EFFICIENT PAC LEARNING FOR EPISODIC TASKS WITH ACYCLIC STATE SPACES AND THE OPTIMAL NODE VISITATION PROBLEM IN ACYCLIC STOCHASTIC DIGRAPHS

A Thesis
Presented to
The Academic Faculty

by

Theologos N. Bountourelis

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Industrial and Systems Engineering

Georgia Institute of Technology
May 2009

EFFICIENT PAC LEARNING FOR EPISODIC TASKS WITH ACYCLIC STATE SPACES AND THE OPTIMAL NODE VISITATION PROBLEM IN ACYCLIC STOCHASTIC DIGRAPHS

Approved by:

Professor Spyros Reveliotis, Advisor
School of Industrial and Systems Engineering
Georgia Institute of Technology

Professor Hayriye Ayhan
School of Industrial and Systems Engineering
Georgia Institute of Technology

Professor David Goldsman
School of Industrial and Systems Engineering
Georgia Institute of Technology

Professor Bert Zwart
School of Industrial and Systems Engineering
Georgia Institute of Technology

Professor Jeff Shamma
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: December 5, 2008

To my family.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor, Dr. Spyros Reveliotis, for his direction and support throughout this research. His patience, devotion, and academic integrity will inspire me long after my graduation. Spyros is not only an advisor but a lifetime friend.

I also want to thank, Dr. Hayriye Ayhan, Dr. David Goldsman, Dr. Jeff Shamma and Dr. Bert Zwart for their willingness to serve on my thesis committee. Furthermore, I am grateful to the School of Industrial and Systems Engineering for the excellent academic environment, and particularly the head of the graduate program, Dr. Gary Parker, for the financial support he provided whenever needed. Furthermore, I would like to thank the National Science Foundation which has supported most part of my research.

I am thankful to my friends that made my life so much beautiful during the course of this study. I particularly want to thank Andrei Prudius for his friendship and his ingenious suggestions on the programming task of my research.

Finally, I want to thank my family back home in Greece, my father Nikolaos, my mother Maria, and my sister Amalia, for their unconditional love and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The first part of this research program concerns the development of customized and easily implementable Probably Approximately Correct (PAC)-learning algorithms for episodic tasks over acyclic state spaces. The defining characteristic of our algorithms is that they take explicitly into consideration the acyclic structure of the underlying state space and the episodic nature of the considered learning task. The first of the above two attributes enables a very straightforward and efficient resolution of the "exploration vs exploitation" dilemma, while the second provides a natural regenerating mechanism that is instrumental in the dynamics of our algorithms. Some additional characteristics that distinguish our algorithms from those developed in the past literature are (i) their direct nature, that eliminates the need of a complete specification of the underlying MDP model and reduces their execution to a very simple computation, and (ii) the unique emphasis that they place in the efficient implementation of the sampling process that is defined by their PAC property.

More specifically, the aforementioned PAC-learning algorithms complete their learning task by implementing a systematic episodic sampling schedule on the underlying acyclic state space. This sampling schedule combined with the stochastic nature of the transitions taking place, define the need for efficient routing policies that will help the algorithms complete their exploration program while minimizing, in expectation, the number of executed episodes. The design of an optimal policy that will satisfy a specified pattern of arc visitation requirements in an acyclic stochastic graph, while minimizing the expected number of required episodes, is a challenging problem, even under the assumption that all the branching probabilities involved

are known a priori. Hence, the sampling process that takes place in the proposed PAC-learning algorithms gives rise to a novel, very interesting stochastic control / scheduling problem, that is characterized as the problem of the *Optimal Node Visitation (ONV)* in acyclic stochastic digraphs. The second part of the work presented herein seeks the systematic modelling and analysis of the ONV problem.

The last part of this research program explores the computational merits obtained by heuristical implementations that result from the integration of the ONV problem developments into the PAC-algorithms developed in the first part of this work. We study, through numerical experimentation, the relative performance of these resulting heuristical implementations in comparison to (i) the initial version of the PAC-learning algorithms, presented in the first part of the research program, and (ii) standard Q-learning algorithm variations provided in the RL literature. The work presented in this last part reinforces and confirms the driving assumption of this research, i.e., that one can design customized RL algorithms of enhanced performance if the underlying problem structure is taken into account.

CHAPTER I

INTRODUCTION

Machine Learning in control applications The field of *Machine Learning* (ML) [29] is a discipline that has received extensive attention in the recent years. It concerns the development of computer programs and systems that improve their performance from their experience. Learning is one of the most significant components of intelligent behavior, and the analysis of methods that would enable a computer to learn has always been a challenging question for the broader scientific community. In the last few decades, researchers have developed the statistical and computational methods needed to establish the theoretical foundations of ML. This effort was concentrated primarily towards the development of the theory that would enable the ML researchers and practitioners to (i) characterize the nature of the computations and experience sufficient for successful learning, and (ii) develop the algorithms that extract information from given data sets and observations based on statistical and computational principles.

Presently, the field of ML has evolved from a field of laboratory experimentation to a field of widely used commercial products. *Learning algorithms* are routinely used in commercialized computer programs designed for a wide spectrum of tasks, ranging from data mining, credit card fraud detection and speech recognition, to autonomous agents that navigate and adapt in their environment through experience. Among these learning applications, there is a wide class of sequential decision making nature, were the objective is to learn a control policy in order to achieve a certain set of goals. This includes, for example, sequential scheduling applications, such as choosing aircraft departure and arrival schedules in order to minimize passenger waiting times and fuel costs. It also includes manufacturing optimization problems where a series of manufacturing decisions have to be made in order to maximize the production volume while minimizing the production costs. In this work, we shall focus to such a control application of ML.

When it comes to control applications of ML, *Reinforcement Learning (RL)* or *Neuro-Dynamic Programming (NPD)* [5] is one of the most conspicuous and prolific areas. RL is concerned with the question of how an autonomous agent can successfully learn control policies to achieve its objectives while interacting with its environment. The basic structure of a typical RL problem, as depicted in Figure 1, involves a *control agent* evolving in a *discrete state space* through the execution of a series of *actions*. The agent observes the current *state* of the environment and chooses to execute upon it some *action* that will change this state. The objectives of the agent are expressed by a *reward function* that assigns a numerical payoff to each action executed by the agent at each state. The task of the agent is to select a sequence of actions in order to maximize some *objective function* of the sequence of the rewards collected over time. In order to decide which action should perform, the agent must take into account the experienced history of states and rewards, in an effort to predict the resultant states and the (expected) immediate reward.

More formally, RL seeks to incrementally compute an optimal policy for problems with a Markov Decision Process (MDP) [2] structure in the absence of complete information about the environment, by taking advantage of the information contained in the observed transitions and the collected rewards. Its main contributions are focused on (i) the development of algorithms that will converge asymptotically to an optimal policy, and (ii) the efficient representation of the information necessary for the efficient learning of the target (near-)optimal policies, especially in the case of tasks with very large discrete state spaces. In the most typical RL implementations, the optimal action selection scheme can be characterized by an *optimal value function*

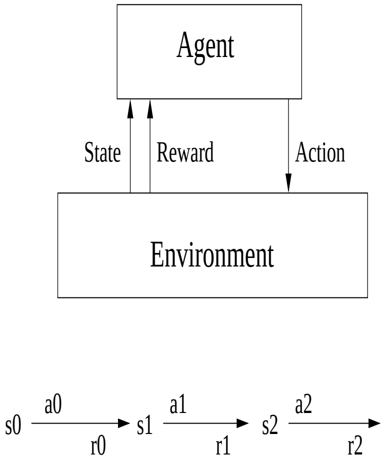

Figure 1: The basic RL framework: The agent chooses an action a_t, at state s_t, and receives a reward r_t as a result of this state-action transition. This produces a sequence of states and rewards as shown in the figure. The agent must use this sequence in order to determine a policy that maximizes a function of the collected rewards.

that associates an (expected) *value* with every state-action pair, such that the optimal actions for any given state are the maximizers of the restriction of this value function to that state. Hence, given an objective function, the RL controller tries to identify an optimal policy by "*learning*" the corresponding optimal value function. More specifically, the learning controller maintains an *estimate* of this value function, that is *initialized* to some arbitrary set of values, and is subsequently *updated* every time that a new reward observation is obtained, in a way that brings the maintained value estimates closer to the value function corresponding to the observed behavior. On the other hand, the running estimate of the optimal value function affects the action selection process itself, since, at every *decision epoch*, actions are selected in a way that seeks to balance the conflicting objectives of (i) maximizing the resulting value, as perceived by the currently available estimate of the optimal value function, and (ii) enhancing the quality of this estimate through further exploration over the state-action space. This trade off is known as the "*exploration vs. exploitation*" *dilemma* in the relevant terminology, and its pertinent resolution constitutes one of the key challenges in the design of RL algorithms with good convergence behavior. In most typical cases it has been addressed by a number of heuristics (”rules of thumb”) that try to adjust a set of probabilities that randomize the action selection process in a way that improves (or seems to improve) the empirical performance of the algorithm.

A particular class of RL algorithms that attempts to learn a value function defined over states and actions and then implement the optimal policy in terms of this value function, is known as *Q-learning*. Q-learning algorithms maintain a set of *Q-factors*, defined for each state-action pair (i, u), such that the *optimal* Q-factor values - to be denoted by $Q^*(i, u)$ - express the *expected (total) value that results by selecting action u at state i and following the optimal policy thereafter* [42, 5]. Obviously, the learning agent can acquire the *optimal action* at every state i, $\pi^*(i)$, by learning Q^*:

$$\pi^*(i) = \arg\max_u \{Q^*(i, u)\}$$

According to the previous discussion of RL algorithms, a Q-learning algorithm will try to develop accurate estimates, $Q(i, u)$, of the optimal Q-factor values, $Q^*(i, u)$, by exploiting the information contained in the sequence of the received rewards. Under some standard assumptions, the Q-learning algorithm can be applied with guaranteed (asymptotic) convergence to optimality [5]. However, the original studies of the Q-learning algorithm did not provide any further information regarding its *rate of convergence*, while some more recent developments indicate that, under some of its typically used configurations, the algorithm might need an exponentially large amount of sampling in order to attain any ϵ-optimal performance [17].

The study of the complexity of the RL problem and algorithms falls into the realm of *computational learning theory (CLT)* [27]. This line of research has tried to establish that, under certain assumptions, the considered RL problems can be resolved by *Probably Approximately Correct (PAC)*-learning algorithms, i.e., by algorithms that, for any given parameters $\delta \in (0, 1/2)$ and $\epsilon > 0$, will execute finitely, using a number of observations that are polynomially related to $1/\delta, 1/\epsilon$, and some other parameters characterizing the "problem size", and, upon termination, with probability $1 - \delta$, they will return, an ϵ-*optimal policy.* For example, the work in [19] considers the discounted payoff case for MDP's equipped with a "reset" action that directs the agent to a set of given starting states and allows the partition of the learning process into *episodes* of fixed length. The learning agent uses an "indirect" or "model-based" approach by observing its environment and updating a model of it. In particular, the learning agent updates empirical estimates of the transition probabilities and immediate rewards every time a particular action is exercised. Subsequently, it uses those estimates to compute an approximation of the optimal policy while exercising actions that are considered the least accurate, according to some accuracy measure.

More recently, the work in [26] introduced the so-called E^3 PAC-learning algorithm, over general MDP's for both discounted and un-discounted cases. This is also an "indirect" algorithm that maintains and updates a partial model on the environment and navigates by computing an optimal policy for it. The departing point of the analysis is the partition of the state space into "known", "visited but unknown" and "non-visited" states and the introduction of the notion of"*mixing-time*" into the context of RL. In the same spirit is the work of [11], where the similar, R-Max algorithm is presented for two-player stochastic games with deterministic rewards. Another line of research has also established PAC-style bounds for the classical multi-armed bandit problem and uses them to derive RL algorithms for MDP's [16]. We should further notice that the algorithms presented in the aforementioned work, except from providing polynomial-type bounds on the sampling required for learning a near-optimal policy, they also incorporate built-in mechanisms that seek to resolve the "exploration vs exploitation" dilemma.

However, it is widely accepted in the RL community that the sampling effort proposed by the PAC-learning algorithms found in the literature, is overly conservative and, primarily, facilitates the analysis of the problem complexity rather than practical implementations. This belief is clearly reflected in the closing statement found in [11]:

"At this point, we have little intuition about how well R-MAX will perform in practice. Practical environments are likely to have many states (exponentially many in the number of state variables) and a large mixing-time. Thus, convergence-time polynomial in these parameters is not likely to be a useful practical guarantee. In that case, algorithms that take into account the structure of the state space are likely to be required, and the generation and analysis of such algorithms is an important area of future work."

In the light of the above discussion, one of the major contributions of this research program is the development of customized and easily implementable PAC-learning algorithms for the RL problem under consideration. The defining characteristic of our algorithms is that they take explicitly into consideration the acyclic structure of the underlying state space and the episodic nature of the considered learning task. The first of the above two attributes enables a very straightforward and efficient resolution of the "exploration vs exploitation" dilemma, while the second provides a natural regenerating mechanism that is instrumental in the dynamics of our algorithms. Some additional characteristics that distinguish our algorithms from those developed in the past literature are (i) their direct nature, that eliminates the need of a complete specification of the underlying MDP model and reduces their execution to a very simple computation, and (ii) the unique emphasis that they place on the efficient implementation of the sampling process that is defined by their PAC property. All these aspects will be revealed and substantiated in the material provided in the rest of this document. We start, however, with some discussion of the practical application that motivated this entire line of work.

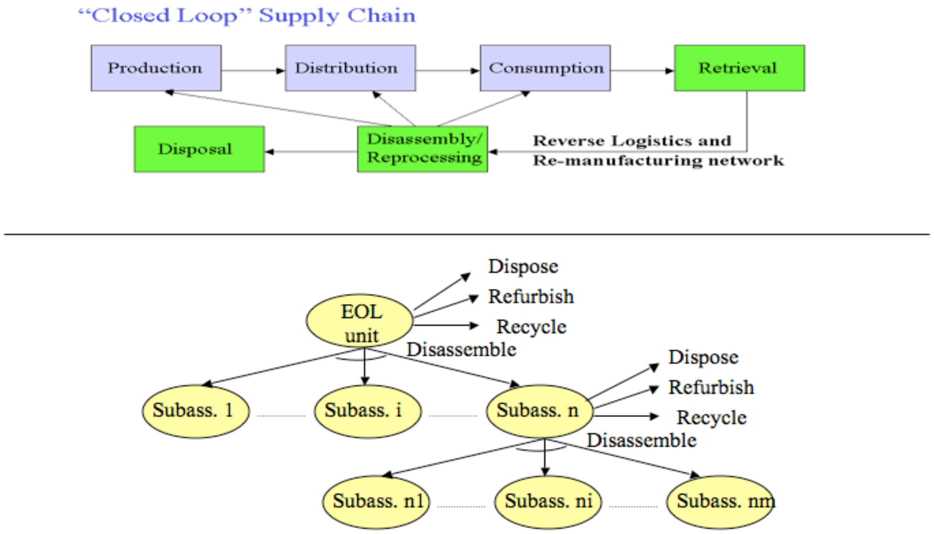

Figure 2: The Optimal Disassembly Planning Problem

The motivating application The work proposed and pursued in this thesis is motivated by an effort to apply RL theory in the emerging area of reverse logistics. Reverse logistic processes are characterized by high levels of uncertainty that differentiate them from the processes encountered in the more traditional forward logistics. The uncertainty present in those processes results (primarily) from the fact that their input stream comes from an unobservable environment -i.e. the end users themselves - and renders many significant attributes of the processed items an unknown parameter in the relevant process design and control problem. Next we concretize these observations by providing a brief description of the "*Optimal Disassembly Planning*" (ODP) problem [37, 36, 35] which has been the main focus of our work.

Presently, under the pressure of environmental, safety, and other societal and economic concerns, a number of industrial sectors have started deploying additional operations that seek to retrieve the corresponding product units at the end of their functional life, extract from them any possible value, and dispose the remaining material in an environmentally friendly manner. Hence, the retrieved product units are brought to designated facilities, known as *re-manufacturing centers*, where those in fairly good condition are refurbished and re-introduced in the supply chain[1], while the remaining are disassembled in order to retrieve potentially re-usable components and/or raw material. Any remaining unusable material is forwarded to landfills and/or incineration (see the upper part of Figure 2). Typically, decisions will be made without complete knowledge of the quality status of the various components and their salvage value. However, the missing information is partially regained through a number of measurements that are performed on the considered artifact and classify it to a certain quality category, that will eventually influence the decision making process.

The classification procedure applied to the initial item will repeat itself on the derived components until all the obtained artifacts have been directed to a particular disposition venue. The dynamics of the aforementioned process are depicted in the lower part of Figure 2: Starting with the initially retrieved unit, the decision maker must select among the disposition options depicted in the figure, until the complete disposition of the unit. It is clear from this description that the decision making process is sequential and it evolves in *episodes* over a *finite*, *acyclic* state space, with each episode corresponding to the complete disposition of a particular product unit. Furthermore, in an optimized setting, the decision taken at each stage must maximize the value extracted from the corresponding artifact. Hence, we can model the ODP problem as a problem of computing an optimal policy for an MDP, under lack of complete *a priori* knowledge of (i) the branching probability distributions determining the evolution of the process state upon the execution of different actions, and (ii) the probability distributions characterizing the immediate rewards returned by the environment as a result of the execution of these actions. In addition, the underlying process evolves in a repetitive, episodic manner, with each episode starting from a

[1] typically directed to a secondary market

well defined initial state and evolving over an acyclic state space.

In this research program we design a customized RL algorithm for the ODP problem presented above. Our starting point is that the acyclic structure and episodic nature that are inherent in this problem, imply that that the information necessary to characterize the optimal policy, flows from the leaf nodes towards the root node. Hence, we should bias the execution of the algorithm so that it facilitates this information flow. The design of this algorithm is at the core of the thesis program outlined next.

The thesis research program As discussed above, the topic of this research program is motivated by the application of Optimal Disassembly Planning (ODP), and it concerns the modeling and management of the uncertainty that is inherent in that application. Hence, the first part of this research program develops a theory of PAC learnability for tasks evolving over discrete acyclic state spaces. Our main objective is to establish that in the case of repetitive tasks evolving episodically over acyclic state spaces, as is the case of the aforementioned ODP problem, one can obtain PAC-learning algorithms that are computationally efficient and effectively implementable. As it will be shown in the subsequent parts of this document, the suggested computational and implementational efficiency stem from (i) the ability to characterize and exploit the flow of the information necessary for the computation of the optimal policy, and (ii) results coming from the area of statistical inference - in particular, the area known as "*ranking and selection* "(RS) [28]- that enable the resolution of the action selection problem arising at the different problem states.

Our work proceeds as follows: First, we characterize the environment in which the considered learning task takes place. Then, we propose an algorithm that, for any given parameters $\delta \in (0, 1/2)$ and $\epsilon > 0$, with probability $1 - \delta$, will return a ϵ-optimal policy in a number of episodes polynomially related to $1/\delta, 1/\epsilon$ and some other parameters characterizing the problem size. This algorithm bases the identification of the returned policy on a number of observations taken across all the state-action pairs of the underlying acyclic state space. More specifically, the proposed algorithm starts from the *terminal* states of the underlying acyclic state space, and maintains a "*frontier set of actively explored states*", for which it will try to learn and assign an "*apparent optimum action*" when a pre-defined sampling program is complete. A state reaching that point is declared "*fully explored*" and is removed from the set of actively explored states. On the other hand, a state enters the set of actively explored states when all of its successors become fully explored. In the proposed scheme, sampling a state-action pair during some episode corresponds to collecting the total reward obtained by the learning agent when selecting the particular action, and subsequently following the apparent optimum actions for the fully explored states that will be visited until the end of the episode. The amount of sampling for each state-action pair is obtained by applying results coming from the RS theory. The algorithm pursues the above exploration pattern for a pre-defined number of episodes, and terminates either upon the selection of an apparent optimum action for the initial state or upon the exhaustion of the specified budget of episodes. Hence, the above algorithm can fail either (i) because the algorithm failed to materialize the sampling program within the allocated number of episodes, or (ii) because the objective value of the returned policy differs from the optimal value by more than ϵ.

It is evident from the above description that the notions of *actively explored* states and the *visitation requirements* associated with them, when combined with the stochastic nature of the transitions taking place in the underlying state space, define the need for efficient routing policies that will help the algorithm complete its exploration program while minimizing the number of executed episodes. As we will show in the following developments, the design of an optimal policy that will satisfy a specified pattern of arc visitation requirements in an acyclic stochastic graph, while minimizing the expected number of required episodes, is a challenging problem, even under the assumption that all the branching probabilities involved are known a priori. In other words, the sampling process that takes place in the proposed PAC-learning algorithm gives rise to a novel, very interesting stochastic control / scheduling problem, that will be characterized as the problem of the *Optimal Node Visitation (ONV)* in acyclic stochastic digraphs.

Hence, the second part of the work presented herein, seeks the systematic modelling and analysis of the ONV problem, which can be abstracted as follows: *Given a stochastic, acyclic, connected digraph with a single source node and a control agent that repetitively traverses this graph, each time starting from the source node, we want to define a control policy that will enable this agent to visit each of the graph terminal nodes a prespecified number of times while minimizing the expected number of graph traversals.* In the this research program, first we provide a detailed formulation of the ONV problem as a specially structured MDP. It is shown that the problem admits a straightforward Stochastic Shortest Path formulation but the state space of this formulation grows exponentially with respect to the problem size. Therefore we introduce a problem relaxation that further implies a randomized policy which is implementable in polynomial time and asymptotically optimal; more specifically, the ratio of the value of this policy to the value of the optimal policy converges to unity, as the non-zero node visitation requirements grow uniformly to infinity. Furthermore, the proposed randomized policy admits a closed form performance evaluation, and this capability subsequently enables (i) a more detailed analysis of the asymptotic performance of the policy, and (ii) its embedding in suboptimal control schemes that can lead to even more enhanced performance. In particular, we propose some adaptive implementation schemes of the aforementioned randomized policies that experimentally are found to have a very attractive performance while they maintain computational efficiency. Those results stem from the special structure of the considered problem and the application of ideas and results coming from MDP suboptimal control [4, 2].

In a subsequent step, we complement the empirical results on the very high efficiency of our adaptive suboptimal policies for the ONV problem, with the analytical treatment of the performance of a particular adaptive implementation that is suggested by the problem relaxation. We prove that, for a wide range of visitation requirement choices, the expected performance of this adaptive policy is within a constant factor from the expected optimal cost as the visitation requirements grow uniformly to infinity. To deliver this result, we also develop an alternative characterization of the problem relaxation which provides useful geometric insights for the underlying dynamics.

The next part of our work extends many of the aforementioned results for the ONV problem, to some additional variations that are defined from the following two assumptions: (i) The control agent is replaced by *tokens* that traverse the graph and can "*split*" during certain transitions to a number of (sub-)tokens, allowing thus the satisfaction of many visitation requirements during a single graph traversal. (ii) There are additional visitation requirements attached to the *internal* graph nodes, which, however, can be served only when the visitation requirements of their successors have been fully met. Notice that these new assumptions accommodate the more complex version of the PAC-learning algorithm pursued in our work, where a certain task can split into a number of subtasks that execute in parallel, as in the case with the ODP problem. It is shown that, similar to the basic ONV case, the new problem variations admit a Stochastic Shortest Path formulation with a state space that grows exponentially with respect to the size of the problem-defining graph and the number of its target nodes, and that it is possible to obtain a computationally efficient suboptimal policy for each of those formulations by exploiting the information provided in the optimal solution of the continuous relaxation of the original problem. Furthermore, the resulting policies are asymptotically optimal. A final contribution of our work with respect to the ONV problem is a set of results that place the considered ONV variations in the complexity hierarchy established by the theory of computational complexity and establish their relationship to other well known stochastic scheduling problems.

The last part of this research program explores the computational merits obtained by heuristical implementations that are obtained by integrating the ONV problem developments into the initial PAC framework that was outlined above and is further detailed in the rest of this document. We study, through numerical experimentation, the performance of the resulting heuristical implementations, in comparison to (i) the initial version of the PAC-learning algorithm, as presented in the first part of the research program, and (ii) standard Q-learning algorithm variations provided in the RL literature. The work presented in this last part, reinforces and confirms the driving assumption of this research, i.e., that one can design customized RL algorithms of enhanced performance if the underlying problem structure is taken into account.

Recapitulating the above discussion, we summarize the main tasks of our research program as follows:

- Development of a learning algorithm with a PAC capability for tasks evolving episodically over discrete acyclic state spaces.

- Formal characterization of the basic ONV problem and a number of its variations as specially structured stochastic shortest path problems; investigation of their computational complexity and their relationship to other well known stochastic scheduling problems; development of a series of computationally efficient and asymptotically optimal policies for these variations, that seek to establish a systematic trade-off between operational efficiency and computational tractability.

- Integration of the initial PAC algorithm with the ONV problem developments, that aim at the specification of more efficient sampling schemes for the algorithm. Investigation of the performance of the derived learning algorithms and comparison with standard Q-learning algorithm variations available in the RL literature.

The rest of this document is organized as follows: Chapter 2 presents the development of the PAC-learning algorithm for tasks with discrete acyclic state spaces. Chapters 3 and 4 are concerned with the formal characterization of the ONV problem and the relevant results. Subsequently, Chapter 5 introduces the further variations of the ONV problem and it extends the results of Chapter 3 to these new variations. Chapter 6 is concerned with the investigation of the computational complexity of the ONV problem variations and their relationship to other well known stochastic scheduling problems. Finally, Chapter 7 is concerned with the integration of the previous developments into a practical learning algorithm for the ODP problem context. It also provides some closing remarks for this work.

CHAPTER II

EFFICIENT PAC-LEARNING FOR EPISODIC TASKS WITH ACYCLIC STATE SPACES

The main objective of the first part of the presented research program, is to establish that in the case of repetitive tasks evolving over acyclic state spaces, as is the case of the ODP problem, one can obtain PAC-learning algorithms that are computationally efficient and effectively implementable. The problem considered in this chapter can be stated as follows: A certain task is executed repetitively in an episodic manner. Each episode starts from a well-defined initial state and evolves sequentially over an acyclic state space. At each state, the task evolution is the result of an action that is selected by a controlling agent; the execution of this action determines probabilistically the subsequent state, but it also results in a certain reward for the controlling agent. The returned reward can be of arbitrary sign,[1] and it is a random quantity drawn from some bounded[2] general distribution that is dependent on the current state and the commanded action. The agent's objective is to select the actions to be commanded at the different states of the underlying task in a way that maximizes the expected total reward to be collected over any single episode. However, the initial knowledge of the controlling agent about the underlying task and its operational environment is limited to (i) the set of the environmental states, (ii) the available actions at each state, (iii) an upper bound for the magnitude of the experienced rewards, and (iv) a set of action sequences that can lead with positive probability to each of the environmental states, together with a lower bound for the corresponding state-reaching probabilities. The

[1] A negative reward can be considered as a cost.

[2] i.e., a distribution with bounded support

agent knows neither (v) the branching probability distributions that determine the state evolution as a result of the different actions, nor (vi) the type and the moments of the distributions determining the experienced rewards. Hence, the agent must compute a (near-)optimal policy for the aforestated objective, while observing and appropriately compiling the results of its decisions on the operational environment.

The rest of the chapter is organized as follows: Section 2.1 provides a rigorous characterization of the learning problem considered in this work, and Section 2.2 presents a notion of computational efficiency that is pertinent to machine learning algorithms. Subsequently Section 2.3 proposes a methodology for developing efficient learning algorithms for the learning task under consideration, and finally, Section 2.4 concludes the chapter by summarizing the key contributions of this first part of the research program and by providing with some further comments.

2.1 A formal characterization of the considered learning problem

We begin our discussion of the learning problem of interest in this work, by providing a formal characterization of the *"environment"* in which the considered learning task will take place. This environment essentially constitutes a *discrete-time Markov Decision Process (DT-MDP)* [5], the structure of which is completely characterized by a quadruple:

$$\mathcal{E} = (X, \mathcal{A}, \mathcal{P}, \mathcal{R}) \tag{1}$$

where the components X, $\mathcal{A}$, $\mathcal{P}$, and $\mathcal{R}$ are further defined as follows:

- X is the finite set of the process *states*, and it is partitioned into a sequence of *"layers"*, X^0, X^1, $\ldots$, X^L. $X^0 = \{x^0\}$ and defines the *initial state* of the process, while states $x \in X^L$ are its *terminal states*.

- $\mathcal{A}$ is a set function defined on X, that maps each state $x \in X$ to the finite, non-empty set $\mathcal{A}(x)$, comprising all the decisions / actions that are feasible in

x. It is further assumed that for $x \neq x'$, $A(x) \cap A(x') = \emptyset$. For subsequent development, we also define $|\bar{A}| \equiv max_{x \in X} |\mathcal{A}(x)|$.

