 A	361.4	0	25.2	61.9	<b>65.0</b>	38.2	100
w-20-15-25 A	283.7	0	34.5	<b>87.2</b>	72.5	-0.1	100
d-20-20-25 A	229.0	0	31.9	<b>70.5</b>	45.3	35.6	100
w-20-20-25 A	298.4	0	3.7	-7.0	<b>9.7</b>	6.7	100
<b>Average 20</b>	293.1	0	23.8	<b>53.2</b>	48.1	20.1	100
d-80-15-25 A	152.6	0	91.0	71.4	<b>111.2</b>	106.1	100
w-80-15-25 A	217.0	0	44.4	43.1	87.1	<b>87.4</b>	100
d-80-20-25 A	129.7	0	85.3	47.5	96.1	<b>99.3</b>	100
w-80-20-25 A	184.4	0	20.8	39.2	45.4	<b>47.2</b>	100
<b>Average 80</b>	170.9	0	60.4	50.3	84.9	<b>85.0</b>	100
d-50-15-25 A	201.6	0	49.1	48.0	<b>79.1</b>	59.2	100
w-50-15-25 A	187.3	0	<b>54.3</b>	21.9	53.4	35.8	100
d-50-20-25 A	145.1	0	34.8	47.5	<b>66.0</b>	43.3	100
w-50-20-25 A	225.7	0	7.3	10.3	<b>21.9</b>	-17.6	100
<b>Average 50</b>	189.9	0	36.4	31.9	<b>55.1</b>	30.2	100

Table 7.12: 150 loads and robustness of distribution parameter: performance in percentage

One can notice that :

- The expected value scenario algorithm  $EG^{50}$  under-performs.
- Numerically, no calibration of  $TR_{30}^X$  is robust to severe variations against reality, except  $TR_{30}^{50}$ . But any  $TR_{30}^X$  still provides a positive value on average as compared to a myopic policy.
- A trend shows that the better the calibration  $X$  sticks to reality  $Y$ , the better the results are.
- When the probability parameter increases in reality, meaning that  $Y$  is high,  $TR_{30}^X$  fills the gap and sometimes overtakes the practical upper bound  $O^*4$ . Conversely, when the probability parameter decreases in reality, meaning that  $Y$  is low,  $TR_{30}^X$  is close to the lower bound  $O^*1$ . So, the expected value of the accessible information (EVAI) decreases with the probability parameter  $Y$ .
- $X = 50\%$  is a compromise setting for  $TR_{30}^X$  that resists to under and over availability in reality.  $TR_{30}^{50}$  avoids losses due to an underestimation ( $TR_{30}^{20}$ ) or overestimation ( $TR_{30}^{80}$ ) of the availability parameter in the calibration during the generation of the subtree scenarios.

Regarding the statistical significance, results are as follows (see table 7.13):

- The subtree calibrated with a probability parameter of 50%,  $TR_{30}^{50}$ , is never outclassed. Moreover,  $TR_{30}^{50}$  often outclasses  $TR_{30}^{20}$  and  $TR_{30}^{80}$  for  $Y = 50\%$ .

- Logically,  $TR_{30}^{20}$  or  $TR_{30}^{80}$  are often outclassed when the reality is contrary to their estimation.
- Sometimes  $TR_{30}^{50}$  or  $TR_{30}^{80}$  reach the upper bound for  $Y = 80\%$ .

As a consequence, it is recommended in practice to try to approximate the real probability distribution  $Y$  starting from a calibration distribution  $X = 50\%$ . In case of an unstable probability distribution, it is better to stick to  $X = 50\%$  in the algorithmic setting.

Table 7.13: 150 loads and robustness of probability distribution parameter: test ( $Z > 1.65$ )

$\mu_{\pi(1)} > \mu_{\pi(2)}$	$O^*4 > O^*1$	$O^*16 > O^*4$	$O^*4 > EG^{50}$	$O^*4 > TR_{30}^{20}$	$O^*4 > TR_{30}^{50}$	$O^*4 > TR_{30}^{80}$	$TR_{30}^{20} > TR_{30}^{50}$	$TR_{30}^{20} > TR_{30}^{80}$	$TR_{30}^{50} > TR_{30}^{80}$
d-20-15-25 A	3.87	4.40	2.17	1.31	1.09	1.60	-0.12	0.6	0.80
w-20-15-25 A	4.34	6.20	3.94	1.07	2.10	5.60	1.26	5.14	<b>3.82</b>
d-20-20-25 A	5.68	5.49	4.26	2.28	3.40	3.89	1.64	1.97	0.68
w-20-20-25 A	3.57	5.57	3.23	4.32	3.20	3.89	-0.59	-0.55	0.10
d-80-15-25 A	11.62	9.03	1.03	3.83	-1.25	-0.97	<b>-6.65</b>	-5.06	0.78
w-80-15-25 A	4.80	8.58	3.28	3.46	0.87	1.04	<b>-2.68</b>	-2.91	-0.02
d-80-20-25 A	14.44	6.06	2.59	8.73	1.04	0.15	<b>-8.92</b>	-8.01	-0.86
w-80-20-25 A	5.64	9.25	4.51	4.69	3.45	4.70	-0.42	-0.55	-0.14
d-50-15-25 A	5.19	6.28	3.71	3.26	1.54	2.71	<b>-2.36</b>	-0.80	<b>1.90</b>
w-50-15-25 A	4.57	5.02	3.01	5.16	3.79	4.34	<b>-2.31</b>	-0.74	1.34
d-50-20-25 A	6.81	6.27	7.72	5.65	3.88	5.13	<b>-2.50</b>	0.43	<b>2.94</b>
w-50-20-25 A	4.00	5.74	4.46	3.71	3.57	6.28	-0.62	1.07	<b>1.93</b>

### 7.8.6 Subtree algorithm sensitivity to horizon length

Another drawback, that can happen in practice on the probability distribution is the reduction of the number of forecasting periods inside the rolling horizon. In that case, the interest for a sophisticated algorithm such as the subtree one might decrease. Taking back results from table 7.6, tests were performed for  $TR_{30}$  with a reduced rolling horizon length of 3 and 2 periods instead of 4 initially. The label  $TR_{30} - H3$  in the following table stands for the subtree algorithm  $TR_{30}$  with a rolling horizon  $H$  of 3 periods. The value for the expected value scenario  $EG$  is still evaluated with a 4 periods rolling horizon length.

Info		LB	EEVS				UB
Inst./Alg.	$O^*16$	$O^*1$	$EG$	$TR_{30} - H2$	$TR_{30} - H3$	$TR_{30} - H4$	$O^*4$
1-10	120.4	0	37.3	33.3	42.1	58.2	100
1-15-25-A	153.0	0	38.4	30.7	53.6	70.2	100
1-15-25-B	153.8	0	49.2	56.0	63.4	75.5	100
1-15-25-C	176.0	0	67.1	47.4	74.7	85.2	100
1-20	135.0	0	52.5	44.4	60.8	71.0	100
1-20-25-A	167.8	0	62.4	47.7	65.4	78.1	100
1-20-25-B	149.6	0	46.2	21.4	55.7	60.0	100
1-20-25-C	199.8	0	24.1	34.7	55.8	67.9	100
1-25	164.9	0	7.9	37.3	47.1	50.6	100

Table 7.14: 150 loads and horizon length parameter: performance in percentage

The subtree algorithm always with 4 (and sometimes with but 3 periods in the rolling horizon) outperforms the expected value scenario algorithm  $EG$ . So, the expected value of the stochastic solution  $EVSS$  is already positive with a shorter rolling horizon length. Providing an extra-period is valuable as  $TR_{30} - H4 > TR_{30} - H3 > TR_{30} - H2 > O^*1$ . Consequently, the subtree is resilient to a false valuation of the probability distribution or a reduction of the rolling horizon length.

## 7.9 Conclusions

In this chapter, we present a methodology to tackle stochastic multi-period problems in transportation which are often reduced to a sequence of mono-period deterministic models in the scientific literature and practice. In this kind of settings, several decision policies have to be compared; firstly with the bound provided by the myopic optimization over the deterministic part of the horizon; and secondly with the bounds generated by the a-posteriori optimization using the deterministic information over the rolling and the whole horizon. Using the expected value of the expected value solution and a toolbox of easily implementable algorithms, one can:



Prove the managerial benefit a multi-periods model; Evaluate the profit of using the stochastic information; Measure the expected value of perfect information. Scenario simulations make it possible to statistically validate numerical results and also to differentiate the performances of algorithms. In the present vehicle assignment example, a commercial LP solver fortunately provides an optimal integer solution to the subtree formulation. Moreover, this methodology and algorithmic developments significantly close the gap between the myopic and a posteriori bounds in a wide range of datasets and for several probability distributions and graphs. Furthermore, even if the performances slightly decrease on average, the subtree algorithm facing a probability distribution which deviates from the calibration one is robust when using a default 50% calibration parameter. Finally, despite the fact that the framework does not ensure quantitative results in advance, this pragmatic and generic approach is flexible and can be applied in a systematic way for other logistical problems. This leads to measure several values of information which might imply to modify the booking process or to valueate the benefit for using easily implementable scenario-based algorithms.

## **7.10 Acknowledgments**

We thank Yasemin Arda and Frédéric Semet for their comments offered in the course of our research and Isabelle Peere for her editing of the present document. The project leading to these results was partially funded by the Belgian Scientific Policy Office (Grant P7/36).

## **Part IV**

# **Conclusions and perspectives**



## Chapter 8

# Conclusions and perspectives

### 8.1 Conclusions

Finalizing the thesis requires to point the contributions brought by the research according to the thesis subject proposed in the first chapter. It is also important to show the limits of what has been performed and what could be done in future research by depicting perspectives.