- $\mathcal{P}$ is the *state transition function*, defined on $\bigcup_{x \in X} \mathcal{A}(x)$, that associates with every action a in this set a discrete probability distribution $p(\cdot; a)$, that is unknown to the learning agent. The support sets, $\mathcal{S}(a)$, of the distributions $p(\cdot; a)$ are subsets of the state set X that satisfy the following property: For any given action $a \in \mathcal{A}(x)$ with $x \in X^i$ for some $i = 0, \ldots, L-1$, $\mathcal{S}(a) \subseteq \bigcup_{j=i+1}^{L} X^j$; for $a \in \mathcal{A}(x)$ with $x \in X^L$, $\mathcal{S}(a) = X^0$. In words, the previous assumption implies that the environment operates in an *episodic* fashion, where each episode is an *acyclic* traversal of the underlying state space from the initial state to a terminal state. Furthermore, it is assumed that every state $x \in X$ can be reached from the initial state x^0 with positive probability, through some sequence of actions, and that the learning agent knows (i) at least one such sequence of actions for every state, and also (ii) a lower bound, $\underline{q}$, for the corresponding state-reaching probabilities.[3]

- $\mathcal{R}$ is the *immediate reward function*, defined on $\bigcup_{x \in X} \mathcal{A}(x)$, that associates with each action a in this set a probability distribution, $\mathcal{D}(\mu(a), v(a))$, characterizing the *immediate reward* experienced by the *learning agent* every time that action a is selected and executed. The parameters $\mu(a)$ and $v(a)$ denote respectively the *mean* and the *maximum possible magnitude* of the rewards drawn from the distribution $\mathcal{D}(\mu(a), v(a))$, and they take finite values for every a. On the other hand, the only information initially available to the learning agent about $\mathcal{R}()$ is an upper bound $\bar{v}$ for the quantities $v(a)$, $a \in \bigcup_{x \in X} \mathcal{A}(x)$.

[3] The reader should notice that this characterization of the state transition function ignores the *multi-threading* effect that results from the disassembly operation in the ODP problem. We have opted to omit this particular feature from the basic positioning of the problem considered in this chapter, since it would complicate the exposition of the main ideas without contributing substantially to the underlying analysis. The extension of the developed results in order to accommodate this particular feature is very straightforward and it is briefly outlined in the concluding section.

The learning agent controls the selection of the action to be executed at every state of the environment. More specifically, starting from the initial state x^0 at period $t = 0$, and at every consecutive period $t = 1, 2, 3, \ldots$, the agent (i) observes the current state of the environment, x_t, (ii) selects an action $a_t \in \mathcal{A}(x_t)$ and commands its execution upon the environment, and subsequently (iii) it experiences a reward r_t, where the latter is a random sample drawn from the distribution $\mathcal{D}(\mu(a_t), v(a_t))$. Hence, at the end of some period t, the agent has experienced an entire *"history"*

$$x_0, a_0, r_0, x_1, a_1, r_1, \ldots, x_t, a_t, r_t \tag{2}$$

The ultimate objective of this agent is to determine an action selection scheme – or, in the relevant terminology, a *policy* – π^*, that maps each state $x \in X$ to an action $\pi^*(x) \in \mathcal{A}(x)$ in a way that maximizes the expected total reward experienced over any single episode. Letting M denote the (random) duration of any single episode in terms of number of periods, t, the aforestated objective can be formally expressed as follows:

$$\pi^* = \arg\max_{\pi} E_\pi[\sum_{t=0}^{M} r_t | x_0 = x^0] \tag{3}$$

In the above equation, the expectation $E_\pi[\cdot]$ is taken over all the possible episode realizations under policy π. It is easy to see that, in the considered operational context, an optimal policy π^* can be obtained through the following simple recursion:

$\forall x \in X^L$,

$$\begin{aligned} \pi^*(x) &:= \arg\max_{a \in \mathcal{A}(x)} \{\mu(a)\} & (4) \\ V^*(x) &:= \mu(\pi^*(x)) & (5) \end{aligned}$$

$\forall x \in X^i,\ i = L-1, \ldots, 0,$

$$\begin{aligned}\pi^*(x) &:= \arg\max_{a \in \mathcal{A}(x)} \{\mu(a) + \\ &\qquad \sum_{x' \in \mathcal{S}(a)} p(x'; a) \cdot V^*(x')\} \qquad (6)\\ V^*(x) &:= \mu(\pi^*(x)) + \\ &\qquad \sum_{x' \in \mathcal{S}(\pi^*(x))} p(x'; \pi^*(x)) \cdot V^*(x') \qquad (7)\end{aligned}$$

The quantity $V^*(x)$ appearing in the above recursion is known as the *(optimal) value* of the corresponding state x, and it expresses the expected return to be collected by the learning agent in a single episode, when it starts from state x and follows the optimal policy π^* until the completion of the episode.

Yet, despite the fact that it provides a pertinent characterization of the optimal policy, the algorithm defined by Equations 4–7 is not directly applicable in the considered problem context, since the quantities $\mu(a)$ and $p(x; a)$, $x \in X,\ a \in \mathcal{A}(x)$, are not initially known to the agent. Furthermore, as discussed in the introductory chapter, the application of standard RL algorithms, like Q-learning [42], guarantees only asymptotic convergence to optimality, and to the best of our knowledge, currently there are no formal results characterizing the convergence rate of the "standard" Q-learning algorithm to an optimal policy. At the same time, Q-learning is notorious for rather slow convergence. Hence, in the rest of this chapter, we seek to exploit the special structure of the considered problem in order to derive customized learning algorithms with proven convergence and complexity properties. However, before delving into this discussion, we shall formalize the notions of computational complexity and efficiency to be employed in the considered problem context.

2.2 Efficient PAC learnability

In computational learning theory [27], a learning algorithm is characterized as *probably approximately correct (PAC)*, if, upon its completion, it provides with probability at least $(1-\delta)$, an approximation, $\hat{h}$, of the target concept h^*, that differs from h^* by an *"error"*, $\texttt{err}(\hat{h})$, less than or equal to ε. In this definition, both δ and ε are externally specified parameters, and the quantity $\texttt{err}(\hat{h})$ is appropriately specified from the attributes of the target concept h^*. In addition, a PAC algorithm is said to be *efficient*, if it executes in a number of steps that is a polynomial function of $1/\delta$, $1/\varepsilon$, and some additional parameters that characterize the complexity of the learning task and the *"size"* of the target concept h^*.

In the context of the learning problem considered in this work, the target concept is any optimal policy π^*, and a natural solution space is the set Π consisting of all the *deterministic* policies π that map each state $x \in X$ to a unique action $\pi(x) \in \mathcal{A}(x)$. For these policies, we define:

$$\texttt{err}(\pi) = E_{\pi^*}[\sum_{t=0}^{M} r_t | x_0 = x^0] - E_{\pi}[\sum_{t=0}^{M} r_t | x_0 = x^0] \tag{8}$$

On the other hand, the complexity of the considered learning problem is measured by the magnitude of the environmental parameters $|X|$, $|\bar{A}|$, L, $\bar{v}$, and $1/\underline{q}$, respectively characterizing the size of the task state space, the extent of choice at each state, the length of the decision sequences, the magnitude of the collected rewards, and the difficulty of accessing the various states of the task state space.

In the light of the above characterizations, *an efficient PAC algorithm for the learning problem considered in this work* is defined as follows:

Definition 1: An *efficient PAC* algorithm for the RL problem considered in this work is an algorithm that, for any environment $\mathcal{E} = (X, \mathcal{A}, \mathcal{P}, \mathcal{R})$ and parameters $\delta \in (0, 1/2)$ and $\varepsilon > 0$,

i. will execute in a finite number of steps, that is polynomial with respect to $1/\delta$, $1/\varepsilon$, $|X|$, $|\bar{A}|$, L, $\bar{v}$ and $1/\underline{q}$, and

ii. upon its completion, will return, with probability at least $1-\delta$, a policy $\hat{\pi}$ with $\texttt{err}(\hat{\pi}) \leq \varepsilon$. □

The next section discusses the special structure of the considered RL problem that enables the development of an efficient PAC algorithm for it.

2.3 Developing an efficient PAC learning algorithm for the considered RL problem

The PAC algorithm proposed in this work for the RL problem of Section 2.1 is strongly based upon the following fundamental observation:

Observation 1: According to Equations 4–7, that characterize the structure of an optimal policy, π^*, for the RL problem considered in this work, we can assess the optimal value $V^*(x)$ of any state $x \in X$ only when we have already computed the optimal values $V^*(x')$ for all the states $x' \in X$ that constitute successor states of x through some action $a \in \mathcal{A}(x)$. Therefore, any learning agent that will base the identification of an optimal policy on the computation of the optimal value function V^*, must acquire its knowledge proceeding from the terminal states, $x \in X^L$, of the underlying state space to the initial state, x^0. This observation subsequently suggests the following basic structure for the proposed algorithm:

- Starting with the set of terminal states, X^L, the proposed algorithm maintains a *"frontier set of actively explored states"*, for which it tries to learn the optimal action $\pi^*(x)$.

- An actively explored state x is assigned an *"apparent optimum action"*, $\hat{\pi}(x)$, when it satisfies a criterion to be defined in the following. At that point, x is declared as *"fully explored"*, and it leaves the "frontier" set, while action $\hat{\pi}(x)$ is the action to be executed at state x, any time that this state is visited until the completion of the algorithm.

- On the other hand, a state $x \in \bigcup_{i=0}^{L-1} X^i$ becomes an actively explored state as soon as all the states $x' \in X$ that constitute successor states of x through some action $a \in \mathcal{A}(x)$, become fully explored.

- The algorithm pursues the above exploration pattern for a pre-defined number of episodes, N, and it terminates either upon the selection of an action $\hat{\pi}(x^0)$, for the initial state x^0, or upon the depletion of the episode budget, N. In the first case, the algorithm returns the computed policy $\hat{\pi}(x)$, $\forall x \in X$, while in the second case it reports failure. □

Notice that the algorithm defined in Observation 1 can fail either (i) because it did not manage to determine a complete policy $\hat{\pi}(x)$, $\forall x \in X$, within the specified episode budget, N, or (ii) because the chosen policy $\hat{\pi}(x)$, $\forall x \in X$, had an error $\texttt{err}(\hat{\pi}) > \varepsilon$. Letting δ^I and δ^{II} denote the respective probabilities of failure according to the modes (i) and (ii), we obtain, by Boole's inequality, that

$$\delta \leq \delta^I + \delta^{II} \tag{9}$$

where δ denotes the total probability of failure of the considered algorithm. Therefore, we can guarantee a success probability of at least $1 - \delta$ for this algorithm, by picking δ^I and δ^{II} such that $\delta^I + \delta^{II} = \delta$. For expository purposes, in the following we shall assume that $\delta^I = \delta^{II} = \delta/2$.

Generally speaking, the proposed algorithm will fail according to mode (ii) only because it was not able to assess adequately the consequences of its various actions upon the environment, which further translates to inadequate observation and exploration of these consequences. Hence, the ability of the proposed algorithm to satisfy a particular PAC requirement, expressed in terms of the tolerated error ε and the failing probability $\delta^{II} = \delta/2$, will depend on the establishment of a pertinent and adequate exploration scheme. On the other hand, in order to prevent failure according to mode (i), the proposed exploration scheme must be *efficient*, i.e., there must exist an episode budget, N, that is polynomially related to $1/\delta$, $1/\varepsilon$, $|X|$, $|\bar{A}|$, L, $\bar{v}$ and $1/\underline{q}$, and will permit the execution of the aforementioned exploration scheme with probability at least $1 - \delta^I = 1 - \delta/2$. We address each of these two issues below.

2.3.1 Establishing the PAC capability of the proposed algorithm

In this section we establish that the aforestated PAC requirement for the algorithm of Observation 1 – i.e., the requirement that the policy $\hat{\pi}(x)$, $\forall x \in X$, returned by the algorithm defined in Observation 1, will have an error $\texttt{err}(\hat{\pi}) \leq \varepsilon$ with probability at least $1 - \delta^{II} = 1 - \delta/2$ – can be replaced by the following more local requirement: At every actively explored state, x, the algorithm must be able to identify, with a certain probability of success, $1 - \delta(x)$, an action $\hat{\pi}(x) \in \mathcal{A}(x)$ with an expected total reward that differs by at most $\varepsilon(x)$ from the *maximal* expected total reward that can be collected by performing some action $a \in \mathcal{A}(x)$, while in state x, and subsequently following the pre-determined policy, $\hat{\pi}$, until the environment resets itself to the initial state x^0. This localized version of the PAC policy resolution problem can be addressed through results from statistical inference. One particular approach is that presented in the next theorem, which constitutes a generalization of Bechhofer's *"indifference-zone" (IZ)* approach for the "ranking-and-selection" (R&S) problem [1], to populations with bounded general distributions.

Theorem 1 *Suppose that there are k populations distributed according to some bounded general distributions with respect to some attribute of interest, and that $\bar{v}$ constitutes a known uniform absolute bound for this attribute. Furthermore, suppose that the means μ_i of these k populations are unknown, and that it is desired to determine which population has the largest mean μ_i. In particular, the experimenter specifies a* confidence level $1 - \delta$ *and an* "indifference" parameter ε *with the requirement that*

$$\mu_{[k]} - \mu_{[k-1]} \geq \varepsilon \implies PCS \geq 1 - \delta \tag{10}$$

where $\mu_{[1]} \leq \ldots \leq \mu_{[k]}$ are the ordered population means and PCS is the probability for correct selection*, i.e., the probability of correctly identifying the population with the largest mean μ_i.*

Then, this problem can be resolved by:

1. *taking a sample from each of the k populations, of* size
$$n = \lceil \frac{4\bar{v}^2}{\varepsilon^2} \ln(\frac{k-1}{\delta})\rceil \tag{11}$$
2. *computing the corresponding sample means $\bar{X}_i,\ i = 1, \ldots, k$, and*
3. *selecting the population with the largest sample mean.*

Proof: The probability for correct selection, PCS, can be bounded as follows:

$$\begin{aligned} PCS &= \Pr\{\text{select } k\} & (12)\\ &= \Pr\{\bar{X}_k > \bar{X}_i,\ \forall i \neq k\} & (13)\\ &= \Pr\{\bar{X}_i - \bar{X}_k - (\mu_i - \mu_k) < -(\mu_i - \mu_k),\ \forall i \neq k\} & (14)\\ &\geq \Pr\{\bar{X}_i - \bar{X}_k - (\mu_i - \mu_k) < \varepsilon,\ \forall i \neq k\} & (15)\end{aligned}$$

Equation 12 above holds by the definition of PCS and the assumption that $\mu_{[k]} = \mu_k$ and $\mu_k - \mu_i \geq \varepsilon,\ \forall i \neq k$, Equation 13 holds from the definition of the selection procedure provided in Theorem 1 (c.f. Step 3), and Equation 15 holds from the fact that $\mu_k - \mu_i \geq \varepsilon,\ \forall i \neq k$. Setting

$$E_i \equiv \{\bar{X}_i - \bar{X}_k - (\mu_i - \mu_k) < \varepsilon\},\ \forall i \neq k \tag{16}$$

and applying the Bonferroni inequality, we get:

$$\begin{aligned} PCS &\geq \Pr(\bigcap_{i=1}^{k-1} E_i) & (17)\\ &\geq 1 - \sum_{i=1}^{k-1}[1 - \Pr(E_i)] & (18)\\ &= 1 - \sum_{i=1}^{k-1}\Pr\{\bar{X}_i - \bar{X}_k - (\mu_i - \mu_k) \geq \varepsilon\} & (19)\end{aligned}$$

Recognizing that each single observation X_i will belong in the interval $[-\bar{v}, \bar{v}]$, for all $i \in \{1, \ldots, k\}$, and using the relevant Hoeffding inequality for the difference of two sample means (c.f. [22], Eq. 2.7) , we get:

$$\Pr\{\bar{X}_i - \bar{X}_k - (\mu_i - \mu_k) \geq \varepsilon\} \ \leq \ e^{-n\varepsilon^2/(2\bar{v})^2} \tag{20}$$

When combined with Equations 17–19, Equation 20 implies that

$$PCS \;\geq\; 1-(k-1)e^{-n\varepsilon^2/(2\bar{v})^2} \tag{21}$$

Setting

$$1-(k-1)e^{-n\varepsilon^2/(2\bar{v})^2} \geq 1-\delta \tag{22}$$

and solving for n, we get:

$$n \geq \frac{4\bar{v}^2}{\varepsilon^2}\ln(\frac{k-1}{\delta}) \tag{23}$$

Equation 23 implies the selection of the sample size n stated in Theorem 1, and concludes the proof. □

In order to completely characterize the application of Theorem 1 in the context of the algorithm outlined in Observation 1, we need to specify the parameters $\delta(x)$, $\varepsilon(x)$ and $\bar{v}(x)$ to be employed at each state $x \in X$. The pricing of $\bar{v}(x)$ is a direct consequence of the acyclic structure presumed for the underlying task state space.

Observation 2: For any given state $x \in X^l$, $l = 0, 1, \ldots, L$, any action sequence leading the environment from state x to the initial state x^0 will contain at most $L - l + 1$ actions. As a result, the magnitude of the total reward collected by the execution of any such action sequence will be bounded from above by $(L-l+1)\bar{v}$. □

The pricing of the remaining parameters $\varepsilon(x)$ and $\delta(x)$ is performed through the following two lemmas:

Lemma 1 *Under the assumption that $PCS = 1$ at every state $x \in X$, the policy $\hat{\pi}$, returned by the algorithm defined in Observation 1, will have* $\texttt{err}(\hat{\pi}) \leq \varepsilon$*, if the "indifference" parameter $\varepsilon(x)$, employed during the implementation of Theorem 1 at each state $x \in X$, is set to a value $\varepsilon(x) \leq \varepsilon/(L+1)$.*

Proof: Let $V^{\hat{\pi}}(x)$ denote the value of state x under policy $\hat{\pi}$, i.e., the expected total reward to be obtained by starting from state $x \in X$ and following policy $\hat{\pi}$

until the environment resets itself to the initial state x^0. We shall prove Lemma 1 by establishing the stronger result that

$$\forall l = 0, \ldots, L,\ \forall x \in X^l, \quad V^*(x) - V^{\hat{\pi}}(x) \ \leq\ \frac{L - l + 1}{L + 1} \cdot \varepsilon \tag{24}$$

This last result is proven with induction on l, starting from $l = L$ and proceeding to $l = 0$. The satisfaction of the base case for $l = L$ is immediately implied by the definition of $V^*(x)$ and $V^{\hat{\pi}}(x)$ for $x \in X^L$ (e.g., c.f. Equation 5) and the proposed value for the "indifference" parameter, $\varepsilon/(L + 1)$. Next, suppose that the inequality of Equation 24 holds true for $x \in \bigcup_{i=l}^{L} X^i$. We shall show that it also holds true for $x \in X^{l-1}$. For this, consider a state $x \in X^{l-1}$ and let $\hat{\pi}(x)$ denote the action selected by policy $\hat{\pi}$. Also, let $Q^{\hat{\pi}}(x, a)$ (resp., $Q^*(x, a)$) denote the expected total reward obtained by initializing the environment at state x, executing the action $a \in \mathcal{A}(x)$ in that state, and following the policy $\hat{\pi}$ (resp., an optimal policy π^*) thereafter, until the environment resets itself to state x^0. Finally, let $\hat{a} = \arg\max_{a \in \mathcal{A}(x)}\{Q^{\hat{\pi}}(x, a)\}$ and $a^* = \arg\max_{a \in \mathcal{A}(x)}\{Q^*(x, a)\}$. Then,

$$\begin{aligned}
V^*(x) - V^{\hat{\pi}}(x) &= Q^*(x, a^*) - Q^{\hat{\pi}}(x, \hat{\pi}(x)) && (25)\\
&= Q^*(x, a^*) - Q^{\hat{\pi}}(x, \hat{a}) + \\
&\quad\ Q^{\hat{\pi}}(x, \hat{a}) - Q^{\hat{\pi}}(x, \hat{\pi}(x)) && (26)\\
&\leq Q^*(x, a^*) - Q^{\hat{\pi}}(x, \hat{a}) + \frac{\varepsilon}{L + 1} && (27)\\
&= Q^*(x, a^*) - Q^{\hat{\pi}}(x, a^*) + Q^{\hat{\pi}}(x, a^*) \\
&\quad\ -Q^{\hat{\pi}}(x, \hat{a}) + \frac{\varepsilon}{L + 1} && (28)\\
&\leq Q^*(x, a^*) - Q^{\hat{\pi}}(x, a^*) + \frac{\varepsilon}{L + 1} && (29)
\end{aligned}$$

Equation 25 is an immediate consequence of the definitions of $V^*(x)$, $V^{\hat{\pi}}(x)$, $Q^*()$, $Q^{\hat{\pi}}()$, a^*, and $\hat{\pi}(x)$. Equation 27 results from the definition of the "indifference" parameter $\varepsilon(x)$ for state x and the assumption that $PCS = 1$ at every node $x \in X$.

Finally, Equation 29 results from the definition of $\hat{a}$. We also have that:

$$\begin{aligned} Q^*(x, a^*) - Q^{\hat{\pi}}(x, a^*) &= \sum_{x' \in \mathcal{S}(a^*)} p(x'; a^*) \cdot \\ &\quad [V^*(x') - V^{\hat{\pi}}(x')] && (30) \\ &\leq [\sum_{x' \in \mathcal{S}(a^*)} p(x'; a^*)] \cdot \\ &\quad \frac{L - l + 1}{L + 1} \cdot \varepsilon && (31) \\ &= \frac{L - l + 1}{L + 1} \cdot \varepsilon && (32) \end{aligned}$$

Equation 30 results from the definition of $Q^*()$ and $Q^{\pi}()$, Equation 31 results from the induction hypothesis, and Equation 32 results from the fact that $\mathcal{S}(a)$ is the support set for the discrete distribution $p(\cdot; a)$. The combination of Equations 29 and 32 gives:

$$\begin{aligned} V^*(x) - V^{\hat{\pi}}(x) &\leq \frac{L - l + 1}{L + 1} \cdot \varepsilon + \frac{1}{L + 1} \cdot \varepsilon && (33) \\ &= \frac{L - (l - 1) + 1}{L + 1} \cdot \varepsilon && (34) \end{aligned}$$

and completes the inductive argument for the proof of Equation 24.

Finally, the proof of Lemma 1 is established by applying Equation 24 for $l = 0$. □

Lemma 2 *The policy $\hat{\pi}$, returned by the algorithm defined in Observation 1, will have $\texttt{err}(\hat{\pi}) \leq \varepsilon$, with probability at least $1 - \delta/2$, if, during the implementation of Theorem 1 at each node $x \in X$,*

1. *the "indifference" parameter $\varepsilon(x)$ is set to $\varepsilon(x) = \varepsilon/(L + 1)$, and*
2. *the PCS parameter $\delta(x)$ is set to $\delta(x) = \delta/(2|X|)$.*

Proof: The validity of Lemma 2 is an immediate consequence of the proof of Lemma 1, when noticing that, for a nodal PCS value of $1 - \delta(x)$, the conditions of Lemma 1 will hold with probability $[1-\delta(x)]^{|X|}$. Hence, setting $1-\delta/2 \leq [1-\delta(x)]^{|X|}$, we obtain $\delta(x) \leq 1 - (1 - \delta/2)^{1/|X|}$. The result of Lemma 2 is implied from this last inequality, when noticing that, for $\delta \in (0, 1)$, $(1 - \delta/2)^{1/|X|} \leq 1 - (1/|X|) \cdot \delta/2$. □

2.3.2 Establishing the efficiency of the proposed algorithm

In order to establish the efficiency of the proposed algorithm, we need to show that, for any environment $\mathcal{E} = (X, \mathcal{A}, \mathcal{P}, \mathcal{R})$ and parameters $\delta \in (0, 1/2)$ and $\varepsilon > 0$, the sampling scheme defined by Observation 1, the results of Theorem 1, Observation 2 and Lemma 2 can be executed, with probability of success at least $1 - \delta^I = 1 - \delta/2$, within an episode budget, N, that is polynomially related to $1/\delta$, $1/\varepsilon$, $|X|$, $|\bar{A}|$, L, $\bar{v}$ and $1/\underline{q}$. This result is established in the rest of this section. In the subsequent discussion, $\Psi(x, a)$ denotes an observation of the total reward obtained by the learning agent when selecting action $a \in \mathcal{A}(x)$, while in state $x \in X$, and subsequently following the pre-determined policy $\hat{\pi}$ until the end of the running episode.

As a first step, we argue that focusing on the number of the budgeted episodes in order to establish the polynomial complexity of the proposed algorithm is justifiable, since the execution of each single episode is of polynomial complexity with respect to the parameters of interest. This result is formally stated in the following observation:

Observation 3: The combined computational cost experienced by the proposed algorithm with respect to (i) the action selection and (ii) the collection and processing of a single observation, $\Psi(x, a)$, during any single episode, is $O(L|\bar{A}|)$. Also, for any actively explored state x that has completed the sampling requirements of Theorem 1, the determination of the apparent optimum action $\hat{\pi}(x)$ is of complexity $O(|\bar{A}|)$. Hence, the overall computational cost experienced by the proposed algorithm during any single episode is $O((L + 1)|\bar{A}|)$. $\square$

On the other hand, the total number of observations $\Psi(x, a)$ that must be taken across all the state-action pairs, (x, a), of the environment, is $\sigma = \sum_{x \in X} |\mathcal{A}(x)| \cdot n(x)$, where $n(x)$ is obtained from Equation 11, by substituting: (i) k with $|\mathcal{A}(x)|$; (ii) $\bar{v}$ with $(L - l + 1)\bar{v}$, where l is the level of node x; (iii) ε with $\varepsilon(x) = \varepsilon/(L + 1)$; and (iv) δ with $\delta(x) = \delta^{II}/|X| = \delta/(2|X|)$. Hence, we have:

Observation 4: The total number of observations, $\Psi(x, a)$, that must be taken across all the state-action pairs, (x, a), of the environment, is

$$\sigma = O(\frac{|X||\bar{A}|(L+1)^4\bar{v}^2}{\varepsilon^2}\ln(\frac{|X||\bar{A}|}{\delta})) \tag{35}$$

i.e., σ is a polynomial function of $1/\varepsilon$, $1/\delta$ and the parameters $|\bar{A}|$, $|X|$, L and $\bar{v}$, that characterize the problem "size". □

However, the stochastic nature of the state transitions taking place in the considered environment, implies that the number of episodes $n(\sigma)$ required to collect these σ observations will be, in general, larger than σ; in fact, $n(\sigma)$ can be infinitely large, in the worst case. A systematic characterization of the statistics of $n(\sigma)$ can be facilitated by the following observation.

Observation 5: Under the assumption that, at every unexplored state x, the proposed algorithm selects the exercised action $a \in \mathcal{A}(x)$ in a way that maintains a positive probability for accessing the set of actively explored states, the materialization of an observation $\Psi(x, a)$ at any single episode constitutes a *Bernoulli trial* [18], with its probability of success bounded from below by $\underline{q}$. □

Next, we show that the combination of Observations 4 and 5 enables the determination of an episode budget, N, that is polynomially related to the problem parameters and will suffice for the collection of the σ requested observations, with probability at least $1 - \delta^I = 1 - \delta/2$. For this, we make the very conservative but simplifying assumption that the requested observations, $\Psi(x, a)$, are pursued one at a time; i.e., at every single episode, the algorithm focuses on a particular observation $\Psi(x, a)$ that it tries to achieve, and it ignores any other potentially available observations. Focusing on this particular algorithmic implementation enables the determination of the required budget, N, according to the decomposing scheme described below in Observation 6. Furthermore, the derived result remains applicable to the more practical algorithmic implementations where actively explored states are sampled in parallel, since these more realistic operational schemes do increase the probability of reaching an actively explored state at any single episode.

Observation 6: An episode budget, N, that is adequate for the collection of the σ observations of Equation 35, with probability at least $1 - \delta^I = 1 - \delta/2$, and is polynomially related to the problem parameters $1/\delta$, $1/\varepsilon$, $|X|$, $|\bar{A}|$, L, $\bar{v}$, $1/\underline{q}$, can be obtained by:

a. first determining an episode budget, $N(x, a)$, that (i) will enable the considered algorithm to obtain a single observation $\Psi(x, a)$ with probability of success $1 - \delta^I/\sigma = 1 - \delta/(2\sigma)$ and (ii) it is polynomially related to σ and the aforementioned parameters of interest, and

b. subsequently setting $N = \sigma \cdot N(x, a)$.

□

The result of Observation 6 is an immediate consequence of the application of the Bonferroni inequality to the particular algorithmic implementation described above. The next lemma determines an episode budget $N(x, a)$ that satisfies the requirements of Observation 6.