In the second chapter, the situation of multi-period stochastic optimization problems in transportation is analyzed and positioned within the scientific literature. Even if some works in transportation management have investigated multi-period models or stochastic optimization problems, the intersection of these three fields is still sparse and consequently a prospective area of research. This sector suffers from: a lack of standardization on the terminology, variations in the models tackled, few benchmark instances, diverse ways to report results and to assess the algorithmic performances. This does not lead to easy comparisons among models and algorithms. A preliminary contribution of the thesis relies on this statement and on proposing a methodology to face this situation.

In the third chapter, which focuses on problems in a multi-period deterministic environment, successive theoretical notions are described in order to formalize the rolling horizon procedure. The rolling horizon parts and sizes are discussed as well as the dynamism of the rolling process. Then, the inherent warm-up and end-of-horizon biases are explained. Moreover, a-priori and a-posteriori bounds and several concepts of value of information are defined and some new ones are proposed as contributions. All these bounds and information values are reported in a chart for visualization. This leads to an understandable managerial tool to evaluate the benefit of dealing with multi-period problems. Even if this framework is mostly a blend of previously known notions, some new theoretical ones have been proposed and it is

the first time that they are put together in such an explicit representation.

The fourth chapter states advantages and drawbacks of the algorithmic techniques proposed to deal with stochastic optimization problems. Several scenario-based models and related algorithms are depicted and used for simulation. This refers to mono-scenario approximation, multiple scenario approaches and subtree optimization. The value of the expected value solution and the value of the stochastic solution are added to the decision framework. They allow managers to evaluate the necessity of dealing with stochastic information and of using complex stochastic optimization methods. These policy values and in particular the policy based on the expected value scenario solution can be reported in the chart including bounds and information values. As a contribution, a complete visualization framework aggregating together all these notions is proposed.

In the fifth chapter, the statistical validation of policy performances is considered. Previous notions from the literature are evoked as a survey. Advantages and drawbacks of these performance measurements are highlighted. This leads to propose a systematic validation methodology to check the statistical significance of differences between policies. This allows to validate the selection of an optimization strategy among others and to evaluate its performance by comparing the policy values to the bounds. Even, if the statistical test is standard, its integration and repetitive utilization in the experimental process in this field of operations research can be considered as a valuable contribution. Moreover, the importance of robustness analysis which accounts for the variations of the projected probability distributions is brought to light for the final selection of the optimization tool.

In the second part of the thesis, in chapters 6 and 7, two applications that were the supports for the development of the methodology are presented. The process leading to the methodology started from the field according to an inductive process. Therefore, the chapters do not comply formally with all the aspects of the submitted methodology which was issued from trials and errors while dealing with these applications.

As practical contributions, both chapters propose new problems, formulations and solution approaches, and they draw conclusions on performing algorithmic policies.

The first application proposes and tests solution methods for a vehicle loading problem. A simple heuristic based on a single scenario optimization performs very well. The selected heuristic is robust and drastically closes the gap between the myopic and the a-posteriori bounds. Hence, the multi-period stochastic problem gently reduces to a timeless deterministic optimization one. This article is published. The scientific contributions are a new model, its formulation and an efficient policy.

The second application deals with a vehicle assignment problem. The problem is depicted and deterministic formulations are proposed to solve scenario-based models approximating the stochastic problem. Mono-scenario approximations and multiple scenario approaches are described and applied on various instances. Results are reported and this leads to eventually propose a more complex subtree optimization algorithm which closes on average two-thirds of the gap between the a-priori and a-posteriori bounds. Moreover, this algorithm statistically outperforms the expected value solution. This proves the benefit of more sophisticated optimization techniques in some cases. Beside this, the subtree algorithm calibrated with a 50% probability distribution is not outclassed by other policies under variations of the probability distribution; i.e., the subtree algorithm leads to a robust policy. Furthermore, practically, the subtree formulation is solvable by a commercial solver as in practice, the optimal integer solution is provided in a few seconds by the optimal solution of the linear relaxation of the problem. Similarly to the previous application, this research proposes a new model, two formulations and a performing policy.

As a summary, both applications are compared in the following table 8.1 to demonstrate that the generic methodology for multi-period stochastic optimization problems can be applied to various transportation settings.