Lemma 3 *An episode budget, $N(x, a)$, that guarantees the collection of a single observation, $\Psi(x, a)$, with probability at least $1 - \delta/(2\sigma)$, is*

$$N(x, a) = \lceil \frac{1}{\underline{q}} \ln(\frac{2\sigma}{\delta}) \rceil \tag{36}$$

Proof: It is well known from basic probability theory [18], that the number of failures, y, before the first success, in a sequence of independent Bernoulli trials with success probability q, follows a geometric distribution with cdf

$$F(y) = \begin{cases} 1 - (1 - q)^{\lfloor y \rfloor + 1} & \text{if } y \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{37}$$

In the context of Lemma 3, we are essentially requesting that the number of failures experienced in the involved sequence of Bernoulli trials, does not exceed $N(x,a) - 1$ with probability at least $1 - \delta/(2\sigma)$. Hence, according to Equation 37, the allocated budget, $N(x,a)$, must satisfy:

$$1 - (1 - \underline{q})^{N(x,a)} \geq 1 - \delta/(2\sigma) \iff \tag{38}$$

$$(1 - \underline{q})^{N(x,a)} \leq \delta/(2\sigma) \tag{39}$$

Notice that, since Equation 38 applies for any state-action pair (x,a), we have used the minimal success probability $\underline{q}$. From the well-known inequality $1 - y \leq e^{-y}$, it follows that Equation 39 can be satisfied by picking $N(x,a)$ such that:

$$e^{-N(x,a)\underline{q}} \leq \delta/(2\sigma) \tag{40}$$

Solving Equation 40 for $N(x,a)$, we obtain:

$$N(x,a) \geq \frac{1}{\underline{q}} \ln(\frac{2\sigma}{\delta}) \tag{41}$$

which proves the validity of the lemma. □

The above discussion can be recapitulated as follows: Observation 3 establishes that the execution of a single episode under the proposed algorithm has a computational cost that is a polynomial function of the problem-defining parameters, $|X|$, $|\bar{A}|$, L, $\bar{v}$ and $1/\underline{q}$. Furthermore, Observation 4 establishes that the total number of observations, σ, that must be collected across all state-action pairs (x,a), in order to guarantee the PAC performance of the proposed algorithm, is polynomially related to the parameters $1/\varepsilon$, $1/\delta$, $|X|$, $|\bar{A}|$, L, $\bar{v}$ and $1/\underline{q}$. Observation 5 establishes that the acquisition of a single observation, $\Psi(x,a)$, can be perceived as a Bernoulli trial with its success probability bounded from below by $\underline{q}$. Lemma 3 exploits this result in order to determine an episode budget, $N(x,a)$, that enables the acquisition of a single observation, $\Psi(x,a)$, with probability at least $1 - \delta/(2\sigma)$ and is polynomially related to the parameters $1/\varepsilon$, $1/\delta$, $|X|$, $|\bar{A}|$, L, $\bar{v}$ and $1/\underline{q}$. Finally, Observation 6 establishes that the episode budget $N = \sigma \cdot N(x, a)$ is adequate for collecting the requested σ observations with probability $1 - \delta^I = 1 - \delta/2$. Since the quantity $\sigma \cdot N(x, a)$ remains a polynomial function of $1/\varepsilon$, $1/\delta$, $|X|$, $|\bar{A}|$, L, $\bar{v}$ and $1/\underline{q}$, the proposed algorithm is *efficient*. A detailed description of this algorithm, according to its basic characterization in Observation 1 and its further parametrization through the results of Sections 2.3.1 and 2.3.2, is provided in Figures 3 and 4. Furthermore, the next theorem is a direct consequence of all the previous developments.

Theorem 2 *The algorithm of Figures 3 and 4 is an efficient PAC algorithm for the RL problem considered in this work.*

2.4 Discussion

In this chapter we investigated the RL problem for the case that the underlying target task evolves in an "episodic" manner over a state space with a well-defined initial state and acyclic structure. The departing point of our analysis was the observation that the acyclic structure of the underlying state space implies a certain information flow for the overall learning process, which admits a natural and effective translation to an exploration strategy. Subsequently, the main body of our results combined this exploration strategy with further relevant results from the area of statistical inference, in order to design a RL algorithm customized to the special structure of the considered problem. The derived algorithm is computationally simple, and therefore, easily implementable in "real-world" applications. It was also shown to be efficient, according to the formal characterizations of efficiency provided by computational learning theory. By taking advantage of the admittedly simpler structure of the RL problem considered in this work, the proposed algorithm presents significantly lower computational complexity than the efficient PAC learning algorithm proposed in [25, 26] for more general RL problems, and it also compares favorably, in terms of computational complexity and implementational feasibility, with the algorithm developed in

[19], which also addresses episodic RL problems evolving over acyclic state spaces. Finally, an additional contribution of the presented work was the extension of the applied statistical theory itself, through Theorem 1, which establishes a new criterion for Ranking & Selection that is applicable to populations with general distributions.

We should notice, at this point, that the developed algorithm is immediately extensible to the more complex version of the problem where, at every episode, a task is splitting to a number of subtasks that execute in parallel and contribute to the total reward collected by the learning agent. This is actually the case with the ODP problem described in the introductory chapter, where a single episode involves the concurrent processing of all the items extracted at the different stages of the disassembly process. The only necessary modification for accommodating this additional problem element concerns the appropriate evaluation of the quantity $\bar{v}(x)$ in a way that it applies to the notion of observations $\Psi(x, a)$ experienced by the learning agent in this new operational context; the resolution of this issue is straightforward and the relevant technical details are omitted.

The proposed methodology is also extensible to the case of RL problems where the immediate rewards can be drawn from unbounded distributions, which, however, possess bounded mean and variance. Under the assumption that the learning agent possesses a uniform upper bound, $\bar{v}$, for the variances of the aforementioned distributions, and using the Chebychev instead of the Hoeffding inequality in the relevant derivation, one can establish a R&S criterion similar to that of Theorem 1, with the new sample size being equal to $n = \lceil \frac{2(k-1)\bar{v}}{\varepsilon^2 \delta} \rceil$, and with the parameters k, ε and δ having the same interpretation with that stated in Theorem 1.

A third important aspect of the results developed herein is that they are directly applicable to *partially observable (PO-)* MDP's. This capability stems from the direct, on-line nature of the proposed algorithm, which enables it to forego the explicit characterization of the internal system dynamics, and to work exclusively on the space induced by the measurement / observation sequences. This situation is exemplified by the state space definition of the ODP problem discussed in the introductory chapter, and it is reminiscent of the class of *Augmented MDP (AMDP)* algorithms in the emerging PO-MDP literature (cf. [41], Chpt. 16).

When viewed from a more practical implementational standpoint, the PAC nature of the proposed algorithm implies that it should be perceived as the *"Phase I"* of a broader learning process, during which the agent tries to learn a (near-)optimal policy fast and with very high probability. Once the execution of this algorithm has been completed, the agent will switch to *"Phase II"*, where a more standard – e.g., the Q-learning – RL algorithm will be employed, in a way that incorporates and exploits the information obtained in Phase I. Notice that maintaining some active exploration in Phase II is important, since (i) this exploration can counter-balance the potential of error tolerated, through the error probability δ, in Phase I, and (ii) it enables the reaction of the learning agent to any non-stationarity of the environmental parameters. On the other hand, the above interpretation of the presented algorithm as a "Phase I" computation in a broader learning process naturally raises the question of how much effort should be expended on it. Clearly, this effort depends on the "tightness" of the PAC requirement, as expressed by the values of the parameters ε and δ, and while the resolution of this issue will be context-specific, in general, some relevant observations are in order. First of all, it should be clear from the above developments that the proposed algorithm is more *"exploration"* than *"exploitation"*-oriented. More specifically, during the algorithm execution, the primary concern underlying the applied action selection policy is the coverage of the necessary sampling requirements that will lead to the identification of a target ε-optimal policy, rather than the maximization of the value accumulated during that period. Such a strategy is consistent with the implicit *stationarity* assumption underlying the problem statement, since in that case, the earlier an optimized strategy is identified, the higher the long-run profitability of this strategy will be. On the other hand, things are different in a non-stationary operational context. In that case, expending an extensive effort to find an optimized policy for the prevailing conditions might be futile, since this policy will be rendered irrelevant by the future evolution of the system dynamics. In fact, for highly non-stationary environments, the execution of the considered algorithm might not be even feasible, since the system sojourn in any particular parametric regime might not be long enough in order to perform the necessary sampling. Hence, in the case of non-stationary operational environments, one should compromise for a rapidly obtainable policy with a decent performance and maintain a high level of exploration in the algorithm implementing the "Phase II" computation.[4] From a more technical standpoint, the selection of a pertinent value for the performance parameter ε, that characterizes the suboptimality of the derived policy, should be relativized with respect to the magnitude of the expected immediate rewards. In fact, this relativistic interpretation of the value of ε is applied automatically by the algorithm of Figures 3 and 4, since in the calculation of $n(x)$ – the only place where the parameter ε is actually involved during the algorithm implementation – the factor $4\bar{v}(x)^2/\varepsilon(x)^2$ can be rewritten as $1/(\varepsilon(x)/(2\bar{v}(x)))^2$. At the same time, the formulae determining the episode budget, N, in Figures 3 and 4, also reveal that this quantity is affected by the second performance parameter, δ, only through the value of $\ln(1/\delta)$. Hence, one can afford to be more demanding regarding the *success* of the computation performed in Phase I rather than the *quality* of the result of this computation. This last remark corroborates the above suggestion that, in many applications, a pertinent implementation of the developed algorithm should seek to compute with high success an initial near-optimal policy, rather than expend a very large amount of effort for getting (very close) to the optimal policy.

[4] More generally, there is an obvious trade-off between the "tightness" expressed by the parameters ε and δ, and the level of exploration that must be maintained in Phase II.

Finally, it is clear from all the previous discussion that the main focus of the above developments was the establishment of the PAC nature of the proposed algorithm and its efficiency, where this last concept was interpreted according to the definitions provided by computational learning theory. In particular, all the presented developments sought to explicitly establish the ability of the proposed algorithm to guarantee the PAC requirement, within a number of episodes that is polynomially related to the parameters of interest, rather than provide the tightest possible bound for such an episode budget. Additional future research should seek to identify a tighter bound for the episode budget, N, that will guarantee the PAC performance of the algorithm. Such an immediate improvement can be achieved, for instance, by replacing the calculation $N = \sigma N(x,a) = \sigma \lceil (1/\underline{q}) \ln(2\sigma/\delta) \rceil$, in the algorithm of Figures 3 and 4, with another calculation that computes the required episode budget, N, by inverting the binomial distribution for any given triplet of σ, $\underline{q}$ and $1-\delta^I$. In a similar vein, one can consider the possibility of replacing the R&S criterion of Theorem 1 with other R&S criteria that will employ sampling techniques of more sequential nature, e.g., similar to those discussed in [15, 28]. Finally, two additional issues that concern the further detailing of the implementation of the algorithm outlined in Figures 3 and 4, are (i) the design of more pertinent strategies to be followed by the algorithm when trying to obtain the required samples $\Psi(x,a)$ for the different state-action pairs (x,a), and (ii) the organization of the information provided in the collected rewards in a concise set of data structures that will enable the more expedient application of the applied R&S criteria. Some of the above issues are addressed in subsequent parts of this document. In particular, the design of pertinent sampling strategies motivates the Optimal Node Visitation (ONV) problem that is defined and studied in the next four chapters of this work.

Input: L; $X^l,\ l = 0,\dots,L;\ \mathcal{A}(x), \forall x \in X;\ \bar{v};\ \underline{q};\ \varepsilon;\ \delta$ **Output (under successful completion):** $\hat{\pi}(x), \forall x \in X$

```
I. Initialization
    (a) Compute $X \equiv \bigcup_{l=0}^{L} X^l;\ \ |X|;\ \ |\mathcal{A}(x)|, \forall x \in X;$
    (b) Set
        $\bar{v}(x) := (L - l + 1)\bar{v},\ \forall l = 0,\dots,L,\ \forall x \in X^l;$
        $\varepsilon(x) := \varepsilon/(L+1),\ \forall x \in X;$
        $\delta(x) := \delta/(2|X|),\ \forall x \in X;$
        $n(x) := \lceil \frac{4\bar{v}(x)^2}{\varepsilon(x)^2} \ln(\frac{|A(x)|-1}{\delta(x)}) \rceil, \forall x \in X;$
        $\sigma := \sum_{x \in X} |\mathcal{A}(x)| n(x);$
        $N := \sigma \lceil (1/\underline{q}) \ln(2\sigma/\delta) \rceil;$
        $Q(x,a) := 0, \forall x \in X, \forall a \in \mathcal{A}(x);$
        $O(x,a) := 0, \forall x \in X, \forall a \in \mathcal{A}(x);$
        $AE := X^L;\ \ UE := \bigcup_{l=0}^{L-1} X^l;$
        $i := 1$
```

Figure 3: The proposed PAC algorithm for the RL problem considered in this chapter: Initialization

```
II. Policy Computation
    while ($AE \neq \emptyset \wedge i \leq N$) do
      (a) Initiate a new episode by placing a token at the initial state, $x^0$, and try to route this token to an actively explored state, $x \in AE$, by picking actions that maintain a positive probability to reach such a state;
      (b) If successful
            i. select an action $a \in \mathcal{A}(x)$ for which $O(x, a) < n(x)$;
           ii. obtain an observation $\Psi(x, a)$, by accumulating the total reward obtained by exercising action $a$ at state $x$, and subsequently following the pre-computed policy $\hat{\pi}$ until the termination of the current episode;
          iii. $Q(x, a) := Q(x, a) + \Psi(x, a);\; O(x, a) := O(x, a) + 1;$
           iv. If ($O(x, a) = n(x)$)
                 • $Q(x, a) := Q(x, a)/n(x);$
                 • If ($\forall a' \in \mathcal{A}(x), O(x, a') = n(x)$)
                   – $\hat{\pi}(x) := \arg\max_{a \in \mathcal{A}(x)}\{Q(x, a)\};$
                   – remove state $x$ from $AE$;
                   – Remove from $UE$ every state $x' \in UE$ for which all the immediately successor states are not in $AE \cup UE$, and add them to $AE$.
      (c) $i := i + 1;$
    endwhile

III. Exit
    If ($AE = \emptyset$) return $\hat{\pi}(x), \forall x \in X$, else report failure
```

Figure 4: The proposed PAC algorithm for the RL problem considered in this chapter: Policy computation and Exit.

CHAPTER III

EFFICIENT SCHEDULES FOR THE PROBLEM OF OPTIMAL NODE VISITATION IN ACYCLIC STOCHASTIC DIGRAPHS

The problem addressed in this chapter can be stated as follows: Given a stochastic, acyclic, connected digraph with a single source node and a control agent that repetitively traverses this graph, each time starting from the source node, we want to define a control policy that will enable this agent to visit each of the graph terminal nodes a prespecified number of times, while minimizing the expected number of the graph traversals. A practical motivation for this problem has been the work presented in Chapter 2, where a learning agent must compute on-line an optimal policy for a task that evolves episodically over a state space that is stochastic and acyclic, and it has a single source state that defines the task initial state. As established in Chapter 2, the agent can obtain an ϵ-optimal policy with probability at least $1 - \delta$, by sampling the various actions available at each state a certain number of times[1] and selecting the action that results to the highest sample mean. Furthermore, this sampling must be performed on a layer by layer basis, starting from the terminal states and proceeding towards the initial state of the underlying state space. Higher-level states that have covered all the required sampling and have their actions selected are declared "fully explored" and abandon the layer of "actively explored" states. On the other hand, lower-level states join the layer of "actively explored" states when all their immediate successors become fully explored. It is clear that, in this setting, expedient learning translates to the completion of all the required sampling in a minimum number of

[1] that depends on the graph structure and the performance parameters ϵ and δ

episodes. However, this minimization can be defined only in an expected sense, since the stochasticity of the environment implies that the agent might fail to reach any of the actively explored states during some episodes, under any policy. Another potential application context for the problem considered in this chapter is provided by various experimental setups where the subject must be studied in a number of states that are obtained from an initial state through some sequential treatment with probabilistic outcomes at the various stages. Assuming that the performed treatment has a destructive effect on the subject, one would like to obtain the required measurements while minimizing the number of subjects utilized in the experiment.

From a methodological standpoint, the aforementioned problem falls in the broader category of stochastic scheduling problems [30, 33]. As indicated in [30], most stochastic scheduling problems are notoriously hard to solve optimally, and one has to compromise for solutions that are suboptimal but computationally tractable. In particular, the last few years have seen the emergence of a number of works that seek to provide suboptimal solutions to various stochastic scheduling problems by exploiting some "relaxed" version of the original problem. Furthermore, in many cases, this line of analysis also provides guaranteed bounds for the potential suboptimality of the derived policies; c.f., for instance, the works of [6, 7] and the references provided therein.

Our results follow the spirit of these broader developments. Hence, in the first part of the chapter, we provide a formal characterization of the considered problem and we show that it abstracts to a specially structured *"stochastic shortest path" (SSP)* problem [5]. However, the solution of this SSP formulation through standard approaches based on Dynamic Programming is of non-polynomial complexity with respect to the underlying problem size, and therefore, in the rest of the chapter we develop a series of suboptimal policies that seek to trade off operational efficiency for computational tractability. Some important traits of these policies are that (a) they are *asymptotically optimal*, with the ratio of their performance to the performance of the optimal policy converging to unity as the node visitation requirements grow uniformly to infinity,[2] and (b) collectively they establish a broad range of options for the effective and systematic resolution of the aforementioned trade-off between efficiency and computational expedience and tractability. The development of these policies is based on (i) the pertinent exploitation of a continuous – or *"fluid"* – relaxation of the problem towards the characterization of an efficient randomized policy, and (ii) the ability to derive a closed-form expression for the performance of this randomized policy, which further enables (iii) the optimization of the policy parameters, and (iv) its embedding to adaptive control schemes that can lead to even more enhanced performance. Our results regarding item (i) above are similar *in spirit* to the results of [6, 7] concerning the computation of near-optimal policies for the job shop scheduling problem, but the underlying analysis is substantially different. The results regarding item (ii) are based on our ability to represent the dynamics generated by the considered randomized policy as a *Generalized Semi-Markov*[3] *Scheme (GSMS)* [20]. The results on item (iii) employ standard techniques borrowed from non-linear optimization [3], and those on item (iv) are building on notions borrowed from adaptive control and *"rollout"* algorithms [4, 5].

The rest of the chapter is organized as follows: Section 3.1 provides a formal characterization of the considered problem and its abstraction to a specially structured SSP. Section 3.2 introduces the aforementioned suboptimal policies, establishes their properties, including their asymptotic optimality, and investigates their relevant dominance. Subsequently, Section 3.3 complements the results developed in Section 3.2 through a number of computational experiments that demonstrate and validate them, but also offer additional practical insights. Finally, Section 3.4 concludes the chapter

[2] We also identify significant special structure that guarantees stronger convergence results for the proposed policies.

[3] actually*, Markovian*

and suggests directions for the extension of the presented results.

3.1 Problem description and its MDP formulation

3.1.1 A formal description of the considered problem

An instance of the problem considered in this chapter is completely defined by a quadruple $\mathcal{E} = (X, \mathcal{A}, \mathcal{P}, \mathcal{N})$, where

- X is a finite set of *nodes*, that is partitioned into a sequence of *"layers"*, X^0, X^1, ..., X^L. $X^0 = \{x^0\}$ defines the *source* or *root node*, while nodes $x \in X^L$ are the *terminal* or *leaf* nodes.

- $\mathcal{A}$ is a set function defined on X, that maps each $x \in X$ to the finite, non-empty set $\mathcal{A}(x)$, comprising all the *decisions / actions* that can be executed by the control agent at node x. It is further assumed that for $x \neq x'$, $\mathcal{A}(x) \cap \mathcal{A}(x') = \emptyset$.

- $\mathcal{P}$ is the *transition function*, defined on $\bigcup_{x \in X} \mathcal{A}(x)$, that associates with every action a in this set a discrete probability distribution $p(\cdot; a)$. The support sets, $\mathcal{S}(a)$, of the distributions $p(\cdot; a)$ are subsets of the set X that satisfy the following property: For any given action $a \in \mathcal{A}(x)$ with $x \in X^i$ for some $i = 0, \ldots, L-1$, $\mathcal{S}(a) \subseteq \bigcup_{j=i+1}^{L} X^j$; for $a \in \mathcal{A}(x)$ with $x \in X^L$, $\mathcal{S}(a) = X^0$. In words, the previous assumption implies that the control agent traverses the considered graph in an iterative manner, where each iteration is an acyclic traversal that starts from the root node and ends at a leaf node $x \in X^L$. Furthermore, it is assumed that for every $x \in X$, there exists at least one action sequence $\xi(x) = a^{(0)} a^{(1)} \ldots a^{(k(x))}$ such that (i) $a^{(0)} \in A(x^0)$, (ii) $\forall i = 1, \ldots, k(x)$, $a^{(i)} \in A(x^{(i)})$ with $p(x^{(i)}; a^{(i-1)}) > 0$, and (iii) $p(x; a^{k(x)}) > 0$; we shall refer to this action sequence as an *action path* from node x^0 to node x.

- $\mathcal{N}$ is the *visitation requirement vector*, that associates with each node $x \in X^L$ a visitation requirement $\mathcal{N}_x \in \mathbb{Z}^+ \cup \{0\}$. The *support* $||\mathcal{N}||$ of $\mathcal{N}$ is defined by

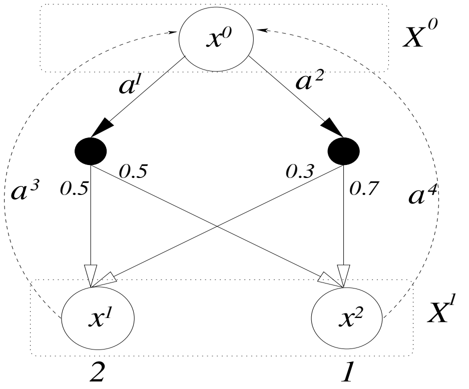

Figure 5: An example problem instance

the nodes $x \in X^L$ with $\mathcal{N}_x > 0$; we shall refer to nodes $x \in ||\mathcal{N}||$ as the problem *target* nodes.

- Finally, we define the *instance size* $|\mathcal{E}| \equiv |X| + |\bigcup_{x \in X} \mathcal{A}(x)| + |\mathcal{N}|$, where application of the operator $|\ |$ on a set returns the cardinality of this set, while application on a vector returns its l_1 norm.

In the subsequent discussion we shall also employ the variable vector $\mathcal{N}^c$ to denote the *vector of the remaining visitation requirements*. The control agent starts from the initial node x^0 at period $t = 0$, sets $\mathcal{N}^c := \mathcal{N}$, and at every consecutive period $t = 1, 2, 3, \ldots$, it (i) observes its current position, x, on the graph, and the vector of the remaining node visitation requirements, $\mathcal{N}^c$, (ii) selects an action $a \in \mathcal{A}(x)$ and commands its execution, and (iii) upon reaching one of the terminal nodes, $x \in X^L$, updates $\mathcal{N}_x{}^c$ to $(\mathcal{N}_x{}^c - 1)^+$, and subsequently, *resets* itself back to the initial node x^0, in order to start another traversal. The entire operation terminates when all the node visitation requirements have been satisfied, i.e., $\mathcal{N}^c$ has been reduced to zero. Our intention is to determine an action selection scheme – or, a *policy* – π, that maps each tuple $(x, \mathcal{N}^c)$ to an action $\pi(x, \mathcal{N}^c) \in A(x)$ in a way that minimizes the expected number of graph traversals until $\mathcal{N}^c = \mathbf{0}$.

Example 1 Figure 5 depicts a problem instance $\mathcal{E}$ where X is partitioned into layers $X^0 = \{x^0\}$ and $X^1 = \{x^1, x^2\}$. The decisions associated with each node are $\mathcal{A}(x^0) = \{\alpha^1, \alpha^2\}$, $\mathcal{A}(x^1) = \{\alpha^3\}$, and $\mathcal{A}(x^2) = \{\alpha^4\}$. The corresponding transition probabilities are $p(x^1; \alpha^1) = 0.5$, $p(x^2; \alpha^1) = 0.5$, $p(x^1; \alpha^2) = 0.3$, $p(x^2; \alpha^2) = 0.7$, and $p(x^0; \alpha^3) = p(x^0; \alpha^4) = 1$. Finally, the visitation requirement vector is defined by $\mathcal{N}_{x^1} = 2$, $\mathcal{N}_{x^2} = 1$. □

3.1.2 The induced stochastic shortest path problem

The problem defined in Section 3.1.1 can be further abstracted to a Discrete Time Markov Decision Process (DT-MDP), $\mathcal{M} = (S, A, t, c)$, where

- S is the finite set of *states*, identified with the tuples $(x, \mathcal{N}^c)$, where $x \in X$ and $\mathcal{N}^c \in \prod_{x \in X^L} \{0, \ldots, \mathcal{N}_x\}$.

- A is a set function defined on S that maps each state $s \in S$ to the finite, non-empty set $A(s)$, comprising all the *decisions / actions* that are feasible in s. More specifically, for $s = (x, \mathcal{N}^c)$, $A(s)$ coincides with $\mathcal{A}(x)$ as specified in the definition of $\mathcal{E}$.

- $t : S \times \bigcup_{s \in S} A(s) \times S \longrightarrow [0, 1]$ is the MDP *state transition* function, i.e., a *partial* function defined on all tripletes (s, a, s') with $a \in A(s)$, and with $t(s, a, s')$ being the probability to reach state s' from state s on decision a. More specifically, for $s = (x, \mathcal{N}^c)$, $a \in A(s)$, $s' = (x', \mathcal{N}^{c'})$,

$$t(s, a, s') = \begin{cases} p(x'; a), & \text{if } x \in X^l,\ l \in \{0, \ldots, L-1\},\ x' \in \bigcup_{k=l+1}^{L} X^k, \\ & \mathcal{N}^{c'} = \mathcal{N}^c; \\ 1, & \text{if } x \in X^L, x' = x^0,\ \mathcal{N}_x^{c'} = (\mathcal{N}_x^c - 1)^+,\ \mathcal{N}_y^{c'} = \mathcal{N}_y^c, \\ & \forall y \in X^L / \{x\}; \\ 0, & \text{otherwise.} \end{cases} \tag{42}$$

- $c : S \longrightarrow \{0,1\}$ is the *cost function*, where for $s = (x, \mathcal{N}^c)$,

$$c(s) = \begin{cases} 1, & \text{if } x \in X^L \text{ and } \mathcal{N}^c \neq \mathbf{0}; \\ 0, & \text{otherwise.} \end{cases} \tag{43}$$

Notice that the cost function defined by Equation 43 assigns a unit cost to every resetting transition that takes the control agent from a leaf node back to the root node, but only when there is at least one leaf node with a positive requirement. Hence, the set of states $s = (x, \mathcal{N}^c)$ with $\mathcal{N}^c = \mathbf{0}$ constitute a *closed* class which is also *cost-free*, i.e., once the process enters this class of states it will remain in it, and there will be no further cost accumulation. For the purposes of the subsequent development, we shall represent this entire class of states with a single aggregate state, s^T, which we shall refer to as the problem *terminal state*; clearly, s^T is *absorbing* and *cost-free* under any policy π. Furthermore, the MDP state set S will be redefined to $S \equiv \{(x, \mathcal{N}^c) | \mathcal{N}^c \neq \mathbf{0}\} \cup \{s^T\}$, and the action, state transition and cost functions, A, t and c, will also be appropriately redefined to reflect the above aggregation. In particular, for the terminal state s^T, we define $A(s^T) = \{a^T\}$ with $t(s^T, a^T, s^T) = 1$; $t(s^T, a^T, s) = 0$, $\forall s \in S \backslash \{s^T\}$, and $c(s^T) = 0$. The redefinition of the remaining elements of A, t and c is straightforward and the relevant details are omitted. Figure 6 exemplifies the above construct by depicting the state transition diagram for the MDP induced by the problem instance depicted in Figure 5.

In the above MDP modelling framework, we are particularly interested in a policy, π^*, that, starting from the *initial state* $s^0 \equiv (x^0, \mathcal{N})$, will drive the underlying process to the terminal state s^T with the minimum expected total cost. Formally,

$$\pi^* = \arg\min_{\pi \in \Pi} \; E_\pi[\sum_{t=0}^{\infty} c(s_t) | s_0 = s^0] \tag{44}$$

where Π denotes the entire set of policies and the expectation $E_\pi[\;]$ is taken over all possible realizations under policy π. This specification of π^* brings the considered MDP problem to a particular class of MDP problems known as *stochastic shortest*

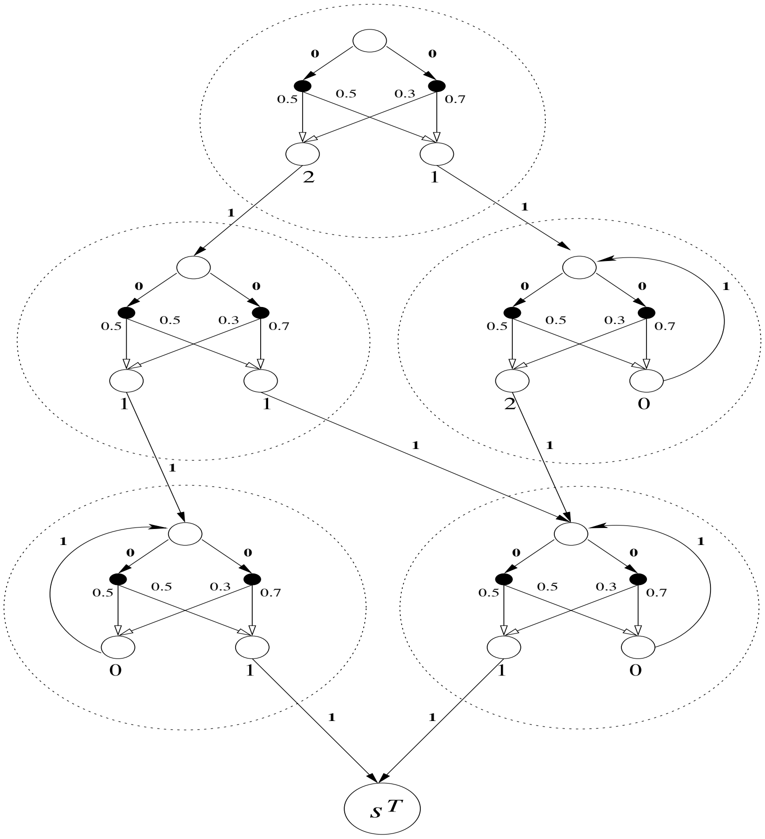

Figure 6: The State Transition Diagram for the stochastic shortest path problem induced by the problem instance depicted in Figure 5

path (SSP) problems [5]. It is easy to see that, under the assumptions stated in Section 3.1.1, this SSP problem is well-defined, and therefore, according to [5]:

Theorem 3 *For the SSP formulation characterizing the problem considered in this chapter there exists a unique vector $V^*(s)$, $s \in S$, with $V^*(s^T) = 0$ and with its remaining components satisfying the Bellman equation*

$$\begin{aligned} & \forall s \in S \backslash \{s^T\}, \\ V^*(s) \;&=\; \min_{a \in A(s)} \{c(s) + \sum_{s' \in S} t(s,a,s') \cdot V^*(s')\} \end{aligned} \tag{45}$$

Furthermore, the vector $V^*(s)$ defines an optimal policy π^* by setting

$$\begin{aligned} \forall s \in S \backslash \{s^T\}, \quad \pi^*(s) \;&:= \\ \arg \min_{a \in A(s)} \{c(s) + \sum_{s' \in S} t(s,a,s') \cdot V^*(s')\} \end{aligned} \tag{46}$$

The vector $V^*(s)$ introduced in Theorem 3 is known as the *optimal value function* or the *optimal cost-to-go vector* for the considered SSP formulation. Each component of $V^*(s)$ expresses the expected total cost of initiating the underlying process at state $s \in S$ and subsequently following an optimal policy. In particular, the expected performance for an optimal policy π^* is characterized by $V^* \equiv V^*(s^0)$. From a computational standpoint, $V^*(s)$ can be obtained through a number of approaches coming from the broader area of *Dynamic Programming (DP)* [5]. Next, we focus on an approach that employs a linear programming (LP) formulation and it will be useful in the subsequent developments presented in this document.

Theorem 4 *[5] The optimal value vector $V^*(s),\ s \in S\backslash\{s^T\}$, for the SSP formulation considered in this chapter is the optimal solution of the following linear program:*

$$\max \sum_{s \in S\backslash\{s^T\}} V(s) \tag{47}$$

s.t.

$$\forall s \in S\backslash\{s^T\},\ \forall a \in A(s),$$

$$V(s) \ \leq \ c(s) + \sum_{s' \in S\backslash\{s^T\}} t(s,a,s') \cdot V(s') \tag{48}$$

From a practical computational standpoint, the value of Theorems 3 and 4 in the determination of the optimal policy for any given problem instance, $\mathcal{E} = (X, \mathcal{A}, \mathcal{P}, \mathcal{N})$, is severely limited by the fact that the size of the state space, S, of the induced SSP problem grows exponentially to the number of the problem target nodes, $|(||\mathcal{N}||)|$, since $|S| = |X| \cdot \prod_{x \in X^L}(\mathcal{N}_x + 1) - |X| + 1$. On the other hand, the monotonic decrease of $\mathcal{N}^c$, and the acyclic structure in the underlying state space that is implied by this effect, enable the incremental solution of the formulation of Theorem 4 through a series of subproblems that are defined on the subspaces obtained by fixing the value for the remaining visitation requirement vector $\mathcal{N}^c$. Clearly, each of these subproblems will be of polynomial complexity with respect to $|\mathcal{E}|$. But the set of all possible values for $\mathcal{N}^c$ is an exponential function of $|(||\mathcal{N}||)|$, and therefore, the complexity of the overall approach remains super-polynomial. Motivated by these observations, in the next section we develop a number of suboptimal policies for the considered problem that seek to trade off some operational efficiency for computational tractability. However, all of the presented policies maintain *asymptotic* optimality, in that the ratio of their expected value over the expected value of the optimal policy converges to unity as the node visitation requirements grow uniformly to infinity. Furthermore, when viewed from a collective standpoint, the proposed policies define a broad and systematic range of options for effecting the aforementioned trade-off between performance and computational expediency and tractability.

3.2 Suboptimal control policies

3.2.1 The class of simple randomized policies

It is clear from the concluding discussion of the previous section that the main reason for the non-polynomial complexity presented by the standard DP-based approaches when applied to the considered SSP problem, is the exponentially large number of the possible values of the vector $\mathcal{N}^c$ that constitutes part of the system state $s = (x, \mathcal{N}^c)$. This observation motivates the introduction and study of a class of policies that is defined only on the basis of the first component of the system state, i.e., the position $x \in X$ of the acting agent. This idea is formalized by the concept of *simple* randomized policy as follows:

Definition 1 *Given a problem instance $\mathcal{E}$, the class of* simple randomized *policies,* Π^S*, is defined by the following two properties: (i) For any* $\pi \in \Pi^S$ *and* $s = (x, \mathcal{N}^c) \in$

S, the action $\pi(s)$ is chosen according to a probability distribution $\mathcal{D}^{\pi}(\cdot; s) = \mathcal{D}^{\pi}(\cdot; x)$, i.e., this distribution depends only upon the first component of s. (ii) For $\mathcal{N}_x > 0$, $x \in X^L$, π connects $(x^0, \mathcal{N})$ and $(x, \mathcal{N})$ with a path of positive probability.□

The satisfaction of Assumption (ii) in Definition 1 is guaranteed by the existence of action paths from node x^0 to any node $x \in X$, that was presumed in the problem statement, and the policy randomization. The next proposition establishes that simple randomized policies are characterized uniquely by the action selection probabilities that they induce for any single traversal of the underlying graph, and it provides an interesting *"flow"* interpretation for these probabilities.

Proposition 1 *There is a bijection between the space of simple randomized policies Π^S and the space $\mathcal{X}$ of vectors $\chi = \{\chi_a |\ \alpha \in \mathcal{A}(x), x \in X \backslash X^L\}$ satisfying*

$$\sum_{\alpha \in \mathcal{A}(x^0)} \chi_\alpha = 1 \tag{49}$$

$$\sum_{\alpha : x \in \mathcal{S}(\alpha)} \chi_\alpha \cdot p(x, \alpha) = \sum_{\alpha \in \mathcal{A}(x)} \chi_\alpha, \ \forall x \in X \backslash \{x^0, X^L\}, \tag{50}$$

$$\sum_{\alpha : x \in \mathcal{S}(\alpha)} \chi_\alpha \cdot p(x, a) > 0, \ \forall x \in X^L, \mathcal{N}_x > 0. \tag{51}$$

Proof: First we prove by induction that, given a simple randomized policy π, there is a unique vector χ^{π} such that $\chi^{\pi}_a,\ \alpha \in \mathcal{A}(x),\ x \in X \backslash X^L$, denotes the probability that action a will be executed during a single graph traversal under π, and this vector satisfies Constraints 49–51. Our induction runs on the number of layers, L, of the underlying acyclic graph. Hence, first consider a problem instance with $L = 1$ and assume two different simple randomized policies π and π' and the respective vectors χ^{π} and $\chi^{\pi'}$ defined by $\chi^{\pi}_a = \mathcal{D}^{\pi}(\alpha; x^0)$ and $\chi^{\pi'}_a = \mathcal{D}^{\pi'}(\alpha; x^0)$, $\alpha \in \mathcal{A}(x^0)$.[4] Since $\pi \neq \pi'$ and $L = 1$, there is an $\alpha \in \mathcal{A}(x^0)$ such that $\mathcal{D}^{\pi}(\alpha; x^0) \neq \mathcal{D}^{\pi'}(\alpha; x^0)$,

[4] We remind the reader that, according to the definitions provided in Section 3.1.1, $L = 1$ implies a two-layered graph $\mathcal{G}$, where the first layer consists of the source node x^0, and the second layer consists of the terminal nodes.

which further implies that $\chi^\pi \neq \chi^{\pi'}$. Next, assume that the hypothesis holds for all problem instances with $L \leq n$. We consider a problem instance with $L = n+1$ and two different simple randomized policies π, π'. To proceed, first consider the two policies, π, π', on the truncated acyclic graph consisting of the layers $X^0, \ldots, X^n$. According to our induction hypothesis, there exist vectors $\psi^\pi, \psi^{\pi'}$ such that for all $\alpha \in A(x), x \in X^l, 0 \leq l \leq n-1$, the components $\psi^\pi_a, \psi^{\pi'}_a$ denote the probability that action α will be executed during a single traversal of the truncated graph, under π and π' respectively. Define the vector χ^π where

$$\chi^\pi_\alpha = \begin{cases} \psi^\pi_\alpha, \text{ if } \alpha \in \mathcal{A}(x),\ x \in X^l,\ 0 \leq l \leq n-1 \\ (\sum_{\alpha' : x \in \mathcal{S}(\alpha')} \psi^\pi_{\alpha'} \cdot p(x, \alpha')) \cdot \mathcal{D}^\pi(\alpha; x), \\ \qquad\quad \text{if } a \in \mathcal{A}(x),\ x \in X^n \end{cases} \tag{52}$$

The vector $\chi^{\pi'}$ is defined accordingly. Clearly, χ^π_α,$\chi^{\pi'}_\alpha$ denote the probability that action α will be executed during a single graph traversal under π and π' respectively. Now let $l^* = \min\{l | \alpha \in \mathcal{A}(x), x \in X^l, \mathcal{D}^\pi(a;x) \neq \mathcal{D}^{\pi'}(a;x), 0 \leq l \leq n\}$. In words, l^* is the first graph layer where the two policies π, π' disagree. If $l^* \leq n-1$ then, according to the induction hypothesis, there is an $\alpha \in \mathcal{A}(x),\ x \in X^l,\ 0 \leq l \leq n-1$, such that $\psi^\pi_\alpha \neq \psi^{\pi'}_\alpha$, which together with Equation 52 imply that $\chi^\pi \neq \chi^{\pi'}$. On the other hand, if $l^* = n$, there is an $\alpha \in \mathcal{A}(x),\ x \in X^L$, such that $\mathcal{D}^\pi(\alpha; x) \neq \mathcal{D}^{\pi'}(\alpha; x)$, whereas $\psi^\pi_a = \psi^{\pi'}_a$ for all $\alpha \in \mathcal{A}(x),\ x \in X^l,\ 0 \leq l \leq n-1$, which when combined with Equation 52, imply again that $\chi^\pi \neq \chi^{\pi'}$. Hence for every simple randomized policy π, there is a unique vector χ^π such that $\chi^\pi_\alpha,\ \alpha \in \mathcal{A}(x),\ x \in X \backslash X^L$, denotes the probability that action α will be executed during a single traversal of graph $\mathcal{G}$ under π. Clearly, χ^π should satisfy the balance conditions expressed by Equations 49-50. Furthermore, part (ii) of Definition 1 implies that every target leaf node, $x \in X^L$, of the underlying graph $\mathcal{G}$, is reachable under π, and therefore, Equation 51 is also satisfied by χ^π. Hence, $\chi^\pi \in \mathcal{X}$ and $\pi \to \chi^\pi$ is injective.

On the other hand, given a vector $\chi \in \mathcal{X}$, we define the simple randomized policy π that assigns to a state $s = (x, \mathcal{N}^c)$, with $x \in X \backslash X^L$ and $\sum_{\alpha \in \mathcal{A}(x)} \chi_a > 0$, an action $\pi(x, \mathcal{N}^c) \in A(s)$ according to the probability distribution

$$\mathcal{D}^{\pi}(\alpha; x) = \frac{\chi_a}{\sum_{a \in \mathcal{A}(x)} \chi_a}, \quad \alpha \in \mathcal{A}(x). \tag{53}$$

Furthermore, for states $s = (x, \mathcal{N}^c)$ with $x \in X \backslash X^L$ and $\sum_{\alpha \in \mathcal{A}(x)} \chi_\alpha = 0$, the policy is indeterminate. Finally, for states $s = (x, \mathcal{N}^c)$, $x \in X \backslash X^L$, the policy executes the unique transition $\alpha \in A(s)$ with probability 1. Then, it can be shown, with a simple induction on the number of layers of graph $\mathcal{G}$, that χ_α, $\alpha \in \mathcal{A}(x)$, $x \in X \backslash X^L$, denotes the probability that action α will be executed during a single graph traversal under π. Hence, for every terminal node $x \in X^L$, the underlying process guided by the randomized policy π, reaches x with probability

$$\rho_x = \sum_{\alpha : x \in \mathcal{S}(\alpha)} \chi_\alpha \cdot p(x, a). \tag{54}$$

When combined with Equation 51, this last equality implies that $\rho_x > 0$, for $\mathcal{N}_x > 0$, and establishes that π belongs to the class of simple randomized policies. Thus, the mapping $\pi \rightarrow \chi^{\pi}$ is also surjective. □

As established in the previous proof, the variables χ_α, $\alpha \in \mathcal{A}(x)$, $x \in X \backslash X^L$, denote the probability of executing action α during any single traversal of the graph under policy π. Equations 49-51 also imply that the vector χ can be interpreted as the *"flow"* pattern that would result in the considered graph if a unit flow was induced in the source node x^0 and subsequently it was distributed at the different nodes $x \in X \backslash X^L$ according to the proportions suggested by the distributions $\mathcal{D}^{\pi}(\cdot, x)$.

Given a simple randomized policy π and the corresponding vector $\chi^{\pi} \in \mathcal{X}$, we also define the vector $\rho^{\pi} \equiv \rho(\chi^{\pi})$, of dimensionality $|X^L|$, with

$$\rho_x^{\pi} \equiv \sum_{\alpha : x \in \mathcal{S}(\alpha)} \chi_\alpha^{\pi} \cdot p(x, a), \; x \in X^L \tag{55}$$

Clearly, ρ_x^π, $x \in X^L$, expresses the probability of reaching node $x \in X^L$ during a single traversal of the underlying graph $\mathcal{G}$, under π. The following theorem gives an explicit characterization of the connection between the vector ρ^π and the performance of a policy $\pi \in \Pi^S$.

Theorem 5 *Consider a problem instance $\mathcal{E} = (X, \mathcal{A}, \mathcal{P}, \mathcal{N})$, a simple randomized policy $\pi \in \Pi^S$ for it, and the corresponding probability vector $\rho^\pi = \rho(\chi^\pi)$. Then,*

$$V^\pi = V(\rho^\pi) = E[\max_{j:\mathcal{N}_j>0}\{\frac{1}{\rho_j^\pi}\sum_{i=1}^{\mathcal{N}_j}\Xi_j^i\}] \tag{56}$$

where Ξ_j^i are independent identically distributed exponential random variables with rate $\lambda = 1$.

Proof: Consider a continuous-time version of the problem where the process is guided by the simple randomized policy π and a graph traversal is concluded at random times Y_i generated by a Poisson process with rate $\lambda = 1$. Let T_j denote the time until target leaf node j has satisfied its visitation requirements, and N denote the total number of graph traversals required until every visitation requirement is satisfied. Then it is easy to see that (i) T_j is distributed according to a Gamma distribution with parameters $\mathcal{N}_j$ and ρ_j^π, and (ii) the T_j's are independent. Let $T = \max_{j:\mathcal{N}_j>0}\{T_j\}$. Then

$$\begin{aligned} E[\max_{j:\mathcal{N}_j>0}\{T_j\}] &= E[T] \\ &= E[\sum_{i=1}^{N} Y_i] \\ &= E[E[\sum_{i=1}^{N} Y_i|N]] \\ &= E[N \cdot E[Y_1]] \\ &= E[N] \\ &= V^\pi \end{aligned} \tag{57}$$

Since T_j is equal in distribution to $\frac{1}{\rho_j^\pi}\sum_{i=1}^{\mathcal{N}_j}\Xi_j^i$, we have that

$$E[\max_{j:\mathcal{N}_j>0}\{T_j\}] = E[\max_{j:\mathcal{N}_j>0}\{\frac{1}{\rho_j^\pi}\sum_{i=1}^{\mathcal{N}_j}\Xi_j^i\}]. \tag{58}$$

The result now follows by combining Equations 57 and 58. □

An immediate implication of Theorem 5 is that the performance, V^π, of a simple randomized policy π, can be evaluated through the numerical integration of a continuous function since, for $\rho^\pi = \rho(\chi^\pi)$,

$$\begin{aligned}
V^\pi &= V(\rho^\pi) \\
&= E[\max_{j:\mathcal{N}_j>0}\{\frac{1}{\rho_j^\pi}\sum_{i=1}^{\mathcal{N}_j}\Xi_j^i\}] \\
&= \int_0^\infty P(\max_{j:\mathcal{N}_j>0}\{\frac{1}{\rho_j^\pi}\sum_{i=1}^{\mathcal{N}_j}\Xi_j^i\} > t)dt \\
&= \int_0^\infty (1 - \prod_{j:\mathcal{N}_j>0} P(\frac{1}{\rho_j^\pi}\sum_{i=1}^{\mathcal{N}_j}\Xi_j^i \le t))dt \\
&= \int_0^\infty (1 - \prod_{j:\mathcal{N}_j>0} F_{\mathcal{N}_j}(\rho_j^\pi \cdot t))dt \qquad (59)
\end{aligned}$$

where $F_{\mathcal{N}_j}(t)$ is the cumulative distribution function of the Gamma$(\mathcal{N}_j, 1)$ distribution. Another consequence of Equation 56 is the convexity of the function $V(\rho^\pi)$ with respect to ρ^π. This last property subsequently enables the effective and efficient solution of the optimization problem

$$\min_{\pi\in\Pi^S} V^\pi \tag{60}$$

which, under Theorem 5 and Proposition 1, can be alternatively stated as

$$\min V(\rho) \tag{61}$$

$$\text{s.t.} \quad \rho = \rho(\chi), \quad \chi \in \mathcal{X}.$$

Indeed, the objective function of Formulation 61, $V(\rho)$, is convex in ρ and continuously differentiable. Furthermore, the convexity of space $\mathcal{X}$, as delineated by Equations 49-51, when combined with the linearity of $\rho(\chi)$ with respect to χ, as revealed by Equation 55, imply that the space $\{\rho \mid \rho = \rho(\chi), \chi \in \mathcal{X}\}$ is also convex. Hence, the optimization problem defined by Equation 61 possesses a convex smooth structure and therefore it can be effectively addressed by standard solution techniques coming from the area of non-linear programming; we refer to [3] for the relevant details. In the following, we shall denote an optimal solution for the formulation of Equation 61 by χ^{opt}, and the corresponding simple randomized policy by π^{opt}.

3.2.2 Asymptotically optimal simple randomized policies

In this section we establish that the simple randomized policy π^{opt}, introduced in the previous section, is *asymptotically optimal*, with the ratio of its expected performance to V^* converging to unity, as the node visitation requirement vector, $\mathcal{N}$, grows uniformly to infinity. However, in order to establish this result, we need to introduce and analyze the performance of another simple randomized policy that is obtained through a continuous – or *"fluid"* – relaxation of the original MDP problem. We shall refer to this policy as π^{rel}, and as it will be revealed in the following, π^{rel} has its own merit as a suboptimal policy for the considered problem, since (i) it demonstrates the same asymptotically optimal performance with π^{opt}, but (ii) it is computationally simpler to derive than the latter, and in addition, (iii) as it will be shown in the following, it provides the basis for one of the most efficient suboptimal policies for this problem. The definition of π^{rel} relies on the optimal solution of the following LP formulation, that will be called the *"relaxing LP"*:

$$\min \sum_{a \in \mathcal{A}(x^0)} \chi_a \tag{62}$$

s.t.

$$\sum_{a:x\in\mathcal{S}(a)} p(x;a)\cdot\chi_a = \sum_{a\in\mathcal{A}(x)}\chi_a,\ \forall x\in X\backslash(\{x^0\}\cup X^L) \tag{63}$$

$$\sum_{a:x\in\mathcal{S}(a)} p(x;a)\cdot\chi_a \geq \mathcal{N}_x,\ \forall x\in X^L \tag{64}$$

$$\chi_a \geq 0,\ \forall a\in \bigcup_{x\in X\backslash X^L}\mathcal{A}(x) \tag{65}$$

In the light of the flow-based interpretation of Equations 49–51, a natural interpretation of an optimal solution of the relaxing LP, χ^*, is that it constitutes a flow pattern that can satisfy the flow requirements for the terminal nodes $x\in X^L$ expressed by the visitation requirement vector, $\mathcal{N}$, while minimizing the total amount of flow induced into the graph. Policy π^{rel} is the simple randomized policy induced by χ^* according to Proposition 1.[5] More specifically, given an optimal solution $\{\chi^*_a \mid a\in\bigcup_{x\in X\backslash X^L}\mathcal{A}(x)\}$ of the LP defined by Equations 103-106, policy π^{rel} assigns to a state $s=(x,\mathcal{N}^c)$ with $x\in X\backslash X^L$ and $\sum_{a\in\mathcal{A}(x)}\chi^*_a>0$, an action $\pi(x,\mathcal{N}^c)\in A(s)$ according to the probability distribution

$$\text{Prob}(\pi^{rel}(x,\mathcal{N}^c)=a)=\frac{\chi^*_a}{\sum_{a\in\mathcal{A}(x)}\chi^*_a},\quad a\in\mathcal{A}(x). \tag{66}$$

On the other hand, states $s=(x,\mathcal{N}^c)$ with $x\in X\backslash X^L$ and $\sum_{a\in\mathcal{A}(x)}\chi^*_a=0$, are inaccessible under π^{rel}, and the policy is indeterminate at them. Finally, for states $s=(x,\mathcal{N}^c)$, $x\in X^L$, the policy executes the unique transition $a\in A(s)$ with probability one. Clearly, the deployment and execution of the aforestated policy π^{rel} is of polynomial complexity with respect to the problem size $|\mathcal{E}|$. Furthermore, another consequence of the above characterizations of the relaxing LP and the policy π^{rel}, is the following theorem:

Theorem 6 *Given a problem instance $\mathcal{E}=(X,\mathcal{A},\mathcal{P},\mathcal{N})$, let $V^*_{rel}(\mathcal{N})$ denote the optimal value of the relaxing LP, χ^* denote an optimal solution of it, and $\rho^{rel}=\rho(\chi^*)$.*

[5] Notice that a single problem instance, $\mathcal{E}$, can have more than one instantiations of π^{rel} since, in general, there will be more than one optimal solutions, χ^*, for the corresponding relaxing LP.

Then,

$$V^*_{rel}(\mathcal{N}) = \max_{j:\mathcal{N}_j>0}\{\frac{\mathcal{N}_j}{\rho^{rel}_j}\} \leq V^*. \tag{67}$$

Proof: The validity of the equality part in Equation 67 is immediately obvious when realizing that $\rho^{rel}_j,\ j \in X^L$, denotes the amount of flow routed to node j by the flow pattern corresponding to policy π^{rel}, for every unit of flow induced in the underlying graph (cf. the discussion after Prop. 1).

In order to prove the inequality of Equation 67, first notice that V^* can also be computed by a variation of the LP formulation of Equations 47–48 where the original objective function has been substituted by $\max V(s^0)$; this substitution is legitimate since it is well-known in the relevant MDP theory that the SSP optimal value function $V^*(s),\ s \in S$, is the componentwise maximal vector that satisfies the constraint of Equation 48. Then, taking the *dual* of this new LP formulation, it suffices to show that (i) every feasible solution for this dual problem induces a feasible solution $\chi_a,\ a \in \bigcup_{x\in X\setminus X^L}\mathcal{A}(x)$, for the relaxing LP, and (ii) the corresponding objective values are equal. The considered dual LP formulation is as follows [13]:

$$\min \sum_{s\in S\setminus\{s^T\}:x(s)\in X^L}\ \sum_{a\in A(s)} q(s,a) \tag{68}$$

s.t.

$$\begin{aligned} & & &\forall s \in S\setminus\{s^T\}, & (69)\\ \sum_{a\in A(s)} q(s,a) &=& &\mathbf{1}_{\{s=s^0\}} + \sum_{s'\in S\setminus\{s^T\}}\ \sum_{a\in A(s')} t(s',a,s)\cdot q(s',a) & \\ & & &\forall s \in S\setminus\{s^T\},\ \forall a \in A(s), & \\ q(s,a) &\geq& &0 & (70)\end{aligned}$$

Let $q(s,a),\ s \in S\setminus\{s^T\},\ a \in A(s)$, denote a feasible solution for this formulation, and define

$$\chi_a \equiv \sum_{s \in S \setminus \{s^T\} : a \in A(s)} q(s,a), \quad \forall a \in \bigcup_{x \in X \setminus X^L} \mathcal{A}(x) \tag{71}$$

In the remaining part of this proof we shall show that the vector $\{\chi_a\}$ defined by Equation 71 satisfies the aforestated requirements (i) and (ii).

Clearly, Constraint 106 is immediately satisfied by Constraint 70 and the definition of $\{\chi_a\}$. Next we prove the feasibility of $\{\chi_a\}$ with respect to Constraint 104. Hence, consider a node $x \in X \setminus (\{x^0\} \cup X^L)$. For it, we have that:

$$\begin{aligned}
\sum_{a \in \mathcal{A}(x)} \chi_a &= \sum_{a \in \mathcal{A}(x)} \sum_{s \in S \setminus \{s^T\} : a \in A(s)} q(s,a) \quad \text{(from Eq. 71)} \\
&= \sum_{s \in S \setminus \{s^T\} : x(s) = x} \sum_{a \in A(s)} q(s,a) \quad \text{(by term rearrangement)} \\
&= \sum_{s \in S \setminus \{s^T\} : x(s) = x} \sum_{s' \in S \setminus \{s^T\}} \sum_{a \in A(s')} t(s',a,s) \cdot q(s',a) \quad \text{(from Eq. 69)} \\
&= \sum_{a : x \in \mathcal{S}(a)} p(x;a) \sum_{s' \in S \setminus \{s^T\} : a \in A(s')} q(s',a) \\
&\qquad \text{(from Eq. 42 and term rearrangement)} \\
&= \sum_{a : x \in \mathcal{S}(a)} p(x;a) \cdot \chi_a \quad \text{(from Eq. 71)}
\end{aligned}$$

To prove the satisfaction of Constraint 105 by the vector $\{\chi_a\}$, first notice that this constraint is trivially satisfied for all non-target nodes $x \in X^L$. Hence, consider a node $x \in X^L$ with $\mathcal{N}_x > 0$. Then, by working as in the proof of the validity of Constraint 104, we can easily establish that

$$\sum_{a : x \in \mathcal{S}(a)} p(x;a) \cdot \chi_a = \sum_{s \in S \setminus \{s^T\} : x(s) = x} \sum_{a \in A(s)} q(s,a) \tag{72}$$

In the STD of the underlying SSP problem, consider the arc set $\mathcal{C}_x(\mathcal{N}_x)$, consisting of all the arcs that lead from any state $s \in S_x(\mathcal{N}_x) \equiv \{(x, \mathcal{N}^c) : \mathcal{N}^c_x = \mathcal{N}_x\}$ to the resultant state $s' = (x^0, \mathcal{N}^c - \mathbf{1}_x)$, where $\mathbf{1}_x$ denotes the unit vector of dimensionality $|X^L|$ and with the non-zero component corresponding to node x.[6] Clearly, since x

[6] The reader is referred to Figure 7 for a more concrete visualization of the concepts and arguments

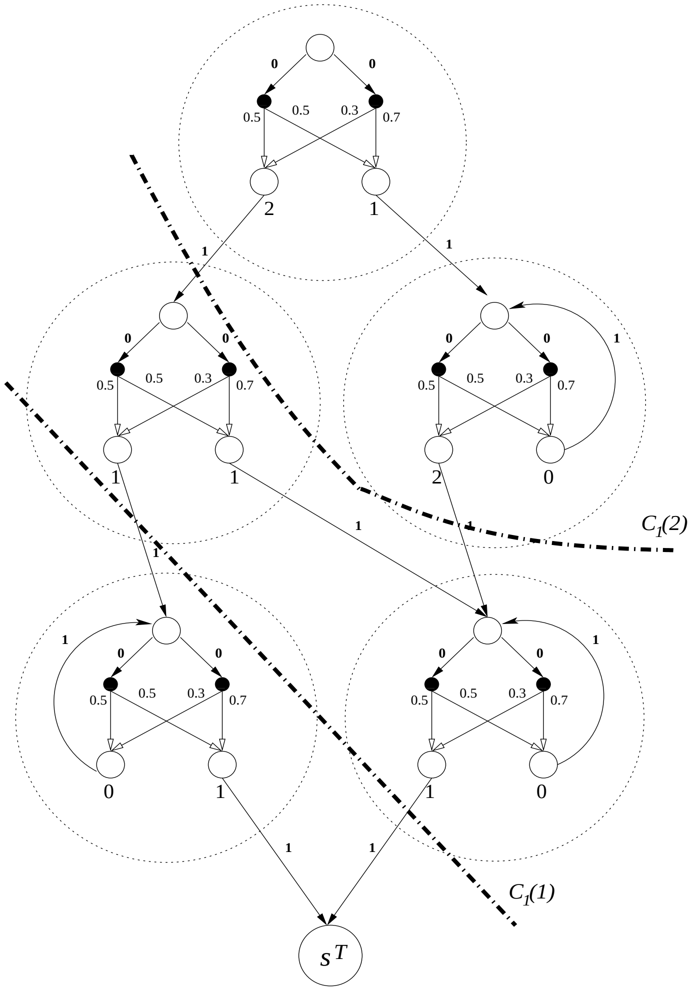

Figure 7: The STD cuts $\mathcal{C}_1(1)$ and $\mathcal{C}_1(2)$ defined by the target leaf node x^1 in the optimal node visitation problem of Figure 5.

is a target node, $\mathcal{C}_x(\mathcal{N}_x)$ is non-empty. Furthermore, since this set aggregates all the possible transitions through which the visitation requirements for x are reduced from $\mathcal{N}_x$ to $\mathcal{N}_x - 1$, it defines a *cut* on the underlying graph defined by S and $A(s),\ s \in S$. This last observation combined with the fact that $\{q(s,a)\}$ can be interpreted as a flow that conveys a unit load from state s^0 to state s^T imply that

$$\sum_{(s,a)\in\mathcal{C}_x(\mathcal{N}_x)} q(s,a) = 1 \tag{73}$$

In the same way, we can define the arc sets $\mathcal{C}_x(\mathcal{N}_x - k),\ k \in \{1,\ldots,\mathcal{N}_x - 1\}$, each

related to this part of the proof.

consisting of all the arcs that lead from any state $s \in S_x(\mathcal{N}_x - k) \equiv \{(x, \mathcal{N}^c) : \mathcal{N}^c_x = \mathcal{N}_x - k\}$ to the state $s' = (x^0, \mathcal{N}^c - \mathbf{1}_x)$, and establish that

$$\sum_{(s,a)\in\mathcal{C}_x(\mathcal{N}_x - k)} q(s,a) = 1, \quad \forall k \in \{1, \dots, \mathcal{N}_x - 1\} \tag{74}$$

But then, the satisfaction of Constraint 105 results immediately from the fact that each of the summations appearing in Equations 73 and 74 is subsumed in the double summation that appears in the right-hand-side of Equation 72.