<b>Criteria</b>	<b>Application 1</b>	<b>Application 2</b>
Problem name	Vehicle loading	Vehicle assignment
Stochastic parameter	Release dates	Load availability
Probability distribution	% per period in a TW	% in 1 period
Stochasticity characteristic	Discrete	Discrete
Routing importance	Low	High
Loading importance	High	None
Instance classes	Natural	Theoretical
Horizon length	20 Periods	20 Periods
Rolling horizon length	4 Periods	4 Periods
Deterministic periods	[t]	[t, t+1]
Action period	Current period	Current period
Frozen period	None	None
Objective function	Min cost	Max profit
Temporality in formulation	Preprocessed	Preprocessed
Preprocessing	Cheapest period	Infeasible links
Solution technique	Cplex default	Cplex default
Branch & Bound	Yes	None $LP \cong IP$
Complexity	NP-Hard	Network flow
CPU Time per decision period	0.5-13 sec	1-5 sec
Variables	Binary	Binary
Warm-up	None (low influence)	1 period
End-of-horizon	cost per ton sent	loaded trips
Number of calibrating scenarios $K$	10	10 and 30
Sensitivity analysis on $K$	6-8-10...20	10-30-50
EVMPM	High	High
EVPI	Low	High
EVAI	Low	High
EVTI	Low	High
EEVS	High	High
A posteriori bound $O_T^*$	$O^*$	$O^* 16$
A posteriori bound $O_H^*$	$O^*(4)$	$O^* 4$
Myopic bound $O_{RH}^*$	$LO$	$O^* 1$
Expected value scenario	Outclassed	Second best outclassed
Modal value scenario	Outclassed	Outclassed
Optimist scenario	Not-outclassed	Outclassed
Optimism definition	At the earliest	Everything available
Consensus	Outclassed	Outclassed
Restricted expectation	Not-outclassed	Outclassed
Subtree	Intractable	Best
Robustness	Optimist Independent	$TR$ calibrated 50%

Table 8.1: Applications : a comparison

Actually, one can see that opposite characteristics can be tackled:

- transportation problems focusing on loading or routing
- NP-Hard or polynomially solvable formulations of the deterministic model
- minimization or maximization objective functions
- various levels for the values of information
- various gaps between the policy values and the bounds
- relative performances of mono and multiple scenario approaches

As a general contribution, this thesis covers a broad area of the scientific research field, including theoretical descriptions of multi-period and stochastic problems, statistical validation of results and managerial advices. Moreover, the applications propose interesting problem formulations and performing robust policies. So, from theory to practice, this thesis brings original contributions and a reproducible experimental framework.

Finally, as most simulation algorithms proposed in this thesis are based on the optimization of a deterministic model, switching to the multi-period stochastic model might not require much efforts for many well-known problems. Indeed, this next step only requires some data management: a dynamic process setting, a scenario generator, a statistical validation procedure, a set of bounds and several values of information. Doing so, one gets for free several mono-scenario optimization policies and with a few developments, consensus and restricted expectation approximations. As a consequence and major contribution, the methodology proposed in this thesis might serve as a generic sequence for researchers to switch from mono-period deterministic problems to multi-period stochastic ones.



## 8.2 Perspectives

The perspectives opened in this thesis are quite wide as the present work entered many domains: problems, models, formulations, algorithms, multi-period decision making, stochastic optimization, simulation, scenario generation, bounds, statistical validation, robustness analysis and policy selection. In transportation, there is still much to be done in all these domains. However, for the operational research community, as a recommendation, it might be of interest to standardize the description of the dynamism of multi-period or on-line problems. Indeed, the dynamic part is not always explicitly described. Another improvement would be to present results together with myopic and a-posteriori bounds and to use a common statistical validation technique for the results. Readers should tend to have similar requirements for modeling, results presentation and statistical validation.

It would also be useful if warm-up and end-of-horizon effects could be analyzed more closely in order to define good practices to define a measurable steady state. Therefore, a standardized simulation framework could be of high interest.

Moreover, subtree optimization seems to be a promising optimization technique. As it often suffers from computational complexity, it would be interesting to develop more efficient solution techniques for subtree optimization such as column generation taking into account the structure of the non-anticipativity constraints.

Models and algorithms dealing with correlated and/or continuous probability distributions should also be of interest. As another option, problems with multiple stochastic parameters could also be modeled, e.g. vehicle routing problem with stochastic customers, stochastic demand and stochastic transportation times, but this seems more prospective than a short-term perspective.