It remains to show that

$$\sum_{a\in\mathcal{A}(x^0)} \chi_a = \sum_{s\in S\setminus\{s^T\}:x(s)\in X^L} \sum_{a\in A(s)} q(s,a)$$

The validity of this equation is established as follows:

$$\begin{aligned}
\sum_{a\in\mathcal{A}(x^0)} \chi_a &= \sum_{s\in S\setminus\{s^T\}:x(s)=x^0} \sum_{a\in A(s)} q(s,a) \quad \text{(as in the proof of Constraint 104)} \\
&= \sum_{s\in S\setminus\{s^T,s^0\}:x(s)=x^0} \sum_{a\in A(s)} q(s,a) + \sum_{a\in A(s^0)} q(s^0,a) \\
&= \sum_{s\in S\setminus\{s^T,s^0\}:x(s)=x^0} \sum_{s'\in S\setminus\{s^T\}} \sum_{a\in A(s')} t(s',a,s)\cdot q(s',a) + \\
&\quad 1 + \sum_{s'\in S\setminus\{s^T\}} \sum_{a\in A(s')} t(s',a,s^0)\cdot q(s',a) \quad \text{(from Eq. 69)} \\
&= 1 + \sum_{s\in S\setminus\{s^T\}:x(s)=x^0} \sum_{s'\in S\setminus\{s^T\}} \sum_{a\in A(s')} t(s',a,s)\cdot q(s',a) \\
&= 1 + \sum_{s\in S:x(s)=x^0} \sum_{s'\in S\setminus\{s^T\}} \sum_{a\in A(s')} t(s',a,s)\cdot q(s',a) \\
&\quad -1 \quad (\text{since} \sum_{s'\in S\setminus\{s^T\}} \sum_{a\in A(s')} t(s',a,(x^0,\mathbf{0}))\cdot q(s',a) = 1) \\
&= \sum_{s\in S\setminus\{s^T\}:x(s)\in X^L} \sum_{a\in A(s)} q(s,a) \quad \text{(from Eq. 42)}
\end{aligned}$$

□

Next, we proceed to establish the asymptotic optimality of π^{rel}. For this, consider the problem sequence, $\{\mathcal{E}(n)\}$, that is induced by a problem instance $\mathcal{E} =$

$(X, \mathcal{A}, \mathcal{P}, \mathcal{N})$, through the scaling of the visitation requirement vector, $\mathcal{N}$, by a factor $n \in \mathbb{Z}^+$. Also, let $\{V^*_{rel}(n)\}$ denote the sequence of the optimal objective values of the relaxing LP implied by the problem sequence $\{\mathcal{E}(n)\}$, and $\{V^*(n)\}$ denote the sequence of the corresponding optimal expected total costs. On the other hand, the perusal of the formulation of Equations 103–106 and of the first part of Equation 67 reveals that the policy π^{rel} remains invariant across the entire sequence $\{\mathcal{E}(n)\}$. Hence, we also define $\{V^{\pi^{rel}}(n)\}$ as the sequence of the expected costs resulting by the application of the randomized policy π^{rel} to the problem instances $\mathcal{E}(n)$. Then, we have:[7]

Theorem 7

$$V^{\pi^{rel}}(n) - V^*(n) = O(\sqrt{n}), \;\; n \in \mathbb{Z}^+. \tag{75}$$

Proof: Since $V^*_{rel}(n) \leq V^*(n), \;\; n \in \mathbb{Z}^+$, it suffices to prove that

$$V^{\pi^{rel}}(n) - V^*_{rel}(n) = O(\sqrt{n}). \tag{76}$$

Observe that

$$\begin{aligned} V^{\pi^{rel}}(n) - V^*_{rel}(n) &= E[\max_{j:N_j>0}\{\frac{1}{\rho^{rel}_j}\sum_{i=1}^{n\mathcal{N}_j}\Xi^i_j\}] - n \cdot \max_{x:\mathcal{N}_x>0}\{\frac{\mathcal{N}_x}{\rho^{rel}_x}\} \\ &\leq E[\max_{j:\mathcal{N}_j>0}\{|\frac{1}{\rho^{rel}_j}\sum_{i=1}^{nN_j}\Xi^i_j - \frac{n\mathcal{N}_j}{\rho^{rel}_j}|\}] \end{aligned} \tag{77}$$

where the above equality results from Theorems 5 and 6 and the inequality is the result of the following property:

$$\forall a_i, b_i \in \mathbb{R}, i = 1, \ldots, n,$$

$$|\max\{a_1, a_2, \ldots, a_n\} - \max\{b_1, b_2, \ldots, b_n\}| \leq \max\{|a_1 - b_1|, |a_2 - b_2|, \ldots, |a_n - b_n|\}$$

[7] We remind the reader that the notation $f(n) = O(g(n))$ implies that there exist positive constants c and n_0 such that $0 \leq f(n) \leq cg(n)$ for all $n \geq n_0$. Similarly, the notation $f(n) = \Theta(g(n))$ implies that there exist positive constants c_1, c_2, and n_0 such that $0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$ for all $n \geq n_0$. [14]

The application of the Central Limit Theorem [9] gives

$$\frac{1}{\sqrt{n}} \cdot (\frac{1}{\rho_j^{rel}} \sum_{i=1}^{n\mathcal{N}_j} \Xi_j^i - \frac{n\mathcal{N}_j}{\rho_j^{rel}}) \Rightarrow \mathbb{N}(0, \mathcal{N}_j/(\rho_j^{rel})^2),\ j : \mathcal{N}_j > 0 \tag{78}$$

where '⇒' denotes convergence in distribution as $n \to \infty$ and $\mathbb{N}(a, b)$ denotes the normal distribution with mean a and variance b.

Also, observe that

$$\frac{1}{n} \cdot E[(\frac{1}{\rho_j^{rel}} \sum_{i=1}^{n\mathcal{N}_j} \Xi_j^i - \frac{n\mathcal{N}_j}{\rho_j^{rel}})^2] = \frac{\mathcal{N}_j}{(\rho_j^{rel})^2},\ n \in \mathbb{Z}^+,\ j : \mathcal{N}_j > 0 \tag{79}$$

which implies the *uniform integrability* [9] of $\{\frac{1}{\sqrt{n}}(\frac{1}{\rho_j^{rel}} \sum_{i=1}^{n\mathcal{N}_j} \Xi_j^i - \frac{n\mathcal{N}_j}{\rho_j^{rel}})|\ n \in \mathbb{Z}^+\}$, for every j with $\mathcal{N}_j > 0$. But then, Equation 78, when combined with the independence of the Ξ_j^i's and the Continuous Mapping Theorem [9], imply that

$$\frac{1}{\sqrt{n}} E[\max_{j:\mathcal{N}_j>0}\{|\frac{1}{\rho_j^{rel}} \sum_{i=1}^{n\mathcal{N}_j} \Xi_j^i - \frac{n\mathcal{N}_j}{\rho_j^{rel}}|\}] \longrightarrow E[\max_{j:\mathcal{N}_j>0}\{|\mathbb{N}(0, \mathcal{N}_j/(\rho_j^{rel})^2)|\}] \tag{80}$$

as $n \to \infty$. Finally, Equation 307 follows by combining Equation 80 with Equation 77. □

An immediate implication of Theorem 7 is the asymptotic optimality of the policy π^{rel}:

Corollary 1

$$\frac{V^{\pi^{rel}}(n)}{V^*(n)} \to 1 \quad as \quad n \to \infty \tag{81}$$

Proof: The combination of Theorems 6 and 7 implies that $\lim_{n\to\infty} \frac{V^{\pi^{rel}}(n)}{V^*(n)} \leq 1$, while the definition of V^* implies that $V^{\pi^{rel}}(n) \geq V^*(n),\ \forall n \in \mathbb{Z}^+$. □

Theorem 7 implies also the asymptotic optimality of the policy π^{opt}, which was defined in Section 3.2.1. To obtain a formal statement of this result, let $\{V^{\pi^{opt}}(n)\}$ denote the sequence of the expected costs that results from the application on the problem sequence $\{\mathcal{E}(n)\}$ of the corresponding randomized policies $\pi^{opt}(n)$. Then, we have:

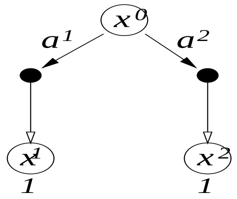

Figure 8: Example 2 – The considered problem instance

Corollary 2

$$\frac{V^{\pi^{opt}}(n)}{V^*(n)} \to 1 \quad as \quad n \to \infty \tag{82}$$

Proof: Equation 82 is an immediate consequence of Corollary 1 when noticing that the definition of $\pi^{opt}(n)$ implies that $V^*(n) \leq V^{\pi^{opt}}(n) \leq V^{\pi^{rel}}(n)$. □

Next we show that the bound implied by Equation 75 can be tight – i.e., that $V^{\pi^{rel}}(n) - V^*(n) = \Theta(\sqrt{n})$ – in certain cases, but there is also a significant problem sub-class for which the difference of Equation 75 converges to zero, as $n \to \infty$. The first of these two results is established through the following example:

Example 2 Consider the very simple problem instance depicted in Figure 8, where the root node, x^0, is immediately connected to the two leaf nodes, x^1 and x^2, through two actions, α^1 and α^2, each leading to the corresponding leaf node with probability 1. Also, assume that the visitation requirement vector is $\mathcal{N} = (1,1)$. Then, it is clear that for any scaled requirement visitation vector $n \cdot \mathcal{N} = (n,n)$, $V^*(n) = V^*_{rel}(n) = 2n$. Furthermore, the problem symmetries imply that $\pi^{opt} = \pi^{rel}$, with $\rho_i^{rel} = \rho_i^{opt} = 0.5,\ i = 1,2$. Finally, $V^{\pi^{opt}}(n) = V^{\pi^{rel}}(n) = E[\max\{\frac{1}{0.5}\sum_{i=1}^n \Xi_1^i, \frac{1}{0.5}\sum_{i=1}^n \Xi_2^i\}]$, which gives

$$\begin{aligned} V^{\pi^{opt}}(n) - V^*(n) = V^{\pi^{rel}}(n) - V^*(n) = V^{\pi^{rel}}(n) - V^*_{rel}(n) = \\ E[\max\{\frac{1}{0.5}\sum_{i=1}^n \Xi_1^i, \frac{1}{0.5}\sum_{i=1}^n \Xi_2^i\}] - 2n = \\ E[\max\{\frac{1}{0.5}\sum_{i=1}^n (\Xi_1^i - 1), \frac{1}{0.5}\sum_{i=1}^n (\Xi_2^i - 1)\}] \end{aligned} \tag{83}$$

According to an argument similar to that provided in the proof of Theorem 7,

$$\frac{1}{\sqrt{(n)}}E[\max\{\frac{1}{0.5}\sum_{i=1}^{n}(\Xi_1^i - 1), \frac{1}{0.5}\sum_{i=1}^{n}(\Xi_2^i - 1)\}] \rightarrow E[\max\{\mathbb{N}(0,4), \mathbb{N}(0,4)\}] \quad (84)$$

as $n \rightarrow \infty$. But then, the $\Theta(\sqrt{n})$ nature of the quantities involved in the different parts of Equation 83 follows immediately from the fact that $E[\max\{\mathbb{N}(0,4), \mathbb{N}(0,4)\}] > 0$. □

Notice that in the previous example, $\mathcal{N}_1/\rho_1^{rel} = \mathcal{N}_2/\rho_2^{rel}$. The equality of these ratios can be interpreted as an *equality of the "difficulty"* for the posed visitation requirements, and it turns out that it is a fundamental reason for the inability of the difference $V^{\pi^{rel}}(n) - V^*(n)$ to converge to zero, as n grows to infinity. A formal statement and a proof for this result is provided in Theorem 8 below. However, first we introduce a technical lemma that is needed in the proof of this theorem.

Lemma 4 *Let $\{\Xi^i\}$ be a sequence of independent identically distributed exponential random variables with rate $\lambda = 1$. Then, for any $\rho \in \mathbb{R}\backslash\{0\}$ and $N \in \mathbb{Z}^+$, it holds that*

$$E[exp\{\frac{1}{\rho}\sum_{i=1}^{n\cdot N}\frac{\Xi^i - 1}{\sqrt{n}}\}] \rightarrow e^{N/2\cdot\rho^2} \quad (85)$$

as $n \rightarrow \infty$.

Proof: For all $n \in \mathbb{Z}^+$ with $\sqrt{n} > \frac{1}{\rho}$, we can write

$$\begin{aligned} E[exp\{\frac{1}{\rho}\sum_{i=1}^{n\cdot N}\frac{\Xi^i - 1}{\sqrt{n}}\}] &= e^{-\frac{\sqrt{n}\cdot N}{\rho}} \cdot (E[exp\{\frac{1}{\rho}\frac{\Xi^1}{\sqrt{n}}\}])^{n\cdot N} \\ &= e^{-\frac{\sqrt{n}\cdot N}{\rho}} \cdot (\frac{1}{1 - \frac{1}{\rho\sqrt{n}}})^{n\cdot N} \\ &= e^{-\frac{\sqrt{n}\cdot N}{\rho}} \cdot (\frac{\rho\sqrt{n}}{\rho\sqrt{n} - 1})^{n\cdot N} \\ &= e^{-\frac{\sqrt{n}\cdot N}{\rho}} \cdot e^{n\cdot N\cdot\ln(\frac{\rho\sqrt{n}}{\rho\sqrt{n}-1})} \end{aligned} \quad (86)$$

But

$$\begin{aligned}
\lim_{n\to\infty} -\frac{\sqrt{n}}{\rho} + n\cdot\ln(\frac{\rho\sqrt{n}}{\rho\sqrt{n}-1}) &= \lim_{n\to\infty}\frac{-\frac{1}{\rho\sqrt{n}} + \ln(\frac{\rho\sqrt{n}}{\rho\sqrt{n}-1})}{1/n} \\
&= \lim_{n\to\infty}\frac{\frac{1}{2\rho}\cdot\frac{1}{n\sqrt{n}} - \frac{1}{2n}\cdot\frac{1}{\rho\sqrt{n}-1}}{-n^{-2}} \\
&= \lim_{n\to\infty}\frac{1}{2}\cdot\frac{\sqrt{n}}{\rho\cdot(\rho\sqrt{n}-1)} \\
&= \frac{1}{2\rho^2} \qquad (87)
\end{aligned}$$

where the second equality above is obtained through application of L' Hôpital's rule. The result now follows from Equations 86 and 87. □

Theorem 8 *Suppose that for a given problem instance $\mathcal{E} = (X, \mathcal{A}, \mathcal{P}, \mathcal{N})$, with $l \geq 2$ target leaf nodes, there exists a target leaf node x^k such that, for any other target leaf node x^j, $\frac{\mathcal{N}_k}{\rho_k^{rel}} > \frac{\mathcal{N}_j}{\rho_j^{rel}}$. Then, as $n \to \infty$,*

$$V^{\pi^{rel}}(n) - V^*_{rel}(n) \to 0 \qquad (88)$$

Proof: Without loss of generality assume that $k = 1$. Then, the left part of Equation 88 can be re-written as follows:

$$\begin{aligned}
&V^{\pi^{rel}}(n) - V^*_{rel}(n) \\
= \;& E[\max_{j:\mathcal{N}_j>0}\{\frac{1}{\rho_j^{rel}}\sum_{i=1}^{n\cdot\mathcal{N}_j}\Xi_j^i\}] - \frac{n\cdot\mathcal{N}_1}{\rho_1^{rel}} \\
= \;& E[\max\{\frac{1}{\rho_1^{rel}}\sum_{i=1}^{n\cdot\mathcal{N}_1}\Xi_1^i, \ldots, \frac{1}{\rho_l^{rel}}\sum_{i=1}^{n\cdot\mathcal{N}_l}\Xi_l^i\}] - E[\frac{1}{\rho_1^{rel}}\sum_{i=1}^{n\cdot\mathcal{N}_1}\Xi_1^i] \\
= \;& E[\max\{0, \frac{1}{\rho_2^{rel}}\sum_{i=1}^{n\cdot\mathcal{N}_2}\Xi_2^i - \frac{1}{\rho_1^{rel}}\sum_{i=1}^{n\cdot\mathcal{N}_1}\Xi_1^i, \ldots, \frac{1}{\rho_l^{rel}}\sum_{i=1}^{n\cdot\mathcal{N}_l}\Xi_l^i - \frac{1}{\rho_1^{rel}}\sum_{i=1}^{n\cdot\mathcal{N}_1}\Xi_1^i\}] \\
\leq \;& E[(\frac{1}{\rho_2^{rel}}\sum_{i=1}^{n\cdot\mathcal{N}_2}\Xi_2^i - \frac{1}{\rho_1^{rel}}\sum_{i=1}^{n\cdot\mathcal{N}_1}\Xi_1^i)^+] + \ldots + E[(\frac{1}{\rho_l^{rel}}\sum_{i=1}^{n\cdot\mathcal{N}_l}\Xi_l^i - \frac{1}{\rho_1^{rel}}\sum_{i=1}^{n\cdot\mathcal{N}_1}\Xi_1^i)^+]
\end{aligned}$$

In order to prove the result of Theorem 8, it suffices to prove that, for all $j = 2, \ldots, l$,

$$E[(\frac{1}{\rho_j^{rel}}\sum_{i=1}^{n\cdot\mathcal{N}_j}\Xi_j^i - \frac{1}{\rho_1^{rel}}\sum_{i=1}^{n\cdot\mathcal{N}_1}\Xi_1^i)^+] \to 0 \qquad (89)$$

as $n \to \infty$.

Hence, consider an arbitrary $j \in \{2, \ldots, l\}$, and let $a_j = \frac{\mathcal{N}_1}{\rho_1^{rel}} - \frac{\mathcal{N}_j}{\rho_j^{rel}} > 0$. Then, by basic probability arguments and the Markov inequality, we get:

$$\begin{aligned}
& E[(\frac{1}{\rho_j^{rel}}\sum_{i=1}^{n\cdot\mathcal{N}_j}\Xi_j^i - \frac{1}{\rho_1^{rel}}\sum_{i=1}^{n\cdot\mathcal{N}_1}\Xi_1^i)^+] \\
= \; & E[(\frac{1}{\rho_j^{rel}}\sum_{i=1}^{n\cdot\mathcal{N}_j}(\Xi_j^i - 1) - \frac{1}{\rho_1^{rel}}\sum_{i=1}^{n\cdot\mathcal{N}_1}(\Xi_1^i - 1) - n(\frac{\mathcal{N}_1}{\rho_1^{rel}} - \frac{\mathcal{N}_j}{\rho_j^{rel}}))^+] \\
= \; & \sqrt{n}\cdot E[(\frac{1}{\rho_j^{rel}}\sum_{i=1}^{n\cdot\mathcal{N}_j}\frac{\Xi_j^i - 1}{\sqrt{n}} - \frac{1}{\rho_1^{rel}}\sum_{i=1}^{n\cdot\mathcal{N}_1}\frac{\Xi_1^i - 1}{\sqrt{n}} - \sqrt{n}(\frac{\mathcal{N}_1}{\rho_1^{rel}} - \frac{\mathcal{N}_j}{\rho_j^{rel}}))^+] \\
= \; & \sqrt{n}\cdot\int_{a_j\sqrt{n}}^{\infty} P(\frac{1}{\rho_j^{rel}}\sum_{i=1}^{n\cdot\mathcal{N}_j}\frac{\Xi_j^i - 1}{\sqrt{n}} - \frac{1}{\rho_1^{rel}}\sum_{i=1}^{n\cdot\mathcal{N}_1}\frac{\Xi_1^i - 1}{\sqrt{n}} > t)dt \\
= \; & \sqrt{n}\cdot\int_{a_j\sqrt{n}}^{\infty} P(\exp\{\frac{1}{\rho_j^{rel}}\sum_{i=1}^{n\cdot\mathcal{N}_j}\frac{\Xi_j^i - 1}{\sqrt{n}} - \frac{1}{\rho_1^{rel}}\sum_{i=1}^{n\cdot\mathcal{N}_1}\frac{\Xi_1^i - 1}{\sqrt{n}}\} > \exp\{t\})dt \\
\leq \; & \sqrt{n}\cdot\int_{a_j\sqrt{n}}^{\infty} e^{-t}E[\exp\{\frac{1}{\rho_j^{rel}}\sum_{i=1}^{n\cdot\mathcal{N}_j}\frac{\Xi_j^i - 1}{\sqrt{n}} - \frac{1}{\rho_1^{rel}}\sum_{i=1}^{n\cdot\mathcal{N}_1}\frac{\Xi_1^i - 1}{\sqrt{n}}\}]dt \\
= \; & \sqrt{n}\cdot e^{-a_j\sqrt{n}}E[\exp\{\frac{1}{\rho_j^{rel}}\sum_{i=1}^{n\cdot\mathcal{N}_j}\frac{\Xi_j^i - 1}{\sqrt{n}} - \frac{1}{\rho_1^{rel}}\sum_{i=1}^{n\cdot\mathcal{N}_1}\frac{\Xi_1^i - 1}{\sqrt{n}}\}] \\
= \; & \sqrt{n}\cdot e^{-a_j\sqrt{n}}\cdot E[\exp\{\frac{1}{\rho_j^{rel}}\sum_{i=1}^{n\cdot\mathcal{N}_j}\frac{\Xi_j^i - 1}{\sqrt{n}}\}]\cdot E[\exp\{-\frac{1}{\rho_1^{rel}}\sum_{i=1}^{n\cdot\mathcal{N}_1}\frac{\Xi_1^i - 1}{\sqrt{n}}\}] \qquad (90)
\end{aligned}$$

The result of Equation 89 follows from Equation 90, when noticing that, according to Lemma 4,

$$E[\exp\{\frac{1}{\rho_j^{rel}}\sum_{i=1}^{n\cdot\mathcal{N}_j}\frac{\Xi_j^i - 1}{\sqrt{n}}\}] \to e^{\mathcal{N}_j/2\cdot(\rho_j^{rel})^2}$$

$$E[\exp\{-\frac{1}{\rho_1^{rel}}\sum_{i=1}^{n\cdot\mathcal{N}_1}\frac{\Xi_1^i - 1}{\sqrt{n}}\}] \to e^{\mathcal{N}_1/2\cdot(\rho_1^{rel})^2}$$

as $n \to \infty$, while

$$\sqrt{n}\cdot e^{-a_j\sqrt{n}} \to 0$$

□

Hence, under the condition of Theorem 8, the performance of all three policies, π^{rel}, π^{opt} and π^*, converges to the lower bound $V^*_{rel}(n)$, as the scaling factor n grows to infinity. Furthermore, Equation 90 indicates that this convergence will be quite fast, and its rate will be determined by the maximum difference $\frac{\mathcal{N}_k}{\rho^{rel}_k} - \frac{\mathcal{N}_j}{\rho^{rel}_j}$ among all the target leaf nodes x^j with $j \neq k$. An intuitive interpretation of this result is that, as this difference grows to larger values, the information contained in the optimal solution of the relaxing LP is adequate in order to strongly bias the system behavior towards the optimal policy. On the other hand, when the maximal ratio $\frac{\mathcal{N}_k}{\rho^{rel}_k}$ is attained at more than one leaf nodes, both π^{rel} and π^{opt} will treat all these nodes as "equally difficult targets". But due to the *static* nature of these policies, this impartiality can turn into a disadvantage in the later stages of the problem evolution, where the original ties have been resolved by the underlying randomness. In the next section we discuss how these problems can be alleviated, and the performnce of the considered policies can be substantially improved, through some *adaptive* implementation mechanisms that enable the applied policy to revise its action selection scheme, and the resultant probability vector ρ^{π}, according to the information provided by the remaining requirement vector $\mathcal{N}^c$.

3.2.3 Adaptive Policies

In order to derive the enhanced suboptimal policies sought in this section, it is pertinent to consider the partitioning of the state space S, of the SSP defined in Section 3.1.2, into the state subsets defined by a common remaining visitation requirement vector, $\mathcal{N}^c$. Each of these subsets defines a notion of *"macro-state"* for the underlying process, while, as it was observed at the end of Section 3.1.2, the monotonic decrease of $\mathcal{N}^c$ implies that the induced space of macro-states is traversed in an acyclic manner. More specifically, the process starts from the macro-state defined by $\mathcal{N}^c = \mathcal{N}$, and at every macro-transition, it proceeds to a macro-state where the corresponding vector $\mathcal{N}^{c'}$ is obtained from $\mathcal{N}^c$ by reducing one of its components by one unit. Next we show that this structure enables the specification of computationally efficient suboptimal policies that perform better than the policy π^{opt} defined in Section 3.2.1. In the subsequent developments, we shall use the notation $\pi(\mathcal{N})$ to denote the instantiation of the policy π on the problem instance $\mathcal{E} = (X, \mathcal{A}, \mathcal{P}, \mathcal{N})$, and this notation will extend to any other element pertaining to the considered policy π.