**Part V**

**Bibliography**



# Bibliography

- Albareda-Sambola, M., Fernández, E., & Laporte, G. (2014). The dynamic multi-period vehicle routing problem with probabilistic information. *Computers & Operations Research*, *48*, 31–39.
- Angelelli, E., Bianchessi, N., Mansini, R., & Speranza, M. G. (2009). Short term strategies for a dynamic multi-period routing problem. *Transportation Research Part C: Emerging Technologies*, *17*, 106–119.
- Angelelli, E., Savelsbergh, M. W. P., & Speranza, M. G. (2007). Competitive analysis for dynamic multi-period uncapacitated routing problems. *Networks*, *49*, 308–317.
- Angelelli, E., & Speranza, M. G. (2002). The periodic vehicle routing problem with intermediate facilities. *European Journal of Operational Research*, *137*, 233–247.
- Arda, Y., Crama, Y., Kronus, D., Pironet, T., & Van Hentenryck, P. (2014). Multi-period vehicle loading with stochastic release dates. *EURO Journal on Transportation and Logistics*, *3*, 93–119.
- Barr, R. S., Golden, B. L., Kelly, J. P., Resende, M. G. C., & Stewart, J., W. R. (1995). Designing and reporting on computational experiments with heuristic methods. *Journal of Heuristics*, *1*, 9–32.
- Bellman, R. (1957). *Dynamic Programming*. (1st ed.). Princeton, NJ, USA: Princeton University Press.
- Berbeglia, G., Cordeau, J.-F., Gribkovskaia, I., & Laporte, G. (2007). Static pickup and delivery problems: A classification schema and survey. *TOP*, *15*, 1–31.
- Berbeglia, G., Cordeau, J.-F., & Laporte, G. (2010). Dynamic pickup and delivery problems. *European Journal of Operational Research*, *202*, 8–15.

- Bertsimas, D., Brown, D., & Caramanis, C. (2011). Theory and applications of robust optimization. *SIAM Review*, *53*, 464–501.
- Bertsimas, D., & Sim, M. (2004). The price of robustness. *Operations Research*, *52*, 35–53.
- Birge, J. R., & Louveaux, F. (1997). *Introduction to Stochastic Programming*. Berlin: Springer.
- Blackburn, J. D., & Millen, R. A. (1980). Heuristic lot-sizing performance in a rolling-schedule environment. *Decision Sciences*, *11*, 691–701.
- Boudia, M., Louly, M., & Prins, C. (2007). A reactive grasp and path relinking for a combined production distribution problem. *Computers and Operations Research*, *34*, 3402–3419.
- Brotcorne, L., Laporte, G., & Semet, F. (2003). Ambulance location and relocation models. *European Journal of Operational Research*, *147*, 451–463.
- Brownlee, J. (2007). *A Note on Research Methodology and Benchmarking Optimization Algorithms*. Technical Report Swinburne University of Technology.
- Chand, S., Hsu, V. N., & Sethi, S. (2002). Forecast, solution and rolling horizons in operations management problems: A classified bibliography. *Manufacturing & Service Operations Management*, *4*, 25–43.
- Chen, Z.-L. (2004). Integrated production and distribution operations: Taxonomy, models and review. In D. Simchi-Levi, S. D. Wu, & Z.-J. Shen (Eds.), *Handbook of Quantitative Supply Chain Analysis: Modeling in the E-business Era* chapter 17. (pp. 711–746). Boston/Dordrecht/London: Kluwer Academic Publishers.
- Chen, Z.-L. (2010). Integrated production and outbound distribution scheduling: Review and extensions. *Operations Research*, *58*, 130–148.
- Coffman, J., E. G., Garey, M. R., & Johnson, D. S. (1983). Dynamic bin-packing. *SIAM Journal on Computing*, *12*, 227–258.
- Cordeau, J.-F. (2007). The dial-a-ride problem : models and algorithms. *Annals of Operations Research*, *153*, 29–46.
- Cordeau, J.-F., Dell’Amico, M., Falavigna, S., & Iori, M. (2014). A rolling horizon algorithm for auto-carrier transportation. *Working paper*, (pp. 1–18).
- Cordeau, J.-F., & Laporte, G. (2003). The dial-a-ride problem (DARP): Variants, modeling issues and algorithms. *4OR Quarterly Journal of the Belgian, French and Italian Operations Research Societies*, *1*, 89–101.

- Crainic, T. G. (2003). Long-haul freight transportation. In R. Hall (Ed.), *Handbook of Transportation Science* International Series in Operations Research and Management Science (pp. 451–516). Boston, MA: Kluwer Academic Publishers.
- Crainic, T. G., & Laporte, G. (1997). Planning models for freight transportation. *European Journal of Operational Research*, *97*, 409–438.
- Dantzig, G. B., & Ramser, J. H. (1959). The truck dispatching problem. *Management Science*, *6*, 80–91.
- Dejax, P., & Crainic, T. (1987). Survey paper - a review of empty flows and fleet management models in freight transportation. *Transportation Science*, *21*, 227–248.
- Desrosiers, J., Dumas, Y., Solomon, M. M., & Soumis, F. (1995). Handbooks in operations research and management science. In M. O. Ball, T. L. Magnanti, C. L. Monma, & G. L. Nemhauser (Eds.), *Network Routing* chapter 2. (pp. 35–139). Elsevier Science B.V. volume 8.
- Dyckhoff, H. (1990). A typology of cutting and packing problems. *European Journal of Operational Research*, *44*, 145–159.
- Eiben, A. E., & Jelasity, M. (2002). A critical note on experimental research methodology in EC. In IEEE (Ed.), *Congress on Evolutionary Computation (CEC 2002)* (pp. 582–587).
- Erengüç, S. S., Simpson, N. C., & Vakharia, A. J. (1999). Integrated production/distribution planning in supply chains: An invited review. *European Journal of Operational Research*, *115*, 219–236.
- Escudero, L. F., Garin, M. A., Merino, M., & Perez, G. (2012). An algorithmic framework for solving large-scale multistage stochastic mixed 0-1 problems with nonsymmetric scenario trees. *Computers & Operations Research*, *39*, 1133–1144.
- Feillet, D., Dejax, P., & Gendreau, M. (2005). Traveling salesman problems with profits. *Transportation Science*, *39*, 188–205.
- Frantzeskakis, L., & Powell, W. (1990). A successive linear approximation procedure for stochastic dynamic vehicle allocation problems. *Transportation Science*, *24*, 40–57.
- Fumero, F., & Vercellis, C. (1999). Synchronized development of production, inventory and distribution schedules. *Transportation Science*, *33*, 330–340.

- Gendreau, M., Laporte, G., & Séguin, R. (1996). Stochastic vehicle routing. *European Journal of Operational Research*, 88, 3–12.
- Gendreau, M., Laporte, G., & Semet, F. (2001). A dynamic model and parallel tabu search heuristic for real-time ambulance relocation. *Parallel Computing*, 27, 1641–1653.
- Gendreau, M., & Potvin, J.-Y. (1998). Dynamic vehicle routing and dispatching. In T. Crainic, & G. Laporte (Eds.), *Fleet Management and Logistics* (pp. 115–126). Boston: Springer.
- Heitsch, H., & Romisch, W. (2007a). A note on scenario reduction for two-stage stochastic programs. *Operations Research Letters*, 35, 731–738.
- Heitsch, H., & Romisch, W. (2007b). Scenario tree modeling for multistage stochastic programs. *Mathematical Programming*, 118, 371–406.
- Hooker, J. N. (1995). *Testing Heuristics: We Have It All Wrong*. Technical Report Carnegie Mellon University.
- Hughes, R. E., & Powell, W. B. (1988). Mitigating end effects in the dynamic vehicle allocation model. *Management Science*, 34, 859–879.
- Hvattum, L. M., Løkketangen, A., & Laporte, G. (2006). Solving a dynamic and stochastic vehicle routing problem with a sample scenario hedging heuristic. *Transportation Science*, 40, 421–438.
- Hvattum, L. M., Løkketangen, A., & Laporte, G. (2007). A branch-and-regret heuristic for stochastic and dynamic vehicle routing problems. *Networks*, 49, 330–340.
- Ichoua, S., Gendreau, M., & Potvin, J.-Y. (2003). Vehicle dispatching with time-dependent travel times. *European Journal of Operational Research*, 144, 379–396.
- Johnson, D. S. (2001). *A Theoretician's Guide to the Experimental analysis of Algorithms*. Technical Report AT and T Labs.
- Kall, P., & Wallace, S. W. (1994). *Stochastic Programming*. (2nd ed.). Chichester: John Wiley and Sons, Inc.
- Keskinocak, P., & Tayur, S. (1998). Scheduling of time-shared jet aircraft. *Transportation Science*, 32, 277–294.

- King, A. J., & Wallace, S. W. (2012). *Modeling with Stochastic Programming*. Springer Series in operations research and financial engineering. New York: Springer.
- Kleywegt, A. J., & Papastavrou, J. D. (1998). The dynamic and stochastic knapsack problem. *Operations Research*, 46, 17–35.
- Kleywegt, A. J., Shapiro, A., & Homem de Mello, T. (2001). The sample average approximation method for stochastic discrete approximation. *SIAM Journal on Optimization*, 12, 479–502.
- Kouvelis, P., & Yu, G. (1997). *Robust Discrete Optimization and its Applications*. Dordrecht, The Netherlands: Kluwer Academic Publisher.
- Laporte, G., & Osman, I. H. (1995). Routing problems: A bibliography. *Annals of Operations Research*, 61, 227–262.
- List, G. F., Wood, B., Nozick, L. K., Turnquist, M. A., Jones, D. A., Kjeldgaard, E. A., & Lawton, C. R. (2003). Robust optimization for fleet planning under uncertainty. *Transportation Research Part E: Logistics and Transportation Review*, 39, 209–227.
- Maes, T., Caris, A., Ramaekers, K., & Janssens, G. (2013). A selective pickup and delivery problem to model carrier decisions in an activity-based freight transportation model. *Working paper*, .
- Malandraki, C., & Daskin, M. S. (1992). Time dependent vehicle routing problems: formulations, properties and heuristic algorithms. *Transportation Science*, 26, 185–200.
- Melo, R. A., & Wolsey, L. A. (2010). Optimizing production and transportation in a commit-to-delivery business mode. *European Journal of Operational Research*, 203, 614–618.
- Meyers, C. A., & Schulz, A. S. (2009). Integer equal flows. *Operations Research Letters*, 37, 245–249.
- Mitrović Minić, S., Krishnamurti, R., & Laporte, G. (2004). Double-horizon based heuristics for the dynamic pickup and delivery problem with time windows. *Transportation Research Part B: Methodological*, 38, 669–685.
- Mitra, S. (2006). *Scenario generation for stochastic programming*. Technical Report Optirisk Systems.