The first improvement to π^{opt} is easily obtained by an adaptive implementation of it, that recomputes the optimized vector χ^{opt} at every visited macro-state, by solving the corresponding optimization problem defined by Equation 61. We shall refer to the resulting policy as π^{adopt}. Next we establish that

Proposition 2 $V^{\pi^{adopt}} \leq V^{\pi^{opt}}$

Proof: We prove this result by induction on $|\mathcal{N}|$, i.e., the total number of visitation requirements. For $|\mathcal{N}| = 1$, the process will visit only one macro-state before its termination, and therefore, $V^{\pi^{adopt}} = V^{\pi^{opt}}$. Next, we assume that the inequality of Proposition 2 holds for $|\mathcal{N}| \leq n$, and we show that it will also hold for $|\mathcal{N}| = n + 1$. To obtain this result, notice that the value function of any proper policy π will satisfy the following recursion:

$$V^{\pi}(x^0, \mathcal{N}) = \frac{1}{\sum_{x \in X^L : \mathcal{N}_x > 0} \rho_x^{\pi(\mathcal{N})}} \cdot [1 + \sum_{x \in X^L : \mathcal{N}_x > 0} \rho_x^{\pi(\mathcal{N})} \cdot V^{\pi}(x^0, \mathcal{N} - \mathbf{1}_x)] \tag{91}$$

where (i) $\rho_x^{\pi(\mathcal{N})}$ denotes the probability of reaching node $x \in X^L$ in any single traversal of graph $\mathcal{G}$ under policy π, while starting from state $(x^0, \mathcal{N})$ (c.f. Equation 55), and (ii) $\mathbf{1}_x$ denotes the unit vector of dimensionality equal to $|X^L|$ and with its non-zero component corresponding to node x. Application of Equation 91 to π^{adopt} gives that

$$V^{\pi^{adopt}(\mathcal{N})}(x^0, \mathcal{N}) \tag{92}$$

$$= \frac{1}{\sum_{x \in X^L : \mathcal{N}_x > 0} \rho_x^{\pi^{adopt}(\mathcal{N})}} \cdot [1 + \sum_{x \in X^L : \mathcal{N}_x > 0} \rho_x^{\pi^{adopt}(\mathcal{N})} \cdot V^{\pi^{adopt}(\mathcal{N})}(x^0, \mathcal{N} - \mathbf{1}_x)] \tag{93}$$

However, the definition of π^{adopt} implies that $\rho_x^{\pi^{adopt}(\mathcal{N})} = \rho_x^{\pi^{opt}(\mathcal{N})}$ and $V^{\pi^{adopt}(\mathcal{N})}(x^0, \mathcal{N} - \mathbf{1}_x) = V^{\pi^{adopt}(\mathcal{N}-\mathbf{1}_x)}(x^0, \mathcal{N} - \mathbf{1}_x)$, for all $x \in X^L$. Furthermore, $V^{\pi^{adopt}(\mathcal{N}-\mathbf{1}_x)}(x^0, \mathcal{N} - \mathbf{1}_x) \leq V^{\pi^{opt}(\mathcal{N}-\mathbf{1}_x)}(x^0, \mathcal{N} - \mathbf{1}_x) \leq V^{\pi^{opt}(\mathcal{N})}(x^0, \mathcal{N} - \mathbf{1}_x)$, $\forall x \in X^L : \mathcal{N}_x > 0$, where the first inequality results from the induction hypothesis and the second from the definition of π^{opt}. But then, Equation 93 implies that

$$\begin{aligned} & V^{\pi^{adopt}(\mathcal{N})}(x^0, \mathcal{N}) && (94) \\ \leq \; & \frac{1}{\sum_{x \in X^L : \mathcal{N}_x > 0} \rho_x^{\pi^{opt}(\mathcal{N})}} \cdot [1 + \sum_{x \in X^L : \mathcal{N}_x > 0} \rho_x^{\pi^{opt}(\mathcal{N})} \cdot V^{\pi^{opt}(\mathcal{N})}(x^0, \mathcal{N} - \mathbf{1}_x)] \\ = \; & V^{\pi^{opt}(\mathcal{N})}(x^0, \mathcal{N}) && (95) \end{aligned}$$

□

When combined with Corollary 2, Proposition 2 implies also the asymptotic optimality of policy π^{adopt}, in the sense of Corollaries 1 and 2. Next we define another class of policies that can outperform π^{opt} and they constitute a customized implementation on the considered MDP problem of the "rollout" policies discussed in [4, 5]. Under this new regime, the policy to be applied at the macro-state defined by the visitation requirement vector $\mathcal{N}^c$, is the "greedy" policy determined by Equation 46 while employing the value function $V(s)$, $s \in \{(x, \mathcal{N}^c) \mid x \in \bigcup_{l=0}^{L} X^l\}$, that is obtained by restricting the LP of Theorem 4 to the considered macro-state and setting the value function of the "boundary" states $(x^0, \mathcal{N}^c - \mathbf{1}_y)$, $y = 1, \ldots, |X^L| \; : \; \mathcal{N}_y^c > 0$, equal to $V^{\pi^{opt}(\mathcal{N}^c - \mathbf{1}_y)}(x^0, \mathcal{N}^c - \mathbf{1}_y)$. The solution of these LP's and the determination of the corresponding local policies is performed every time that the process enters a new macro-state. The resulting policy is characterized as π^{roll}, and it holds that

Proposition 3 $V^{\pi^{roll}} \leq V^{\pi^{opt}}$

Proof: Similar to the case of Proposition 2, we prove this result by induction on $|\mathcal{N}|$. It is clear that for $|\mathcal{N}| = 1$, $V^{\pi^{roll}} = V^*$, and therefore, Proposition 3 is true. Next suppose that Proposition 3 holds true for $|\mathcal{N}| \leq n$. We shall show that it also holds true for $|\mathcal{N}| = n+1$. The application of Equation 91 to policy π^{roll} gives

$$V^{\pi^{roll}(\mathcal{N})}(x^0,\mathcal{N}) = \frac{1}{\sum_{x\in X^L:\mathcal{N}_x>0}\rho_x^{\pi^{roll}(\mathcal{N})}} \cdot [1 + \sum_{x\in X^L:\mathcal{N}_x>0} \rho_x^{\pi^{roll}(\mathcal{N})} \cdot V^{\pi^{roll}(\mathcal{N})}(x^0,\mathcal{N}-\mathbf{1}_x)] \tag{96}$$

The definition of the policy π^{roll} implies that $V^{\pi^{roll}(\mathcal{N})}(x^0,\mathcal{N}-\mathbf{1}_x) = V^{\pi^{roll}(\mathcal{N}-\mathbf{1}_x)}(x^0,\mathcal{N}-\mathbf{1}_x)$, which when combined with the induction hypothesis and Equation 96 imply that

$$V^{\pi^{roll}(\mathcal{N})}(x^0,\mathcal{N}) \leq \frac{1}{\sum_{x\in X^L:\mathcal{N}_x>0}\rho_x^{\pi^{roll}(\mathcal{N})}} \cdot [1 + \sum_{x\in X^L:\mathcal{N}_x>0} \rho_x^{\pi^{roll}(\mathcal{N})} \cdot V^{\pi^{opt}(\mathcal{N}-\mathbf{1}_x)}(x^0,\mathcal{N}-\mathbf{1}_x)] \tag{97}$$

From the definition of the policy π^{roll} we also have that

$$\begin{aligned} &\frac{1}{\sum_{x\in X^L:\mathcal{N}_x>0}\rho_x^{\pi^{roll}(\mathcal{N})}} \cdot [1 + \sum_{x\in X^L:\mathcal{N}_x>0} \rho_x^{\pi^{roll}(\mathcal{N})} \cdot V^{\pi^{opt}(\mathcal{N}-\mathbf{1}_x)}(x^0,\mathcal{N}-\mathbf{1}_x)] \leq \\ &\quad \frac{1}{\sum_{x\in X^L:\mathcal{N}_x>0}\rho_x^{\pi^{opt}(\mathcal{N})}} \cdot [1 + \sum_{x\in X^L:\mathcal{N}_x>0} \rho_x^{\pi^{opt}(\mathcal{N})} \cdot V^{\pi^{opt}(\mathcal{N}-\mathbf{1}_x)}(x^0,\mathcal{N}-\mathbf{1}_x)] \end{aligned} \tag{98}$$

while the definition of the policy π^{opt} further implies that

$$\begin{aligned} &\frac{1}{\sum_{x\in X^L:\mathcal{N}_x>0}\rho_x^{\pi^{opt}(\mathcal{N})}} \cdot [1 + \sum_{x\in X^L:\mathcal{N}_x>0} \rho_x^{\pi^{opt}(\mathcal{N})} \cdot V^{\pi^{opt}(\mathcal{N}-\mathbf{1}_x)}(x^0,\mathcal{N}-\mathbf{1}_x)] \leq \\ &\quad \frac{1}{\sum_{x\in X^L:\mathcal{N}_x>0}\rho_x^{\pi^{opt}(\mathcal{N})}} \cdot [1 + \sum_{x\in X^L:\mathcal{N}_x>0} \rho_x^{\pi^{opt}(\mathcal{N})} \cdot V^{\pi^{opt}(\mathcal{N})}(x^0,\mathcal{N}-\mathbf{1}_x)] = \\ &\qquad V^{\pi^{opt}(\mathcal{N})}(x^0,\mathcal{N}) \end{aligned} \tag{99}$$

But then, Proposition 3 follows immediately from Equations 97–99. □

Clearly, policy π^{roll} is also asymptotically optimal in the sense of Corollaries 1 and 2. Furthermore, by employing $V^{\pi^{adopt}(\mathcal{N}^c-\mathbf{1}_y)}(x^0,\mathcal{N}^c-\mathbf{1}_y)$ instead of $V^{\pi^{opt}}(\mathcal{N}^c-\mathbf{1}_y)(x^0,\mathcal{N}^c-\mathbf{1}_y)$ as an estimate of the value function of the "boundary" states $(x^0,\mathcal{N}^c-\mathbf{1}_y),\ y=1,\ldots,|X^L|\ :\ \mathcal{N}_y^c>0$, and denoting the resulting rollout policy as π^{adroll}, we can also establish through arguments similar to those provided above that

Proposition 4 $V^{\pi^{adroll}} \leq V^{\pi^{adopt}}$

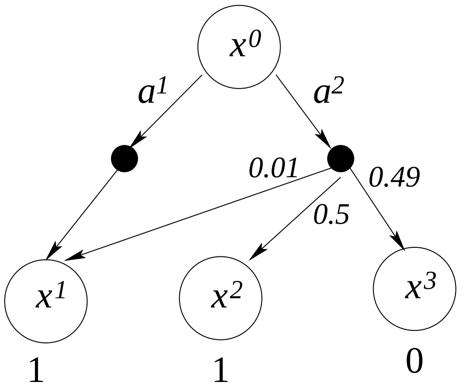

Figure 9: Example 3 – The considered problem instance

One complication regarding the implementation of policy π^{adroll} compared to those of π^{adopt} and π^{roll}, is that the estimates $V^{\pi^{adopt}(\mathcal{N}^c-\mathbf{1}_y)}(x^0,\mathcal{N}^c-\mathbf{1}_y)$ are not available in closed form. However, in most practical cases they should be easily computed through simulation. Finally, one can also envision additional versions of π^{roll} and π^{adroll} where the LP that specifies the policy to be followed at any given macro-state is formulated over an extended state subset that includes the states of the considered macro-state plus the states of all the macro-states that can be reached from it in up to k transitions. These policies are characterized as *k-step rollout* policies, and typically, they will outperform the corresponding policies resulting from single-step lookahead; we refer to [5] for some relevant discussion.

3.3 Computational Studies

In this section we present two examples that provide a concrete demonstration of the convergence results developed in Section 3.2, and also enable an assessment of the relative performance of the policies introduced in that section.

Example 3 This example pursues a detailed study of the problem instance depicted in Figure 9, in an effort to provide some additional insights on (i) the effects underlying the suboptimality of the simple randomized policies π^{rel} and π^{opt}; (ii) the role of the policy adaptation for mitigating this suboptimality, through the recalculation of the

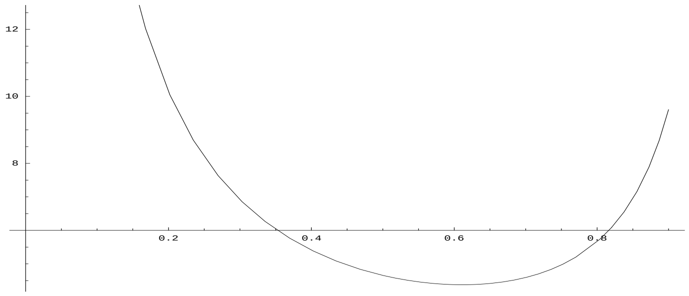

Figure 10: Example 3 – The performance of the simple randomized policies obtained for different values of the selection probability, χ, for action α^2

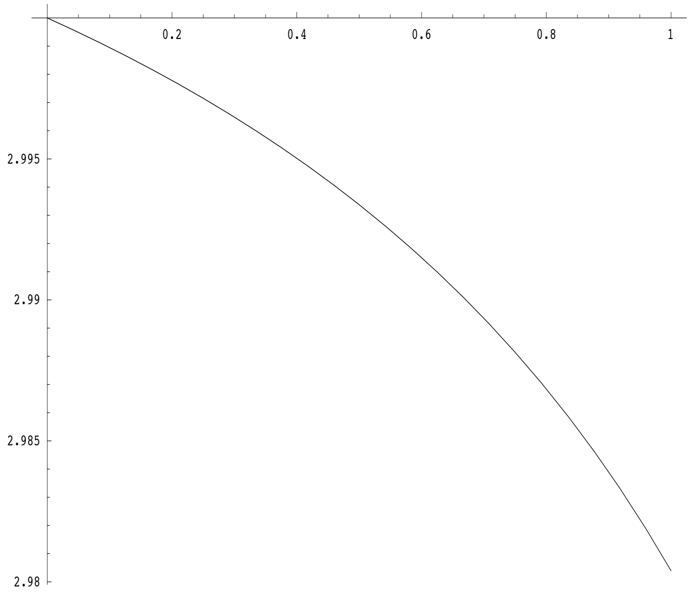

Figure 11: Example 3 – The performance of the adaptive randomized policies obtained for different values of the selection probability, χ, for action α^2 in the initial macro-state

policy-defining vectors χ^{rel} and χ^{opt} at the visited macro-states; and (iii) the relative performance of the resulting policies π^{adrel} and π^{adopt} with respect to each other and the optimal policy.[8] It should be obvious to the reader that, for the case depicted in Figure 9, the optimal policy is to choose action α^2 until the visitation requirement of node x^2 has been satisfied. At this point, if the visitation requirement of node x^1 is still unmet, the policy switches to action α^1 and satisfies this requirement in a single traversal. But any simple randomized policy, π, will fail to take advantage of the deterministic nature of action α^1, as suggested above, since it must maintain a fixed vector ρ^π at every visited macro-state. Hence, both π^{rel} and π^{opt} will apply a randomization over α^1 and α^2 that will maintain a significant positive probability for selecting action α^1, in an effort to increase the accessibility of node x^1. In particular, π^{rel} will choose action α^2 with a probability χ^{rel} that balances the ratios $\mathcal{N}_i/\rho_i^{rel}$ for $i = 1, 2$; i.e.,

$$\frac{1}{0.5\chi^{rel}} = \frac{1}{0.01\chi^{rel} + 1 - \chi^{rel}} \Longleftrightarrow \chi^{rel} = 0.671 \tag{100}$$

Furthermore, Equation 91 implies that the performance, $V^{\pi^{rel}}$, of the resulting policy, can be evaluated by plugging the obtained value for χ^{rel} into the following function:

$$V^{\pi(\chi)} = \frac{1}{0.51\chi + 1 - \chi}\left(1 + \frac{0.5\chi}{0.01\chi + 1 - \chi} + \frac{0.01\chi + 1 - \chi}{0.5\chi}\right) \tag{101}$$

Thus, it is found that $V^{\pi^{rel}} = 4.47$. On the other hand, the χ value that defines π^{opt} can be computed by solving the equation $\frac{dV^{\pi(\chi)}}{d\chi} = 0$ and picking the root that belongs in the interval $[0, 1]$. It turns out that $\chi^{opt} = 0.611291$ and $V^{\pi^{opt}} = 4.37693$.[9] Finally, Figure 10 characterizes the performance of all simple randomized policies for the considered problem instance, by plotting $V^{\pi(\chi)}$ for $\chi \in [0.1, 0.9]$.

Policies π^{adrel} and π^{adopt} present enhanced performance with respect to their static

[8] Obviously, the policy π^{adrel} is defined in a way similar to π^{adopt}, but with the policy π^{rel} being used as the base policy.

[9] It is interesting to notice the proximity of the χ^{rel} and χ^{opt} values. This seems to be a more general effect for the considered problem, with χ^{opt} and the resulting probability vector ρ^{opt} being minor *"corrections"* of χ^{rel} and ρ^{rel}. Furthermore, it can be shown that $\rho^{opt}(n) \to \rho^{rel}$ as $n \to \infty$.

counterparts, π^{rel} ad π^{opt}, since they are able to optimize their decision in the second macro-state, on the basis of the remaining requirement vector $\mathcal{N}^c$. However, they remain suboptimal since their decision in the initial macro-state is compromised by the aforementioned suboptimality of π^{rel} and π^{opt}. A closed-form evaluation of these policies can be based again on Equation 91: The performance of the adaptive randomized policy that selects action α^2 at the initial macro-state with probability χ, and in the next macro-state applies the optimal policy, is given by:

$$V^{ad-\pi(\chi)} = \frac{1}{0.51\chi + 1 - \chi}(1 + 0.5\chi + 2(0.01\chi + 1 - \chi)) \qquad (102)$$

Hence, from Equation 102 we obtain that $V^{\pi^{adrel}} = 2.99$, $V^{\pi^{adopt}} = 2.99127$ and $V^{ad-\pi(1.0)} = 2.98039$. Furthermore, Figure 11 plots the performance of all the adaptive randomized policies that are obtained by varying $\chi \in [0,1]$ and validates our original suggestion that the optimal policy is obtained for $\chi^* = 1.0$.

We conclude the discussion of this example with two additional observations: First, it is interesting to notice the proximity of $V^{\pi^{adrel}}$ and $V^{\pi^{adopt}}$ to each other and to the value of the optimal policy, $V^{ad-\pi(1.0)}$. Second, in this example it even holds that $V^{\pi^{adrel}} < V^{\pi^{adopt}}$, as manifested by the values quoted above and by the strictly decreasing nature of $V^{ad-\pi(\chi)}$. These two observations are indicative of our collective experience with the empirical performance of the aforementioned policies, and when combined with the computational simplicity of π^{adrel} compared to π^{adopt} and π^{roll}, make us believe that π^{adrel} can be the preferred policy in most practical applications. The next example provides further corroboration to this statement. □

Example 4 In this example we consider two problem instances defined by the stochastic graph of Figure 12 and the visitation requirement vectors $\mathcal{N} = (3,1,1,0,0)$ and $\mathcal{N} = (1,2,2,2,1)$. The solution of the corresponding relaxing LPs indicates that the problem instance defined by $\mathcal{N} = (3,1,1,0,0)$ satisfies the conditions of Theorem 8, with the most difficult visitation requirement determined by the leaf node x_4. On

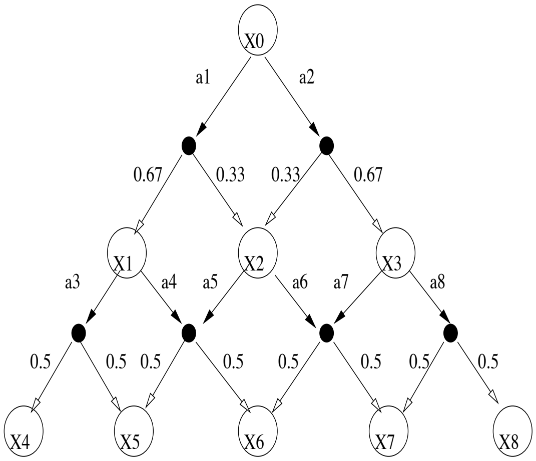

Figure 12: Example 4 – The stochastic graph for the considered problem instances

the other hand, the problem instance defined by $\mathcal{N} = (1, 2, 2, 2, 1)$ has a constant ratio $\mathcal{N}_i/\rho_i^{rel}$ across all $i = 4, \ldots, 8$. Figures 13 and 14 report the performance of the policies π^{rel}, π^{adrel} and π^{roll} in each of these two cases, as the corresponding vector $\mathcal{N}$ is scaled to increasingly larger values. The reported values for the policy π^{rel} were obtained from the closed-form expression that characterizes the performance of a simple randomized policy π as a function of the corresponding probability vector ρ^{π}, that was derived in Section 3.2. The performance of the policies π^{adrel} and π^{roll} was estimated through simulation. As expected from Theorem 8, in the case of the visitation requirement vector $\mathcal{N} = (3, 1, 1, 0, 0)$, the performance of all three policies converges very fast to the lower bound $V_{rel}^*(n)$ – c.f. Figure 13. On the other hand, the ties of the ratios $\mathcal{N}_i/\rho_i^{rel}$, $i = 4, \ldots, 8$, in the case of the visitation requirement vector $\mathcal{N} = (1, 2, 2, 2, 1)$, result in the divergence of the performance of the considered policies from the lower bound $V_{rel}^*(n)$ – c.f. Figure 14. However, as expected, the distance of the performance of these policies from $V_{rel}^*(n)$ increases in a slow, sub-linear manner with respect to n, so that the corresponding ratios $V^{\pi}(n)/V_{rel}^*(n)$ decrease to one. Finally, it is worth-noticing that π^{adrel} outperforms again the other two policies, demonstrating a performance that is pretty close to the lower bound $V_{rel}^*(n)$. □

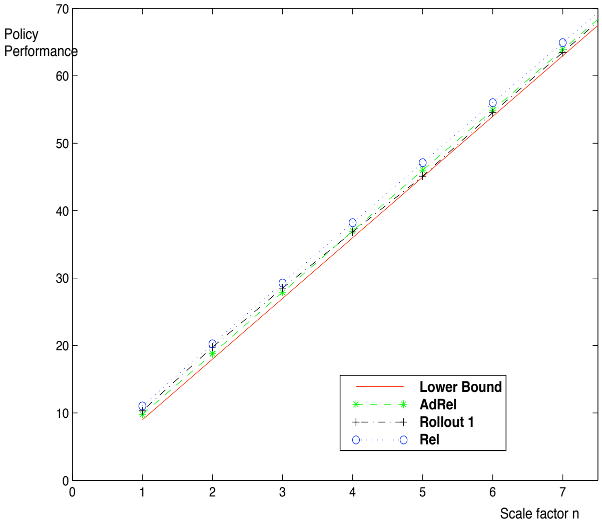

Figure 13: Example 4 – The performance of various simple and adaptive randomized policies compared to the lower bound $V^*_{rel}(n)$, for the basic visitation requirement vector $\mathcal{N} = (3, 1, 1, 0, 0)$ and $n = 1, \ldots, 7$

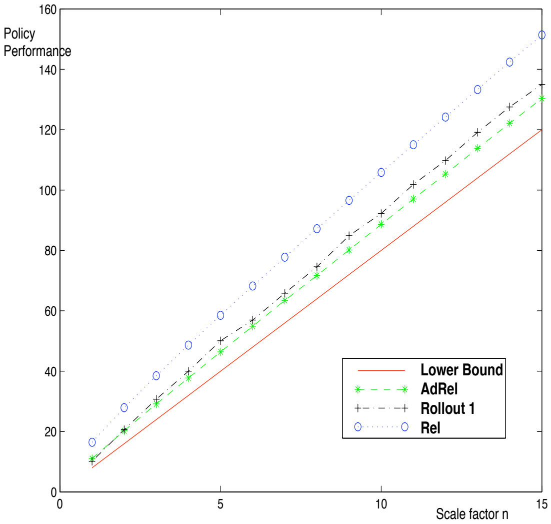

Figure 14: Example 4 – The performance of various simple and adaptive randomized policies compared to the lower bound $V^*_{rel}(n)$, for the basic visitation requirement vector $\mathcal{N} = (1, 2, 2, 2, 2, 1)$ and $n = 1, \ldots, 15$

3.4 Discussion

In this chapter we introduced the problem of the optimal node visitation in acyclic stochastic digraphs, and developed a number of suboptimal but computationally efficient policies for it that are expected to demonstrate very good performance, especially as the posed visitation requirements grow to larger values. The presented results are motivated by and are similar in spirit to some recent developments in stochastic scheduling theory and the suboptimal control of Markov Decision Processes.

A remaining open issue is the formal investigation of the computational complexity of the considered ONV problem. Some partial results along this line are provided in Chapter 6 of this document. In the next chapter, we present an analytic treatment of the performance of π^{adrel} and we prove that for a broad set of requirement vector choices, its expected performance is within a constant from the optimal as the requirement vector is uniformly scaled to infinity. On the other hand, Chapter 5 introduces some additional variations of the ONV problem, that are motivated by the implementational needs of the learning algorithm discussed in Chapter 2, and it extends the results developed in this chapter to these new variations.

CHAPTER IV

PERFORMANCE ANALYSIS OF POLICY π^{ADREL}

In this chapter, we complement the results of Chapter 3 by exploring the dynamics underlying policy π^{adrel}, the adaptive implementation of the randomized policy π^{rel}. We remind the reader that policy π^{adrel} will revise the routing probabilities every time a visitation requirement is satisfied, by formulating and re-solving the relaxing-LP. The computational studies reported in Chapter 3 showed that π^{adrel} has an excellent performance and outperforms any other suboptimal policy applied on the ONV problem. These computational findings are backed up by the key result presented in this chapter, that for a large set of requirement vector choices, the expected performance of π^{adrel} is $O(1)$-i.e, it is within a constant factor from the optimal- as the visitation requirement vector is uniformly scaled to infinity.

4.1 An alternative characterization of the relaxing-LP

In this section we provide with an alternative characterization of the relaxing-LP, defined in Chapter 3, that will guide the subsequent analysis of this chapter. Recall that the relaxing-LP is given by the following linear program:

$$\min \sum_{a \in \mathcal{A}(x^0)} \chi_a \tag{103}$$

s.t.

$$\sum_{a: x \in \mathcal{S}(a)} p(x;a) \cdot \chi_a \;=\; \sum_{a \in \mathcal{A}(x)} \chi_a, \quad \forall x \in X \backslash (\{x^0\} \cup X^L) \tag{104}$$

$$\sum_{a: x \in \mathcal{S}(a)} p(x;a) \cdot \chi_a \;\geq\; \mathcal{N}_x, \quad \forall x \in X^L,\ \mathcal{N}_x > 0 \tag{105}$$

$$\chi_a \;\geq\; 0, \quad \forall a \in \bigcup_{x \in X \backslash X^L} \mathcal{A}(x) \tag{106}$$

An interpretation of an optimal solution of the relaxing LP, χ^*, is that it constitutes a flow pattern that can satisfy the flow requirements for the terminal nodes, $x \in X^L$, expressed by the requirement vector $\mathcal{N}$. Furthermore, let ρ be a vector of dimensionality L that associates with each terminal node, $x \in X^L$, the quantity

$$\rho_x = \frac{\sum_{a:x\in\mathcal{S}(a)} p(x;a) \cdot \chi^*_a}{\sum_{a\in\mathcal{A}(x^0)} \chi^*_a}. \tag{107}$$

Vector ρ is a probability distribution vector where each component ρ_x is the probability of reaching terminal node $x \in X^L$, given that the agent starts its graph traversal at the root node x^0 and subsequently follows the randomized policy π^{rel}. More generally, consider the set $\mathcal{V}$ consisting of the probability distribution vectors ρ' that correspond to the feasible solutions of the relaxing-LP. Assume the polyhedron $\mathcal{P}$ consisting of all those vectors (χ, ρ') that satisfy the following equations:

$$\begin{aligned}
\sum_{a\in\mathcal{A}(x^0)} \chi_a &= 1 && (108)\\
\sum_{a:x\in\mathcal{S}(a)} p(x;a)\cdot\chi_a &= \sum_{a\in\mathcal{A}(x)} \chi_a, \quad \forall x \in X\backslash(\{x^0\}\cup X^L) && (109)\\
\sum_{a:x\in\mathcal{S}(a)} p(x;a)\cdot\chi_a &= \rho'_x, \quad \forall x \in X^L && (110)\\
\chi_a &\geq 0, \quad \forall a \in \bigcup_{x\in X\backslash X^L} \mathcal{A}(x) && (111)\\
\rho'_x &\geq 0,\ x \in X^L, && (112)
\end{aligned}$$

and define the *projection* [8] $\Pi(\mathcal{P})$ by letting

$$\Pi(\mathcal{P}) = \{\rho' |\text{ there exist } \chi \text{ s.t. } (\chi, \rho') \in \mathcal{P}\}. \tag{113}$$

Then we have that $\mathcal{V} = \Pi(\mathcal{P})$. We proceed with the following theorem that provides an alternative characterization of the relaxing-LP in terms of probability vectors $\phi \in \mathcal{V}$.

Theorem 9 *There exist M probability vectors $\phi^1, \dots, \phi^M \in \mathcal{V}$, such that the relaxing-LP of Equations 103-106 is equivalent to the following linear program:*

$$\min\{\sum_{k=1}^{M} x_k\} \tag{114}$$

s.t.

$$\begin{aligned} \sum_{k=1}^{M} x_k \cdot \phi^k - \sum_{k=M+1}^{M+L} x_k \cdot r^{k-M} &= \mathcal{N} && (115) \\ x_k &\geq 0, \ k = 1, \dots, M+L && (116) \end{aligned}$$

where $r^1, \dots, r^L$ are the L-dimensional unit vectors.

Proof Since $\mathcal{V}$ is equal to the projection $\Pi(\mathcal{P})$ then, from Chapter 2.8 of [8], $\mathcal{V}$ is also a polyhedron. It is trivial to check that $\mathcal{V}$ is also bounded. Then, from Theorem 2.9 of [8], the set $\mathcal{V}$ is the convex hull of its extreme points. Hence, $\mathcal{V}$ can be assumed to be of the form

$$\mathcal{V} = \{\sum_{i=1}^{M} \lambda_i \cdot \phi^i \ : \ \sum_{i=1}^{M} \lambda_i = 1, \ \lambda_i \geq 0, \ i = 1, \dots, M\} \tag{117}$$

where $\phi^1, \dots, \phi^M$ are probability vectors and the extreme points of $\mathcal{V}$. From Theorem 6 of Chapter 3, the objective value of the relaxing-LP is equal to

$$\min_{\hat{\rho} \in \mathcal{V}} \max_{x \in X^L} \{\frac{\mathcal{N}_x}{\hat{\rho}_x}\} \tag{118}$$

Note that,

$$\begin{aligned} \max_{x \in X^L} \{\frac{\mathcal{N}_x}{\hat{\rho}_x}\} &= \min\{y | \ y \cdot \hat{\rho} \geq \mathcal{N}\} && (119) \\ &= \min\{y | \ y \cdot \sum_{i=1}^{M} \lambda_i \cdot \phi^i \geq \mathcal{N}\} && (120) \end{aligned}$$

where $\hat{\rho}$ is replaced by a convex combination of the extreme points of $\mathcal{V}$. By combining Equations 118 and 120, the relaxing-LP can now be equivalently written

$$\min\{y\} \tag{121}$$

s.t.

$$\begin{aligned} y \cdot \sum_{i=1}^{M} \lambda_i \cdot \phi^i &\geq \mathcal{N} && (122)\\ \sum_{i=1}^{M} \lambda_i &= 1 && (123)\\ \lambda_i &\geq 0,\ i = 1, \ldots, M, && (124)\\ x &\geq 0. && (125) \end{aligned}$$

Let $x_i = \lambda_i \cdot y,\ i = 1, \ldots, M$. Then, it is not hard to see that the linear program expressed by Equations 121-125 can be re-written as

$$\min\{\sum_{i=1}^{M} x_i\} \qquad (126)$$

s.t.

$$\begin{aligned} \sum_{i=1}^{M} x_i \cdot \phi^i &\geq \mathcal{N} && (127)\\ x_i &\geq 0,\ i = 1, \ldots, M. && (128) \end{aligned}$$

Finally, note that the linear program expressed by Equations 114-116 is the above linear program in standard form. ▮

From now on, when we refer to the relaxing-LP, we assume the formulation given by Equations 114-116. Furthermore, we shall let y^*_x denote the optimal dual variable corresponding to the primal constraint for $x \in X^L$. We shall refer to that dual linear program as the dual-relaxing-LP. A first result that will be useful in the sequel, relates the vector ρ defined by Equation 107, with the optimal solution of the dual-relaxing-LP and is given by the following lemma:

Lemma 5

$$\sum_{x:\mathcal{N}_x > 0} y^*_x \cdot \rho_x = 1. \qquad (129)$$

Proof

$$\begin{aligned}\sum_{x:\mathcal{N}_x>0} y_x^* \cdot \rho_x &= \sum_{x:\mathcal{N}_x>0} y_x^* \cdot (\frac{\sum_{a:x\in\mathcal{S}(a)} p(x;a)\cdot \chi_a^*}{\sum_{a\in\mathcal{A}(x^0)} \chi_a^*}) \qquad (130)\\ &= \sum_{x:y_x^*>0} y_x^* \cdot (\frac{\mathcal{N}_x}{\sum_{a\in\mathcal{A}(x^0)} \chi_a^*}) \qquad (131)\\ &= \frac{\sum_{x:y_x^*>0} y_x^* \cdot \mathcal{N}_x}{\sum_{a\in\mathcal{A}(x^0)} \chi_a^*} \qquad (132)\\ &= 1. \qquad (133)\end{aligned}$$

Equation 130 follows from Equation 107 and Equation 131 follows from the complementary slackness conditions satisfied by the optimal solutions $\{\chi_a^*|\ a \in \mathcal{A}(x),\ x \in X\backslash X^L\}$ and $\{y_x^*|\ x \in X\}$. Finally, Equation 133 is a direct consequence of the strong duality property; the optimal objective value of the relaxing-LP is equal to the optimal objective value of its dual. ▮

In this section we developed an alternative characterization of the relaxing-LP that is given by Theorem 9. This LP will be heavily used in the subsequent developments.

4.2 A first look into the expected performance of π^{adrel}

In order to proceed with the analysis of π^{adrel}, we recall the notion of *macro-state* that was introduced in Section 3.2.3. We remind the reader that the ONV problem state space can be partitioned into the state subsets defined by a common remaining visitation requirement vector, $\mathcal{N}^c$, each of these subsets is called a macro-state. The underlying process starts from the macro-state defined by $\mathcal{N}^c = \mathcal{N}$, and at every *macro-transition*, it proceeds to a macro-state where the corresponding vector $\mathcal{N}^{c'}$ is obtained from $\mathcal{N}^c$ by reducing one of its positive components by one unit. Furthermore, π^{adrel} revises the routing probabilities at every visited macro-state, by replacing the requirement vector $\mathcal{N}$ by the vector of the remaining visitation requirements $\mathcal{N}^c$ and re-solving the relaxing-LP. For each requirement vector $\mathcal{N}^c$, let $V^{\pi^{adrel}}(\mathcal{N}^c)$ denote the expected cost-to-go implied by π^{adrel}, let χ^{*c} be the optimal solution of the relaxing-LP corresponding to $\mathcal{N}^c$, and let $V_{rel}(\mathcal{N}^c)$ denote its optimal value. Furthermore, let ρ^c be the probability vector defined by the optimal solution χ^{*c} and Equation 107.

For the rest of this section, we will work towards characterizing the difference $V^{\pi^{adrel}}(\mathcal{N})$-$V_{rel}(\mathcal{N})$. For every requirement vector $\mathcal{N}^c$, we define the quantities $V^k(\mathcal{N}^c)$, $k = 0, 1, \ldots, |\mathcal{N}^c|$ through the following recursion:

$$\begin{aligned} V^0(\mathcal{N}^c) &= V_{rel}(\mathcal{N}^c) && (134) \\ V^k(\mathcal{N}^c) &= 1 + \sum_{x:\mathcal{N}^c_x>0} \rho^c_x \cdot V^{k-1}((\mathcal{N}^c - I_x)) + \sum_{x:\mathcal{N}^c_x=0} \rho^c_x \cdot V^k(\mathcal{N}^c) && (135) \end{aligned}$$

It is worth noticing that the quantity $V^k(\mathcal{N}^c)$ can be considered as the cost-to-go of a finite horizon process that starts at the macro-state defined by $\mathcal{N}^c$, is driven by π^{adrel} for the first k macro-state transitions, and, thereafter, accumulates a boundary cost given by Equation 134. Therefore, $V^{|\mathcal{N}^c|}(\mathcal{N}^c) = V^{\pi^{adrel}}(\mathcal{N}^c)$. For every $x \in X^L$ with $\mathcal{N}^c_x > 0$, define

$$\delta^{c,k}_x = V^k(\mathcal{N}^c) - V^k(\mathcal{N}^c - I_x),\ k = 0, \ldots, |\mathcal{N}^c| - 1. \qquad (136)$$

Then Equation 135 can be re-written as

$$\begin{aligned} V^k(\mathcal{N}^c) &= \frac{1}{\sum_{x:\mathcal{N}^c_x>0} \rho^c_x} + \frac{1}{\sum_{x:\mathcal{N}^c_x>0} \rho^c_x} \cdot \sum_{x:\mathcal{N}^c_x>0} \rho^c_x \cdot V^{k-1}(\mathcal{N}^c - I_x) && (137) \\ &= \frac{1}{\sum_{x:\mathcal{N}^c_x>0} \rho^c_x} + \frac{1}{\sum_{x:\mathcal{N}^c_x>0} \rho^c_x} \cdot \sum_{x:\mathcal{N}^c_x>0} \rho^c_x \cdot (V^{k-1}(\mathcal{N}^c) - \delta^{c,k-1}_x) && (138) \\ &= V^{k-1}(\mathcal{N}^c) + \frac{1}{\sum_{x:\mathcal{N}^c_x>0} \rho^c_x} \cdot (1 - \sum_{x:\mathcal{N}^c_x>0} \rho^c_x \cdot \delta^{c,k-1}_x) && (139) \end{aligned}$$

For any two visitation requirement vectors $\mathcal{N}^c, \mathcal{N}^{c'}$ such that $\mathcal{N}^{c'} \le \mathcal{N}^c$, let $p_{adrel}(\mathcal{N}^c,$ $\mathcal{N}^{c'})$ denote the probability that the underlying process will visit macro-state $\mathcal{N}^{c'}$ given that it starts from $\mathcal{N}^c$, and it is guided by π^{adrel}. Then, the following proposition will be useful in the sequel:

Proposition 5 *For every $\mathcal{N}^c$ and $0 \leq k \leq |\mathcal{N}^c| - 1$, we have that,*

$$V^{k+1}(\mathcal{N}^c) - V^k(\mathcal{N}^c) = \sum_{\mathcal{N}^{c'} \leq \mathcal{N}^c, |\mathcal{N}^{c'}| = |\mathcal{N}^c| - k} p_{adrel}(\mathcal{N}^c, \mathcal{N}^{c'}) \cdot (V^1(\mathcal{N}^{c'}) - V^0(\mathcal{N}^{c'})) \quad (140)$$

Proof Assume a vector $\mathcal{N}^c$. We will prove Proposition 5 with an induction argument. For $k = 0$, Equation 140 holds trivially. Assume that Proposition 5 holds for $k = n-1$. Then we have that

$$\begin{aligned}
& V^{n+1}(\mathcal{N}^c) - V^n(\mathcal{N}^c) \\
= & \frac{1}{\sum_{x:\mathcal{N}^c_x>0} \rho^c_x} \cdot \sum_{x:\mathcal{N}^c_x>0} \rho^c_x \cdot (V^n(\mathcal{N}^c - I_x) - V^{n-1}(\mathcal{N}^c - I_x)) \\
= & \frac{1}{\sum_{x:\mathcal{N}^c_x>0} \rho^c_x} \cdot \sum_{x:\mathcal{N}^c_x>0} \rho^c_x \cdot \\
& \left(\sum_{\mathcal{N}^{c'}:\ |\mathcal{N}^{c'}| = |\mathcal{N}^c - I_x| - (n-1)} p_{adrel}(\mathcal{N}^c - I_x, \mathcal{N}^{c'}) \cdot (V^1(\mathcal{N}^{c'}) - V^0(\mathcal{N}^{c'})) \right) \\
= & \frac{1}{\sum_{x:\mathcal{N}^c_x>0} \rho^c_x} \cdot \sum_{x:\mathcal{N}^c_x>0} \rho^c_x \cdot \\
& \left(\sum_{\mathcal{N}^{c'}:\ |\mathcal{N}^{c'}| = |\mathcal{N}^c| - n} p_{adrel}(\mathcal{N}^c - I_x, \mathcal{N}^{c'}) \cdot (V^1(\mathcal{N}^{c'}) - V^0(\mathcal{N}^{c'})) \right) \\
= & \sum_{\mathcal{N}^{c'}:\ |\mathcal{N}^{c'}| = |\mathcal{N}^c| - n} p_{adrel}(\mathcal{N}^c, \mathcal{N}^{c'}) \cdot (V^1(\mathcal{N}^{c'}) - V^0(\mathcal{N}^{c'}))
\end{aligned}$$

The first equality follows from Equation 137 whereas the second equality is a direct consequence of the induction hypothesis. Finally, the fourth equality follows from the definition of the probabilities $p_{adrel}(\mathcal{N}^c, \mathcal{N}^{c'})$. Hence, the induction is complete and, thus, Equation 140 holds for all k. ∎

Now we are ready to provide a first characterization of the difference $V^{\pi^{adrel}}(\mathcal{N}) -$

$V_{rel}(\mathcal{N})$. From Proposition 5 we get that

$$
\begin{aligned}
& V^{\pi^{adrel}}(\mathcal{N}) - V_{rel}(\mathcal{N}) && (141)\\
= & \sum_{k=0}^{|\mathcal{N}|-1} (V^{k+1}(\mathcal{N}) - V^{k}(\mathcal{N})) && (142)\\
= & \sum_{k=0}^{|\mathcal{N}|-1} \sum_{\mathcal{N}^c \leq \mathcal{N}, |\mathcal{N}^c| = |\mathcal{N}|-k} p_{adrel}(\mathcal{N}, \mathcal{N}^c) \cdot (V^1(\mathcal{N}^c) - V^0(\mathcal{N}^c)) && (143)\\
= & \sum_{\mathcal{N}^c \leq \mathcal{N}} p_{adrel}(\mathcal{N}, \mathcal{N}^c) \cdot (V^1(\mathcal{N}^c) - V^0(\mathcal{N}^c)) && (144)
\end{aligned}
$$

For the rest of this section, we work towards a refined version of Equation 144. In fact, the sum on the right hand side of Equation 144 can be restricted to a smaller set of requirement vectors $\mathcal{N}^c$. In order to see this, let's take a closer look at the quantities $V^0(\mathcal{N}^c)$ and $V^1(\mathcal{N}^c)$. From Equations 139-134, we have that

$$
\begin{aligned}
& V^1(\mathcal{N}^c) - V^0(\mathcal{N}^c) && (145)\\
= & \frac{1}{\sum_{x:\mathcal{N}_x^c>0} \rho_x^c} \cdot (1 - \sum_{x:\mathcal{N}_x^c>0} \rho_x^c \cdot \delta_x^{c,0}) && (146)\\
= & \frac{1}{\sum_{x:\mathcal{N}_x^c>0} \rho_x^c} \cdot (1 - \sum_{x:\mathcal{N}_x^c>0} \rho_x^c \cdot (V_{rel}(\mathcal{N}^c) - V_{rel}(\mathcal{N}^c - I_x)) && (147)
\end{aligned}
$$

The differences $V_{rel}(\mathcal{N}^c) - V_{rel}(\mathcal{N}^c - I_x)$ that appear at the right hand side of Equation 147 are bounded by the dual variables of the dual-relaxing-LP. That is, if $\{y_x^{c^*} | \ x \in X\}$ is an optimal solution of the Dual-Relaxing-LP, when the right hand side vector $\mathcal{N}$ is replaced by the vector $\mathcal{N}^c$, then we have:

Lemma 6

$$V_{rel}(\mathcal{N}^c) - V_{rel}(\mathcal{N}^c - I_x) \leq y_x^{c^*}, \quad \forall x \in X^L, \ \mathcal{N}_x^c > 0. \qquad (148)$$

Proof The vector y^{c^*} is a *subgradient* [8] of V_{rel} at $\mathcal{N}^c$ that is,

$$V_{rel}(\mathcal{N}^c) + (y^{c^*})^T \cdot (\mathcal{N}^{c'} - \mathcal{N}^c) \leq V_{rel}(\mathcal{N}^{c'}), \ \forall \mathcal{N}^{c'}. \qquad (149)$$

Equation 148 now follows if we set $\mathcal{N}^{c'} = \mathcal{N}^c - I_x$ at Equation 149. ▮

Now, from Lemmas 5 and 6, Equation 147 becomes

$$V^1(\mathcal{N}^c) - V^0(\mathcal{N}^c) \geq \frac{1}{\sum_{x:\mathcal{N}_x^c>0} \rho_x^c} \cdot (1 - \sum_{x:\mathcal{N}_x^c>0} \rho_x^c \cdot y_x^{c^*}) \tag{150}$$

$$= 0 \tag{151}$$

It becomes clear that if we define the set

$$\mathcal{M} = \{\mathcal{N}^c | \; V^0(\mathcal{N}^c) < V^1(\mathcal{N}^c)\}, \tag{152}$$

then, from Equation 151, Equation 144 can be refined to

$$V^{\pi^{adrel}}(\mathcal{N}) - V_{rel}(\mathcal{N}) = \sum_{\mathcal{N}^c \in \mathcal{M}} p_{adrel}(\mathcal{N}, \mathcal{N}^c) \cdot (V^1(\mathcal{N}^c) - V^0(\mathcal{N}^c)). \tag{153}$$

The nature of the set $\mathcal{M}$ To make use of Equation 153, it is necessary to further investigate the set $\mathcal{M}$ defined by Equation 152. From Equation 139 we get that $\mathcal{N}^c \in \mathcal{M}$ if and only if $1 - \sum_{x:\mathcal{N}_x^c>0} \rho_x^c \cdot \delta_x^{c,0} > 0$. This last inequality is equivalent to

$$1 - \sum_{x:\mathcal{N}_x^c>0} \rho_x^c \cdot (V_{rel}(\mathcal{N}^c) - V_{rel}(\mathcal{N}^c - I_x)) > 0. \tag{154}$$

Furthermore, from Lemmas 5 and 6, we get that Equation 154 holds if and only if there is an $x \in X^L$, $\mathcal{N}_x^c > 0$, such that

$$V_{rel}(\mathcal{N}^c) - V_{rel}(\mathcal{N}^c - I_x) < y_x^{c^*}. \tag{155}$$

Now, it is known from duality theory that Equation 155 holds if the two relaxing-LP's corresponding to $\mathcal{N}^c$ and $\mathcal{N}^c - I_x$ have a different optimal basis. Hence, we conclude that the set $\mathcal{M}$ consists of those vectors $\mathcal{N}^c$ for which there is at least one $x \in X^L$, $\mathcal{N}_x^c > 0$, such that the relaxing-LP's corresponding to the requirement vectors $\mathcal{N}^c$ and $\mathcal{N}^c - I_x$ have different optimal bases. This interpretation of the set $\mathcal{M}$ will be extremely useful when we make use of Equation 153.

A result that will further facilitate the use of Equation 153 concerns the following uniform bounding of the differences $V^1(\mathcal{N}^c) - V^0(\mathcal{N}^c)$ for all $\mathcal{N}^c$:

Theorem 10 *There exist a positive number $K > 0$ such that*

$$V^1(\mathcal{N}^c) - V^0(\mathcal{N}^c) \leq K,\ \forall \mathcal{N}^c \tag{156}$$

Proof It follows from Equations 136 and 139 that

$$V^1(\mathcal{N}^c) - V^0(\mathcal{N}^c) \leq \frac{1}{\sum_{x:\mathcal{N}_x^c>0} \rho_x^c}. \tag{157}$$

Now suppose that for every $\epsilon > 0$, there is a requirement vector $\mathcal{N}^{c'}$, such that the corresponding probability vector $\rho^{c'}$, given by

$$\rho^{c'} = \arg\min_{\hat{\rho}\in\mathcal{V}} \max_{x:\mathcal{N}_x^{c'}>0} \{\frac{\mathcal{N}_x^{c'}}{\hat{\rho}_x}\}, \tag{158}$$

satisfies $\sum_{x:\mathcal{N}_x^{c'}>0} \rho_x^{c'} \leq \epsilon$. Furthermore, let $\rho^* = \arg\min_{hatrho\in\mathcal{V}} \max_{x\in X^L}\{\frac{1}{\hat{\rho}}\}$ and assume that ϵ is chosen small enough to satisfy $\epsilon < \rho_x^*,\ \forall x \in X^L$. Then, we have that

$$\begin{aligned}\max_{x:\mathcal{N}_x^{c'}>0} \{\frac{\mathcal{N}_x^{c'}}{\rho_x^{c'}}\} &\geq \max_{x:\mathcal{N}_x^{c'}>0} \{\frac{\mathcal{N}_x^{c'}}{\epsilon}\} && (159)\\ &> \max_{x:\mathcal{N}_x^{c'}>0} \{\frac{\mathcal{N}_x^{c'}}{\rho_x^*}\} && (160)\end{aligned}$$

Now, from Equations 158 and 160, we reach a contradiction. Hence, $\sum_{x\in X^L} \rho_x^c$ is bounded from below for every $\mathcal{N}^c$. Finally, Theorem 10 follows immediately from Equation 157. ▮

In this section, we investigated the difference $V^{\pi^{adrel}}(\mathcal{N}) - V_{rel}(\mathcal{N})$ and derived Equation 153. The main tools that we will use in the sequel is the alternative relaxing-LP characterization given by Theorem 9, Equation 153 and Theorem 10.

4.3 Some observations on the optimal solution of the relaxing-LP

In this section, we take a closer look at the optimal solution of the relaxing-LP and uncover some useful properties of it. The linear program expressed by Equations 114-116, is a *standard form* linear program with L equality constraints and $M+L$ variables.

A vector $\mathbf{x}$ is a *basic solution* if there exist indices $B(1), \ldots, B(L)$ such that the corresponding problem columns are linearly independent and $x_i = 0, i \neq B(1), \ldots B(L)$. The variables $x_i,\ i = B(1), \ldots, B(L)$, are called *basic variables.* Let $\mathbf{x}^*$ be an optimal solution of the linear program expressed by equations 114-116, let $m \geq 1$ be the number of the basic variables x_i for which $1 \in \{1, \ldots, M$ and let

$$\mathbf{B} = [\phi^{B(1)}, \ldots, \phi^{B(m)}, r^{B(m+1)-M}, \ldots, r^{B(L)-M}] \tag{161}$$

be the corresponding optimal basis matrix. Then, we have $x_i = 0$ for every non-basic variable, while the vector of the basic variables, $\mathbf{x}_B^* = [x_{B(1)}, \ldots, x_{B(L)}]$, is given by

$$\mathbf{x}_B^* = \mathbf{B}^{-1} \cdot \mathcal{N} \tag{162}$$

Furthermore, for a standard form linear program with m constraints and n variables $(m < n)$, a basic solution $\mathbf{x}$ is said to be *degenerate* if more than $n - m$ of the components of $\mathbf{x}$ are zero. In other words, a basic solution is degenerate if one or more basic variables is zero. Hence, for the relaxing LP expressed by Equations 114-116, we have the following observation:

Observation 1*: The optimal solution $\boldsymbol{x}^*$ is non-degenerate if and only if $\boldsymbol{B}^{-1} \cdot \mathcal{N} > 0$.*

If we replace the requirement vector $\mathcal{N}$ of Equation 115 by the vector $\mathcal{N}^c$, we want to check whether the basis expressed by the matrix $\mathbf{B}$ is still optimal. The basic matrix $\mathbf{B}$ is optimal if (i) $\mathbf{B}^{-1} \cdot \mathcal{N}^c \geq \mathbf{0}$ and (ii) the reduced costs are no-negative. Since the reduced costs do not involve the vector $\mathcal{N}^c$, this leads us to the following observation:

Observation 2*: If we replace the vector $\mathcal{N}$ of Equation 115 by a vector $\mathcal{N}^c$, the basis $\boldsymbol{B}$ remains optimal if $\boldsymbol{B}^{-1} \cdot \mathcal{N}^c \geq \boldsymbol{0}$.*

We provide with a definition and two additional observations that will be useful in the sequel.

Definition 2 *Let the i^{th} row of the matrix $\boldsymbol{B}^{-1}$ be given by $(b^i)^T = [b_1^i, \ldots, b_L^i]$.*

The next observation concerns the first m rows $(b^i)^T,\ i = 1, \ldots, m$. Recall that m is the number of probability vectors $\phi^{B(i)},\ i = 1, \ldots, m$, that belong to the optimal basis $\mathbf{B}$. Then,

Observation 3:

$$b^i_j = 0,\ 1 \leq i \leq m,\ j \in \{B(m+1) - M, \ldots, B(L) - M\}. \tag{163}$$

To prove Equation 163, notice that, as a direct consequence of Equation 161, the relation $\mathbf{B}^{-1} \cdot \mathbf{B} = \mathbf{I}$, and the definition of the vectors $b^i,\ i = 1, \ldots, L$, we have that

$$(b^i)^T \cdot r_{B(m+1)-M} = \ldots = (b^i)^T \cdot r_{B(L)-M} = 0,\ 1 \leq i \leq m. \tag{164}$$

We also have that

$$(b^i)^T \cdot r_j = b^i_j,\ i, j = 1, \ldots, L \tag{165}$$

Now, Equation 163 follows from Equation 164 and 165.

The next observation concerns the rows $(b^i)^T,\ i = m+1, \ldots, L$. Each such row i corresponds to a terminal node $B(i) - M$. Under the assumption that $\mathbf{B}$ is non-degenerate, the terminal nodes $B(i) - M,\ i = m+1, \ldots, L$ receive fluid that exceeds their respective requirement. We notice that

Observation 4:

$$b^i_j = \begin{cases} 0, & \text{if } j \in \{B(m+1) - M, \ldots, B(L) - M\} \backslash \{B(i) - M\}, \\ -1, & \text{if } j = B(i) - M, \end{cases} \tag{166}$$

and

$$(b^i)^T \cdot \phi_{B(j)} = 0,\ m+1 \leq i \leq L,\ j = 1, \ldots, m. \tag{167}$$

From know on, we define

$$j(i) = B(i) - M,\ i = m+1, \ldots, L. \tag{168}$$

and let

$$\mathcal{B} = \{1, \ldots, L\} \backslash \{B(m+1) - M, \ldots, B(L) - M\} \tag{169}$$

Then for every requirement vector $\mathcal{N}^c$ and $i \in \{m+1, \dots, L\}$, we have

$$\sum_{j=1}^{L} b_j^i \cdot \mathcal{N}_j^c = \sum_{j \in \mathcal{B}} b_j^i \cdot \mathcal{N}_j^c - \mathcal{N}_{j(i)}^c. \tag{170}$$

Recapitulating this section, we defined the notation for the optimal basis and the corresponding solution of the relaxing-LP, and we provided some properties of them. In particular, Observation 1 provided a non-degeneracy condition for the optimal solution of the relaxing-LP, Observation 2 provided an optimality condition for the optimal basis $\mathbf{B}$, and Observations 3 and 4 provided some properties of the rows of $\mathbf{B}^{-1}$ that will be useful in the sequel. Now we are ready to explore in more detail the dynamics governing the ONV problem evolution under π^{adrel}.

4.4 The dynamics of the ONV problem under π^{adrel}

The policy π^{adrel} is obtained by re-solving the relaxing-LP at every visited macro-state. For the rest of this section, when we refer to π^{adrel}, we assume that the process starts at the macro-state defined by $\mathcal{N}^c = n \cdot \mathcal{N}$, and at every macro-transition, it proceeds to a macro-state where the corresponding vector $\mathcal{N}^{c'}$ is obtained by $\mathcal{N}^c$ by reducing one of its positive components by one unit. We also let $W^k, k = 1, \dots, n \cdot |\mathcal{N}|$ denote the requirement vector characterizing the k^{th} visited macro-state.

Since the starting macro-state is characterized by a scaled requirement vector $n \cdot \mathcal{N}$, the optimal basis characterizing the macro-state, is the same basis $\mathbf{B}$ corresponding to vector $\mathcal{N}$. For the initial macro-state, we distinguish two sets of terminal nodes. Those nodes j that receive exactly $n \cdot \mathcal{N}_j$ volume of fluid in the optimal solution of the relaxing-LP, and those that the received fluid exceeds their respective requirement. Given that the optimal solution of the relaxing-LP, described by Equations 114-116, is non-degenerate, we can identify these two sets of nodes by examining the unit vectors r^i, $i = 1, \dots, L$, that are members of the optimal basis $\mathbf{B}$. If, for example, r^1 belongs to $\mathbf{B}$ then the terminal node 1 receives a fluid volume that exceeds its requirement $n \cdot \mathcal{N}_1$. From the description of the basis $\mathbf{B}$, given by Equation 161, the set $\{B(m+1)-M,\ldots,B(L)-M\}$ contains the indices of the terminal nodes that receive positive excess fluid. Hence,the set $\mathcal{B}$ defined by Equation 169 admits the following natural interpretation:

Property B1: For a non-degenerate optimal basis $\mathbf{B}$, the set $\mathcal{B}$ defined by Equation 169 contains the indices of those terminal nodes that receive no excess fluid.

The set of terminal nodes $\mathcal{B}$ will play an important role in the analysis of the expected performance of π^{adrel}. As we will see later, the reductions of the visitation requirements at nodes belonging in set $\mathcal{B}$ play a crucial role in the shaping of the performance of π^{adrel}. This understanding will be an important step towards the main result of this chapter.

Now, let's take a look at the macro-state transition probabilities under π^{adrel}. Recall that W^k denotes the requirement vector characterizing the k^{th} visited macrostate. Furthermore, recall that the terminal node fluid pattern defines the probability distribution of visiting the graph terminal nodes. The following definition will be useful in the sequel:

Definition 3 *Let the terminal node fluid pattern, at the k^{th} macro-state, be given by the vector $f^k = (f_1^k, \ldots, f_L^k)$ and let $f_0^k = \sum_{j:W_j^k>0} f_j^k$ be the total amount of fluid that reaches the nodes with a positive visitation requirement.*

Then the probability of transitioning out of the k^{th} macro-state through terminal node $j : W_j^k > 0$ is given by $p_j^k = \frac{f_j^k}{f_0^k}$. Notice that p_j^k is the conditional probability that the process will visit terminal node $j : W_j^k > 0$ given that the process will visit a terminal node with a positive visitation requirement. Hence, we can state that:

Property B2: Given a requirement vector W^k, the probability of a macro-state transition through terminal node $j : W_j^k > 0$ is given by $p_j^k = \frac{f_j^k}{f_0^k}$.

Next we consider the impact of a requirement reduction (i) on the optimal solution of the relaxing-LP, and, (ii) on the implied terminal node fluid pattern. Assume that we are on the k^{th} macro-state, with a requirement vector W^k, the optimal basis

B is given by Equation 161 and it is non-degenerate. Then the vector of the basic variables is given by $\mathbf{B}^{-1} \cdot W^k$ and, hence, the values of the L basic variables are equal to $\sum_{j=1}^{L} b_j^i \cdot W_j^k,\ i = 1, \ldots, L$. Assume now, that we have a requirement reduction at a node $j \notin \mathcal{B}$, that is, at a node with an excess flow and let W^{k+1} be the resulting requirement vector. Since the reduction is made at a node $j \notin \mathcal{B}$, we must have that $W_i^{k+1} = W_i^k,\ i \in \mathcal{B}$. From Observation 3 we know that $b_j^i = 0$for $1 \leq i \leq m$ and $j \notin \mathcal{B}$. Hence, for $1 \leq i \leq m$, we have that

$$\begin{aligned}
\sum_{j=1}^{L} b_j^i \cdot W_j^{k+1} &= \sum_{j \in \mathcal{B}} b_j^i \cdot W_j^{k+1} &\quad (171)\\
&= \sum_{j \in \mathcal{B}} b_j^i \cdot W_j^k &\quad (172)\\
&> 0 &\quad (173)
\end{aligned}$$

On the other hand, from Observation 4, we know that $b_j^i = 0$ or $b_j^i = -1$ for $m + 1 \leq i \leq L$ and $j \notin \mathcal{B}$. Hence, for $m + 1 \leq i \leq L$ we have that

$$\begin{aligned}
\sum_{j=1}^{L} b_j^i \cdot W_j^{k+1} &= \sum_{j \in \mathcal{B}} b_j^i \cdot W_j^{k+1} - W_{j(i)}^{k+1} &\quad (174)\\
&\geq \sum_{j \in \mathcal{B}} b_j^i \cdot W_j^k - W_{j(i)}^k &\quad (175)\\
&= \sum_{j=1}^{L} b_j^i \cdot W_j^k &\quad (176)\\
&> 0 &\quad (177)
\end{aligned}$$

Hence, from Equations 173 and 177, we get that

$$\sum_{j=1}^{L} b_j^i \cdot W_j^{k+1} > 0,\ i = 1, \ldots, L. \quad (178)$$

Equation 178 implies that, after a requirement reduction from a node $j \notin \mathcal{B}$, the basis **B** remains optimal and non-degenerate. Furthermore, the fluid that reaches each terminal node is unaffected. To see this, notice that the terminal node fluid pattern depends on the first m basic variables given by $\sum_{j=1}^{L} b_j^i \cdot W_j^k,\ i = 1, \ldots, m$. Each such variable $\sum_{j=1}^{L} b_j^i \cdot W_j^k$, denotes a fluid volume that is distributed to the terminal nodes according to proportions defined by the probability vector $\phi_{B(i)}$, $i = 1, \ldots, m$. We already argued that the basis $\mathbf{B}$ remains optimal for W^{k+1}. Also, Equations 171-172 imply that the first m basic variables remain the same for the relaxing-LP's corresponding to the vectors W^k and W^{k+1}. Therefore, the terminal node fluid pattern remains unaffected for the macro-states W^k and W^{k+1}. Hence, we conclude that:

Property B3: Given a non-degenerate optimal basis $\mathbf{B}$, a requirement reduction from a terminal node $j \notin \mathcal{B}$ does not affect either the optimal basis or the terminal node fluid pattern induced by the new relaxing-LP.

Property B3 further implies that if we replace the requirement vector $n \cdot \mathcal{N}$ by $n \cdot \mathcal{N}'$, where

$$\mathcal{N}'_x = \begin{cases} \mathcal{N}_x, \ x \in \mathcal{B}, \\ \\ 0, \ \text{otherwise}, \end{cases} \tag{179}$$

then the new relaxing-LP corresponding to $n \cdot \mathcal{N}'$, has the same optimal basis $\mathbf{B}$ and the terminal node fluid pattern remains the same. Therefore,

Property B4: The optimal value of the relaxing-LP and the terminal node fluid pattern remains unaffected if we replace $n \cdot \mathcal{N}$ by $n \cdot \mathcal{N}'$ defined by Equation 179.