- Mulvey, J. M., Vanderbei, R. J., & Zenios, S. A. (1995). Robust optimization of large-scale systems. *Operations Research*, *43*, 264–281.
- Parragh, S. N., Doerner, K. F., & Hartl, R. F. (2008). A survey on pickup and delivery problems Part II: Transportation between pickup and delivery locations. *Journal für Betriebswirtschaft*, *58*, 21–51.
- Pillac, V., Gendreau, M., Guéret, C., & Medaglia, A. L. (2013). A review of dynamic vehicle routing problems. *European Journal of Operational Research*, *225*, 1–11.
- Powell, W. B. (1986). A stochastic model of the dynamic vehicle allocation problem. *Transportation Science*, *20*, 117–129.
- Powell, W. B. (1988). A comparative review of alternative algorithms for the dynamic vehicle allocation problem. In B. L. Golden, & A. A. Assad (Eds.), *Vehicle routing: Methods and Studies* (pp. 249–291). Amsterdam: Elsevier Science Publishers.
- Powell, W. B. (1996). A stochastic formulation of the dynamic assignment problem, with an application to truckload motor carriers. *Transportation Science*, *30*, 195–219.
- Powell, W. B. (2011). *Approximate Dynamic Programming*. Series in probability and statistics (2nd ed.). Hoboken, New Jersey: John Wiley and Sons, Inc.
- Powell, W. B., Jaillet, P., & Odoni, A. (1995). Handbooks in operations research and management science. In M. O. Ball, T. L. Magnanti, C. L. Monma, & G. L. Nemhauser (Eds.), *Network Routing* chapter 3. (pp. 141–295). Elsevier Science B.V. volume 8.
- Powell, W. B., Simao, H. P., & Bouzaiene-Ayari, B. (2012). Approximate dynamic programming in transportation and logistics: a unified framework. *EURO Journal on transportation and logistics*, *1*, 237–284.
- Powell, W. B., & Topaloglu, H. (2003). Stochastic programming in transportation and logistics. In *Handbooks in Operations Research and Management Science, Volume 10* (pp. 555–636). Amsterdam: Elsevier.
- Powell, W. B., Towns, M. T., & Marar, A. (2000). On the value of optimal myopic solutions for dynamic routing and scheduling problems in the presence of user noncompliance. *Transportation Science*, *34*, 67–85.
- Psaraftis, H. N. (1995). Dynamic vehicle routing : Status and prospects. *Annals of Operations Research*, *61*, 43–164.

- Putterman, M. L. (1994). *Markov Decision Processes Discrete Stochastic Dynamic Programming*. Series in probability and statistics. Hoboken, New Jersey: John Wiley and Sons, Inc.
- Rakotomalala, R. (2011). *Tests de normalité Techniques empiriques et tests statistiques*. Technical Report Université Lumière Lyon 2.
- Rakotomalala, R. (2013). *Comparaison de populations: Tests paramétriques*. Technical Report Université Lumière Lyon 2.
- Rardin, R. L., & Uzsoy, R. (2001). Experimental evaluation of heuristic optimization algorithms: A tutorial. *Journal of Heuristics*, 7, 261–304.
- Razali, N. M., & Wah, Y. B. (2011). Power comparisons of shapiro-wilk, kolgomorov-smirnov, lilliefors and anderson-darling tests. *Journal of Statistical Modeling and Analytics*, 2, 21–33.
- Regan, A. C., Mahmassani, H. S., & Jaillet, P. (1998). Evaluation of dynamic fleet management systems : A simulation framework. In *77th Annual meeting of the transportation research board* (pp. 1–27).
- Reilly, W. J. (1931). *The law of retail gravitation*. New York.
- Rhee, W. T. (1988). Optimal bin packing with items of random sizes. *Mathematics of Operations Research*, 13, 140–151.
- Rockafellar, R. T., & Wets, R. J. (1991). Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of Operations Research*, 16, 119–147.
- Ruml, W. (2010). *The Logic of Benchmarking : A Case Against State-of-the-Art Performance*. Technical Report University of New Hampshire.
- Ruszczyński, A., & Shapiro, A. (2003). Handbooks in operations research and management science. In A. Ruszczyński, & A. Shapiro (Eds.), *Stochastic Programming* chapter 1. (pp. 1–64). Elsevier Science B.V. volume 10.
- Sarmiento, A. M., & Nagi, R. (1999). A review of integrated analysis of production-distribution systems. *IIE Transactions*, 31, 1061–1074.
- Savelsbergh, M. W. P., & Sol, M. (1995). The general pickup and delivery problem. *Transportation Science*, 29, 17–29.
- Secomandi, N. (2000). A rollout policy for the vehicle routing problem with stochastic demands. *Operations Research*, 49, 706–802.

- Sethi, S., & Sorger, G. (1991). A theory of rolling horizon decision making. *Annals of Operations Research*, 29, 387–416.
- Shapiro, A. (2003). Inference of statistical bounds for multistage stochastic programming problems. *Mathematical Methods of Operations Research (ZOR)*, 58, 57–68.
- Shapiro, S. S., & Wilk, M. B. (1965). An analysis of variance test for normality (complete samples). *Biometrika*, 52, 591–611.
- Simao, H. P., Day, J., George, A. P., Nienow, T. G. J., & Powell, W. B. (2009). An approximate dynamic programming algorithm for large-scale fleet management: A case application. *Transportation Science*, 43, 178–197.
- Spivey, M. Z., & Powell, W. B. (2004). The dynamic assignment problem. *Transportation Science*, 38, 399–419.
- Stecke, K. E., & Zhao, X. (2007). Production and transportation integration for a make-to-order manufacturing company with a commit-to-delivery business mode. *Manufacturing & Service Operations Management*, 9, 206–224.
- Tang, J., Yung, K.-L., Ip, A. W. H., & Liu, S. (2007). Synchronized production and transportation planning using subcontracted vehicles in a production-distribution network. *Transportation Planning and Technology*, 30, 113–146.
- Thomas, B. W. (2007). Waiting strategies for anticipating service requests from known customer locations. *Transportation Science*, 41, 319–331.
- Tirado, G., Hvattum, L. M., Fagerholt, K., & Cordeau, J.-F. (2013). Heuristics for dynamic and stochastic routing in industrial shipping. *Computers & Operations Research*, 40, 253–263.
- Tjokroamidjojo, D., Kutanoglu, E., & Taylor, G. D. (2006). Quantifying the value of advance load information in truckload trucking. *Transportation Research Part E: Logistics and Transportation Review*, 42, 340–357.
- Topaloglu, H., & Powell, W. B. (2007). Sensitivity analysis of a dynamic fleet management model using approximate dynamic programming. *Operations Research*, 55, 319–331.
- Toth, P., & Vigo, D. (2001). *The Vehicle Routing Problem*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics.

- Tsilingiris, P. S., & Psaraftis, H. N. (2006). A multi-stage optimization-based approach for the liner shipping problem. In *3rd International Congress on Transportation Research in Greece* (pp. 1–8).
- Tsitsiklis, J. N. (1992). Special cases of traveling salesman and repairman problems with time windows. *Networks*, *22*, 263–282.
- Van Hentenryck, P., & Bent, R. W. (2006). *Online Stochastic Combinatorial Optimization*. Cambridge, Massachusetts: MIT Press.
- Vanderbeck, F. (1999). Computational study of a column generation algorithm for bin packing and cutting stock problems. *Mathematical Programming, Ser A* *86*, 565–594.
- Vanderbeck, F., & Wolsey, L. A. (1995). An exact algorithm for IP column generation. *Operations Research Letters*, *19*, 151–159.
- Vermeij, B., Ahmed, S., Kleywegt, A. J., Nemhauser, G., & Shapiro, A. (2002). *The Sample Average Approximation Method Applied to Stochastic Routing Problems: A Computational Study*. Technical Report Georgia Institute of Technology.
- Vidal, C. J., & Goetschalckx, M. (1997). Strategic production-distribution models: A critical review with emphasis on global supply chain models. *European Journal of Operational Research*, *98*, 1–18.
- Wang, X., & Cheng, T. C. E. (2009). Logistics scheduling to minimize inventory and transport costs. *International Journal of Production Economics*, *121*, 266–273.
- Wäscher, G., Hausner, H., & Schumann, H. (2007). An improved typology of cutting and packing problems. *European Journal of Operational Research*, *183*, 1109–1130.
- Wen, M., Cordeau, J.-F., Laporte, G., & Larsen, J. (2010). The dynamic multi-period vehicle routing problem. *Computers and Operations Research*, *37*, 1615–1623.
- Yang, J., Jaillet, P., & Mahmassani, H. S. (2004). Real-time multivehicle truckload pickup and delivery problems. *Transportation Science*, *38*, 135–148.
- Yung, K.-L., Tang, J., Ip, A. W. H., & Wang, D. (2006). Heuristics for joint decisions in production, transportation, and order quantity. *Transportation Science*, *40*, 99–116.

Zegordi, S. H., Abadi, I. N. K., & Nia, M. A. B. (2010). A novel genetic algorithm for solving production and transportation scheduling in a two-stage supply chain. *Computers and Industrial Engineering*, 58, 373–381.

Zolfagharinia, H., & Haughton, M. (2014). The benefit of advance load information for truckload carriers. *Transportation Research Part E*, 70, 34–54.