Next, we focus on the ONV problem with the visitation requirement vector $\mathcal{N}'$ defined by Equation 179. We have already argued that for $W^0 = n \cdot \mathcal{N}'$ and while the optimal basis is equal to $\mathbf{B}$, the flow reaching a node $j \in \mathcal{B}$ is exactly equal to the remaining visitation requirement. An immediate implication is that, for W^0 and the subsequent macro-states W^k where the optimal basis is $\mathbf{B}$, we choose a node j with probability $\frac{W_j^k}{n|\mathcal{N}'|-k}$, where W_j^k is the number of remaining requirements for the terminal node j. Notice that, while the optimal basis is $\mathbf{B}$, the requirement reduction process for this ONV problem is a sampling without replacement from a population of $n \cdot |\mathcal{N}'|$ objects with $n \cdot \mathcal{N}'_j$ objects of each kind.

Property B5: For $W^0 = n \cdot \mathcal{N}'$ and all the subsequent macro-states where the optimal basis remains equal to $\mathbf{B}$, the requirement reduction process under π^{adrel} is a sampling without replacement from a population of $n \cdot |\mathcal{N}'|$ objects with $n \cdot \mathcal{N}'_j,\ j \in \mathcal{B}$, objects of each kind.

Properties B4 and B5 provide with some properties for the ONV problem corresponding to the truncated requirement vector $\mathcal{N}'$, given by Equation 179. Our further line of analysis will first explore the expected performance of π^{adrel} on this modified ONV problem and it will show it to be $O(1)$ with respect to the performance of the corresponding optimal policy. Subsequently, in a second step, we shall prove that the performance attained by π^{adrel} on the modified ONV problem is close to that attained by the same policy on the original ONV problem.

Summarizing this section, we took a closer look at the solution of the relaxing-LP and its implied macro-state transition dynamics. In particular, in Property B1 we identified a set of terminal nodes that will play a pivotal role in the subsequent developments. In Property B2, we characterized the macro-state transitions according to the fluid volumes reaching the set of terminal nodes. In Property B3, we examined the implications that a requirement reduction has on the solution of the relaxing-LP. Finally, in Properties B4 and B5 we examined an ONV problem version that is defined by the reduced requirement vector given by Equation 179. The next section considers the performance of π^{adrel} on this modified ONV problem.

4.5 Asymptotic optimality of π^{adrel} on the modified ONV problem

In this section we examine the ONV problem where the requirement vector $\mathcal{N}$ is replaced by the truncated vector $\mathcal{N}'$ given by Equation 179. We prove that, under the assumption of the non-degeneracy of the basis $\mathbf{B}$, the expected performance of π^{adrel} is $O(1)$ from the optimal as the requirement vector $\mathcal{N}'$ is scaled by a factor $n \in \mathbb{Z}^+$ to infinity. This is an important step towards proving the same result for the original vector $\mathcal{N}$, since, as we will see in the next section, the two problems do not differ significantly under π^{adrel}.

The starting point for the subsequent derivation will be Equation 153 of Section 4.2. Hence, first we show that

$$\lim_{n\to\infty} \sum_{\mathcal{N}^c \in \mathcal{M}} p_{adrel}(n \cdot \mathcal{N}', \mathcal{N}^c) < \infty. \tag{180}$$

We remind the reader that for any requirement vectors $\mathcal{N}^c$ and $\mathcal{N}^{c'}$, $p_{adrel}(\mathcal{N}^c, \mathcal{N}^{c'})$ denotes the probability that the underlying process will visit macro-state $\mathcal{N}^{c'}$ given that it starts from $\mathcal{N}^c$, and it is guided by π^{adrel} thereafter. In order to prove Equation 180 we should also recall the discussion towards the end of Section 4.2 where we argued that the set of requirement vectors $\mathcal{M}$, given by Equation 152, consists of vectors $\mathcal{N}^c$ with the following property: There exists a terminal node $x \in X^L : \mathcal{N}^c_x > 0$, such that the relaxing-LP's corresponding to $\mathcal{N}^c_x$ and $\mathcal{N}^c_x - I_x$ have a different optimal basis. Since the basis $\mathbf{B}$ is, by hypothesis, non-degenerate, from Observation 1 of Section 4.3 we get that the optimality constraints are strictly positive, i.e., $\sum_{j\in\mathcal{B}} b^i_j \cdot \mathcal{N}_j > 0, \ i = 1, \ldots, L$. Hence, for a large enough $n \in \mathbb{Z}_+$, we may have that $n \cdot \sum_{j\in\mathcal{B}} b^i_j \cdot \mathcal{N}_j - b^i_x > 0, \ \forall x \in X^L, \ i = 1, \ldots, L$. Therefore, from Observation 2, we conclude that the optimal basis $\mathbf{B}$ remains optimal for all requirement vectors $n \cdot \mathcal{N}' - I_x : \mathcal{N}'_x > 0$. In other words, $n \cdot \mathcal{N}' \notin \mathcal{M}$. Hence, we may assume that $W^0 = n \cdot \mathcal{N}' \notin \mathcal{M}$. In order to prove Equation 180 we consider the first time (or first macro-state transition) when the process enters the set $\mathcal{M}$ given that it started from $n \cdot \mathcal{N}', \ n \in \mathbb{Z}_+$. Therefore, we define the hitting time

$$T^n_{\mathcal{M}} = \min\{k \le n \cdot |\mathcal{N}'| : \ W^k \in \mathcal{M}, \ W^0 = n \cdot \mathcal{N}' \notin \mathcal{M}\}. \tag{181}$$

Then,

$$\sum_{\mathcal{N}^c \in \mathcal{M}} p_{adrel}(n \cdot \mathcal{N}', \mathcal{N}^c) = \sum_{k=1}^{n|\mathcal{N}'|} \sum_{\mathcal{N}^c \in \mathcal{M}:\ |\mathcal{N}^c| = n \cdot |\mathcal{N}'| - k} p_{adrel}(n \cdot \mathcal{N}', \mathcal{N}^c) \quad (182)$$

$$= \sum_{k=1}^{n|\mathcal{N}'|} E[I(W^k \in \mathcal{M})] \quad (183)$$

$$= E[\sum_{k=T^n_{\mathcal{M}}}^{n|\mathcal{N}'|} I(W^k \in \mathcal{M})] \quad (184)$$

$$\leq E[n|\mathcal{N}'| - T^n_{\mathcal{M}}]. \quad (185)$$

Therefore if we prove that

$$\lim_{n \to \infty} E[n|\mathcal{N}'| - T^n_{\mathcal{M}}] < \infty, \quad (186)$$

Equation 180 will follow. Equation 186 is proved in the next proposition.

Proposition 6 *Assume a requirement vector $\mathcal{N}$ for which the optimal solution of the relaxing-LP is non-degenerate. Then, for the requirement vector $\mathcal{N}'$ given by Equation 179, and the hitting time $T^n_{\mathcal{M}}$ defined by Equation 181, we have that*

$$\lim_{n \to \infty} E[n|\mathcal{N}'| - T^n_{\mathcal{M}}] < \infty, \quad (187)$$

Proof Before proceeding with the proof, notice that during the evolution of the ONV problem, the visitation requirements at nodes $j \notin \mathcal{B}$ will be zero. From now on, we restrict summations over the set of terminal nodes to the set $\mathcal{B}$. In order to get a first insight on the hitting time $T^n_{\mathcal{M}}$ we define the *cone*, [8],

$$\mathcal{S}_1 = \{\mathcal{N}^c | \ \mathcal{N}^c_j = 0, \ j \notin \mathcal{B}, \ \text{and} \ \sum_{j \in \mathcal{B}} b^i_j \cdot \mathcal{N}^c_j \geq 0, \ i = 1, \ldots, L\}. \quad (188)$$

It is obvious that the requirement vector characterizing the starting macro-state, $W^0 = n \cdot \mathcal{N}'$, lies strictly within the cone $\mathcal{S}_1$. As we already know, the set $\mathcal{M}$ contains those requirement vectors $\mathcal{N}^c$ with the property that there is a leaf node $x : \mathcal{N}_x > 0$, such that the relaxing LP's corresponding to $\mathcal{N}^c$ and $\mathcal{N}^c - I_x$ have different optimal bases. As a result, the states belonging to $\mathcal{S}_1 \cap \mathcal{M}$ are on the "boundary" of the cone $\mathcal{S}_1$; i.e. if $\mathcal{N}^c \in \mathcal{S}_1 \cap \mathcal{M}$, then there is an $x \in \mathcal{B}$ with $\mathcal{N}^c_x > 0$ and an $i \in \{1, \ldots, L\}$ such that

$$\sum_{j \in \mathcal{B}} b^i_j \cdot \mathcal{N}^c_j - b^i_x < 0, \text{ for some } i = 1, \ldots, L. \tag{189}$$

Therefore, when we hit the set $\mathcal{M}$ for the first time, then we have already exited or we are close to exit the cone $\mathcal{S}_1$. In the subsequent analysis, we will construct a cone $\mathcal{S}_2$ such that $\mathcal{S}_2 \subset \mathcal{S}_1$ and the set $\mathcal{S}_2 \cap \mathcal{M}$ is finite. In the sequel, we will prove that our process stays within the cone $\mathcal{S}_2$ all but a finite number of times as $n \to \infty$. This construction will naturally lead us to prove Equation 186.

To construct the cone $\mathcal{S}_2$ we pick an $\epsilon > 0$ and consider the vectors $d^i \in \mathbb{R}^L, \ i = 1, \ldots, L$, such that

$$d^i_j = \begin{cases} b^i_j - \epsilon, & j \in \mathcal{B}, \\ \\ 0, & \text{otherwise.} \end{cases} \tag{190}$$

$i = 1, \ldots, L$. Since the relaxing-LP is non-degenerate, by Observation 1 we have that $\sum_{j=1}^L b^i_j \cdot \mathcal{N}'_j > 0, \ i = 1, \ldots, L$. We choose ϵ small enough such that

$$\sum_{j \in \mathcal{B}} d^i_j \cdot \mathcal{N}'_j > 0, \ i = 1, \ldots, L. \tag{191}$$

Now define the cone

$$\mathcal{S}_2 = \{\mathcal{N}^c | \ \mathcal{N}^c_j = 0, \ j \notin \mathcal{B}, \text{ and } \sum_{j \in \mathcal{B}} d^i_j \cdot \mathcal{N}^c_j \geq 0, \ i = 1, \ldots, L\}. \tag{192}$$

Equation 191 implies that $\mathcal{N}'$ lies strictly in $\mathcal{S}_2$. Next, and, given $W^0 = n \cdot \mathcal{N}'$, consider the hitting time

$$T^n_{\mathcal{S}_2} = \min\{k \leq n \cdot |\mathcal{N}'| : \ W^k \notin \mathcal{S}_2, \ W^0 = n \cdot \mathcal{N}'\}. \tag{193}$$

The hitting time $T^n_{\mathcal{S}_2}$, defined by Equation 193, can be written as

$$T^n_{\mathcal{S}_2} \ = \ \min\{k \leq n|\mathcal{N}'| : \ \sum_{j \in \mathcal{B}} d^i_j \cdot W^k_j < 0, \text{ for some } i \in \{1, \ldots, L\}\}. \tag{194}$$

It is easy to check that $S_2 \subset S_1$. Furthermore, the set $\mathcal{S}_2 \cap \mathcal{M}$ is finite. In order to see it, assume a requirement vector $\mathcal{N}^c \in \mathcal{S}_2$ such that

$$\sum_{j \in \mathcal{B}} \mathcal{N}_j^c > \frac{\max_{i,j} b_j^i}{\epsilon}. \tag{195}$$

Then we have

$$\begin{aligned} \sum_{j \in \mathcal{B}} d_j^i \cdot \mathcal{N}_j^c \geq 0 \;\; &\Rightarrow \;\; \sum_{j \in \mathcal{B}} b_j^i \cdot \mathcal{N}_j^c \geq \epsilon \cdot \sum_{j \in \mathcal{B}} \mathcal{N}_j^c & (196) \\ &\Rightarrow \;\; \sum_{j \in \mathcal{B}} b_j^i \cdot \mathcal{N}_j^c > \max_{i,j} b_j^i. & (197) \end{aligned}$$

$i = 1, \ldots, L$. Hence, from Equations 189 and 197, we conclude that, for any $\mathcal{N}^c \in \mathcal{S}_2$ satisfying the condition given by Equation 195, $\mathcal{N}^c \notin \mathcal{M}$. Therefore, there is a constant $C = \frac{\max_{i,j} b_j^i}{\epsilon}$ such that

$$\mathcal{S}_2 \cap \mathcal{M} \subset \{\mathcal{N}^c : \; \mathcal{N}_j^c \leq C, \; j = 1, \ldots, L\}. \tag{198}$$

Consider the case where $T_{\mathcal{M}}^n \leq T_{\mathcal{S}_2}^n$. Then W^k hits the set $\mathcal{M}$ before it exits the cone $\mathcal{S}_2$. In other words, $W^{T_{\mathcal{M}}^n} \in \mathcal{S}_2 \cap \mathcal{M}$ and, from Equation 198, we have $W^{T_{\mathcal{M}}^n} \in \{\mathcal{N}^c : \mathcal{N}_j^c \leq C, \; j = 1, \ldots, L\}$. Consequently,

$$\begin{aligned} T_{\mathcal{M}}^n \leq T_{\mathcal{S}_2}^n \;\; &\Rightarrow \;\; n \cdot |\mathcal{N}'| - C \cdot L \leq T_{\mathcal{M}}^n \leq n \cdot |\mathcal{N}'| & (199) \\ &\Rightarrow \;\; n \cdot |\mathcal{N}'| \leq T_{\mathcal{M}}^n + C \cdot L. & (200) \\ &\Rightarrow \;\; T_{\mathcal{S}_2}^n \leq T_{\mathcal{M}}^n + C \cdot L. & (201) \end{aligned}$$

Equation 201 implies that, in order to prove 186, it suffices to show

$$E[n \cdot |\mathcal{N}'| - T_{\mathcal{S}_2}^n] < \infty, \text{ as } n \to \infty. \tag{202}$$

For the rest of this proof, consider the random vectors $Y^\lambda \in \mathbb{Z}_+$ defined by

$$Y^\lambda := W^{\lambda-1} - W^\lambda, \; \lambda = 1, \ldots, n \cdot |\mathcal{N}'|. \tag{203}$$

It is evident from Equation 203, that the vector Y^λ characterizes the λ^{th} macro-state transition. In particular, it associates with each terminal node $x \in X^L$ a random variable Y_x^λ, such that $Y_x^\lambda = 1$ if there is a requirement reduction at terminal node x, during the λ^{th} macro-state transition, and $Y_x^\lambda = 0$ otherwise. Then, the process W^λ can be expressed as

$$W^\lambda = n \cdot \mathcal{N}' - \sum_{k=1}^{\lambda} Y^k, \;\; \lambda = 0, \ldots, n \cdot |\mathcal{N}'|. \tag{204}$$

Define

$$\bar{d}^i = \sum_{j \in \mathcal{B}} d_j^i \cdot \frac{\mathcal{N}_j'}{|\mathcal{N}'|}, \tag{205}$$

and

$$X^{i,\lambda} = \sum_{j \in \mathcal{B}} d_j^i \cdot Y_j^\lambda \tag{206}$$

$i = 1, \ldots, L, \;\; \lambda = 1, \ldots, n \cdot |\mathcal{N}|$. Then from Equations 192, 193, 203, 205 and 206,

$$\begin{aligned} T_{\mathcal{S}_2}^n &= \min\{k \leq n \cdot |\mathcal{N}'| : \; \sum_{j \in \mathcal{B}} d_j^i \cdot (n \cdot \mathcal{N}_j' - \sum_{j \in \mathcal{B}} \cdot Y_j^\lambda) < 0, \text{ for some } i \in \{1, \ldots, L\}\} \\ &= \min\{k \leq n \cdot |\mathcal{N}'| : \; \sum_{j \in \mathcal{B}} d_j^i \cdot \sum_{\lambda=1}^{k} Y_j^\lambda > n \cdot \sum_{j \in \mathcal{B}} d_j^i \cdot \mathcal{N}_j', \text{ for some } i \in \{1, \ldots, L\}\} \\ &= \min\{k \leq n \cdot |\mathcal{N}'| : \; \sum_{\lambda=1}^{k} \sum_{j \in \mathcal{B}} d_j^i \cdot Y_j^\lambda > n \cdot \sum_{j \in \mathcal{B}} d_j^i \cdot \mathcal{N}_j', \text{ for some } i \in \{1, \ldots, L\}\} \\ &= \min\{k \leq n \cdot |\mathcal{N}'| : \; \sum_{\lambda=1}^{k} X^{i,\lambda} > n \cdot |\mathcal{N}'| \cdot \bar{d}^i, \text{ for some } i \in \{1, \ldots, L\}\} \end{aligned} \tag{207}$$

From Property B5, we know that while the optimal basis is $\mathbf{B}$, the requirement reduction Y^λ is a sampling without replacement from a population of $n \cdot \mathcal{N}'$ objects with $n \cdot |\mathcal{N}_j'|, \; j \in \mathcal{B}$, objects of each type. Hence, for $\lambda = 0, \ldots, T_{\mathcal{S}_2}^n$ and $i = 1, \ldots, L$, the random variables $X^{i,\lambda}$ can be considered as a sampling without replacement from a population of $n \cdot |\mathcal{N}'|$ objects taking values in the set $\{d_j^i : \; j \in \mathcal{B}\}$ with $n \cdot \mathcal{N}_j'$ objects having a value equal to d_j^i. Then, from Equation 191 we have $\bar{d}^i > 0, \; i = 1, \ldots, L$, and from Proposition 7 in Appendix A, Equation 202 results immediately. ▮

Now we have all the ingredients for the main result of this section. If we combine Proposition 6 and Equation 185 we get that Equation 180 holds. Then, in the light of Equation 153, if we combine Equation 180 and Theorem 10, we get the following result:

Theorem 11 *Assume a requirement vector $\mathcal{N}$ for which the optimal solution of the relaxing-LP is non-degenerate. Then, for the requirement vector $\mathcal{N}'$ given by Equation 179, we have that*

$$\lim_{n\to\infty}(V^{\pi^{adrel}}(n\cdot\mathcal{N}') - V_{rel}(n\cdot\mathcal{N}')) < \infty.$$

Now we are ready to proceed towards the main result of this chapter.

4.6 The asymptotic optimality of π^{adrel}

In the previous section, we investigated the performance of π^{adrel} on a truncated version of the requirement vector $n\cdot\mathcal{N}$, $n\cdot\mathcal{N}'$. In this section we examine the performance of π^{adrel} on the vector $n\cdot\mathcal{N}$. We show that the additional visitation requirements of $\mathcal{N}$ do not contribute to the expected cost as $\mathcal{N}$ and $\mathcal{N}'$ are scaled to infinity. In particular, we prove that

$$V^{\pi^{adrel}}(n\cdot\mathcal{N}) - V^{\pi^{adrel}}(n\cdot\mathcal{N}') < \infty, \tag{208}$$

as $n\to\infty$.

In order to prove Equation 208, we will simulate in parallel the two ONV problems corresponding to $\mathcal{N}$ and $\mathcal{N}'$ using a common random number stream. In particular, we generate in parallel the terminal node visitations using the node visitation probabilities implied by the fluid volumes reaching the terminal nodes at every visited macro-state. Let ONV-1 and ONV-2 be the problems corresponding to $\mathcal{N}$ and $\mathcal{N}'$ respectively. Keep in mind that ONV-1 and ONV-2 have a different macro-state space and, hence, at every step of the simulation, each problem may be at different macro-states.

From Property $B4$, the fluid pattern at the initial macro-state of ONV-1 and ONV-2 is the same. Hence, our simulation will generate the same terminal node visitations for the first and the subsequent simulation steps while the terminal node fluid pattern remains the same for both problems. We want to show that the dynamics of ONV-1 and ONV-2 are the same for the most part of the parallel simulation. Therefore, we are interested in the first time when the terminal fluid pattern for the ONV-1 and ONV-2 will differ, and, hence, the parallel simulation may generate different terminal node visitations. Assume that, at some simulation step, there is a reduction at some node $j \notin \mathcal{B}$ for problem ONV-1. Then, from Property $B3$, the terminal node fluid pattern will remain unaffected. The same thing will happen for problem ONV-2 since there will be no requirement reduction at all. Hence, it is safe to assume that the terminal node fluid pattern for ONV-1 and ONV-2 will differ only after a simulation step that resulted in a reduction at a node $j \in \mathcal{B}$ for both ONV-1 and ONV-2.

In the light of the above observation, we will trace the evolution of the simulation with respect to the ONV-2 problem. For reasons that will be evident later, we define the following stopping time:

Definition 4 *Let $\tau^n \in \{1, \ldots, n \cdot |\mathcal{N}'|\}$ be the macro-state transition for the ONV-2 problem where, for the first time, there is a basis change for this problem or the visitation requirements for some node $j \in \mathcal{B}$ are totally covered.*

Let D^n denote the random difference between the cost realizations of the policy π^{adrel} over the requirement vectors $n \cdot \mathcal{N}$ and $n \cdot \mathcal{N}'$. Then

$$E[D^n] = V^{\pi^{adrel}}(n \cdot \mathcal{N}) - V^{\pi^{adrel}}(n \cdot \mathcal{N}') \tag{209}$$

Now we define an event that will help us in the subsequent analysis:

Definition 5 *Let E^n be the event that up to time τ^n, (i) the terminal node fluid pattern is the same for problem ONV-1 and ONV-2, and (ii) all the visitation requirements at nodes $j \notin \mathcal{B}$ for the ONV-1 problem have been covered.*

It is important to notice that, under E^n, the cost realization for problems ONV-1 and ONV-2 will be the same under the parallel simulation, that is, $E[D^n|\ E^n] = 0$.

Hence, we can write

$$E[D^n] = E[D^n|\ E^n]P(E^n) + E[D^n|\ E'^n] \cdot P(E'^n) \tag{210}$$

$$= E[D^n|\ E'^n] \cdot (1 - P(E^n)) \tag{211}$$

Now it is easy to see that $E[D^n|\ E'^n] = O(n)$. Hence, in order to prove Equation 208, it suffices to prove that

$$1 - P(E^n) = O(n^{-1}). \tag{212}$$

For the rest of this section we will work towards proving Equation 212.

4.6.1 A closer look at the probability $P(E^n)$

For the ONV-2 problem, let $Z_j^k, j \in \mathcal{B}$, be random variables such that $Z_j^k = 1$ if there is a requirement reduction for a node $j \in \mathcal{B}$ during the k^{th} requirement reduction from the set $\mathcal{B}$ and zero otherwise. For $j \notin \mathcal{B}$, let Z_j^k be the number of node visitations to terminal node j between the $(k-1)^{th}$ and the k^{th} reduction from $\mathcal{B}$. Observe that, for the ONV-1 problem, the amount of remaining requirements at a node $j \notin \mathcal{B}$ at time τ^n is given by $(n \cdot \mathcal{N}_j - \sum_{\lambda=1}^{\tau^n} Z_j^\lambda)^+$. We are now ready to describe the event E^n using the notation given above.

From Definition 5, the event E^n is the intersection of the following two events: (i) the ONV-1 and ONV-2 problems have the same terminal fluid pattern up to the time τ^n, and, (ii) the remaining visitation requirements at nodes $j \notin \mathcal{B}$, for problem ONV-1, have been covered by time τ^n. Let's take a closer look at the first event. If the terminal fluid patterns corresponding to problems ONV-1 and ONV-2 become different some time before τ^n, then there should be a basis change for the ONV-1 problem before τ^n (by the definition of τ^n, the basis $\mathbf{B}$ of the ONV-2 problem will remain the same up to τ^n). Remember that the ONV-1 and ONV-2 problems differ because of the non-zero requirements that are present at nodes $j \notin \mathcal{B}$ of the ONV-1 problem. Hence, those requirements should be the cause of a basis change for the

ONV-1 problem before τ^n. This can happen only if the flow reaching a node $j \notin \mathcal{B}$ is not enough to cover the respective requirement at ONV-1. In mathematical terms, an optimality constraint corresponding to an index $i \in \{m+1, \ldots, L\}$ [1] becomes negative before τ^n. Recall that the relaxing-LP optimality constraints for the vector $\mathcal{N}$, are given by $\mathbf{B}^{-1} \cdot \mathcal{N} \geq 0$. Furthermore, recall that the i^{th} row of $\mathbf{B}^{-1}$ is given by the vector $(b^i)^T,\ i \in \{m+1, \ldots, L\}$. Hence, the optimality constraints corresponding to an index $i \in \{m+1, \ldots, L\}$ remain non-negative up to time τ^n if

$$\min_{k \leq \tau^n} \sum_{j=1}^{L} b^i_j \cdot W^k_j \geq 0,\ i \in \{m+1, \ldots, L\}. \tag{213}$$

From Observation 4, and for every requirement vector W^k, we can write

$$\sum_{j=1}^{L} b^i_j \cdot W^k_j = \sum_{j \in \mathcal{B}} b^i_j \cdot W^k_j - b^i_{j(i)} W^k_{j(i)},\ i \in \{m+1, \ldots, L\}. \tag{214}$$

Then Equation 213 can be written

$$\min_{k \leq \tau^n} (\sum_{j \in \mathcal{B}} b^i_j \cdot W^k_j - W^k_{j(i)}) \geq 0,\ i \in \{m+1, \ldots, L\}. \tag{215}$$

From the definition of the random vectors $Z^\lambda, \lambda = 0, \ldots, n \cdots |\mathcal{N}'|$, given at the beginning of this section, the remaining requirement $W^k_j,\ j \in \mathcal{B}$, can be replaced by $n \cdot \mathcal{N}_j - \sum_{\lambda=1}^{k} Z^\lambda_j$. On the other hand, for $j \notin \mathcal{B}$, the requirement W^k_j can be replaced by $(n \cdot \mathcal{N}_{j(i)} - \sum_{\lambda=1}^{k} Z^\lambda_{j(i)})^+$. Then, Equation 215 can be written:

$$\min_{k \leq \tau^n} (\sum_{j \in \mathcal{B}} b^i_j \cdot (n \cdot \mathcal{N}_j - \sum_{\lambda=1}^{k} Z^\lambda_j) - (n \cdot \mathcal{N}_{j(i)} - \sum_{\lambda=1}^{k} Z^\lambda_{j(i)})^+) \geq 0, \tag{216}$$

$i \in \{m+1, \ldots, L\}$. Remember that for $k \leq \tau^n$, we have that $\sum_{j \in \mathcal{B}} b^i_j \cdot (n \cdot \mathcal{N}_j - \sum_{\lambda=1}^{k} Z^\lambda_j) \geq 0,\ i \in \{m+1, \ldots, L\}$, since those last equations are necessary optimality

[1] We remind the reader that, for a requirement vector $\mathcal{N}^c$ with the corresponding relaxing-LP optimal basis equal to $\mathbf{B}$, the vector of optimal basic solutions is given by $\mathbf{B}^{-1} \cdot \mathcal{N}^c$. In the light of Equations 115 and 161, the pricing of the last $L - m$ variables denotes the excess flow at the nodes $j \notin \mathcal{B}$, i.e., each of the optimality constraints, $\sum_{j=1}^{L} b^i_j \cdot \mathcal{N}^c_j \geq 0,\ i \in \{m+1, \ldots, L\}$, denotes the excess flow at nodes $j \notin \mathcal{B}$.

constraints for the ONV-2 problem. Hence, the event expressed by Equation 216 can also be expressed as

$$\min_{k \leq \tau^n} (\sum_{j \in \mathcal{B}} b_j^i \cdot (n \cdot \mathcal{N}_j - \sum_{\lambda=1}^{k} Z_j^\lambda) - (n \cdot \mathcal{N}_{j(i)} - \sum_{\lambda=1}^{k} Z_{j(i)}^\lambda)) \geq 0, \; i \in \{m+1, \ldots, L\} \quad (217)$$

which can be further written as

$$\min_{k \leq \tau^n} \sum_{j=1}^{L} b_j^i \cdot (n \cdot \mathcal{N}_j - \sum_{\lambda=1}^{k} Z_j^\lambda) \geq 0 \quad (218)$$

$$\Rightarrow \quad \max_{k \leq \tau^n} \sum_{\lambda=1}^{k} \sum_{j=1}^{L} b_j^i \cdot Z_j^\lambda \leq n \cdot \sum_{j=1}^{L} b_j^i \cdot \mathcal{N}_j, \; i \in \{m+1, \ldots, L\}. \quad (219)$$

As we already mentioned, the event E^n is the intersection of two events, the first of which is expressed by Equation 219. Furthermore, for the E^n to be true, we want the visitation requirements at nodes $j \notin \mathcal{B}$ for problem ONV-1 to have been covered by time τ^n. This requirement can be expressed as

$$\sum_{\lambda=1}^{\tau^n} Z_j^\lambda \geq n \cdot \mathcal{N}_j, \; j \notin \mathcal{B}. \quad (220)$$

Hence, the event E^n is the intersection of the events expressed by Equations 219 and 220.

In order to simplify the above equations, for every $i \in \{m+1, \ldots, L\}$, we define

$$U^{i,k} = \sum_{\lambda=1}^{k} \sum_{j=1}^{L} b_j^i \cdot Z_j^\lambda, \quad (221)$$

and

$$G^{i,k} = \sum_{j \in \mathcal{B}} b_j^i \cdot (n \cdot \mathcal{N}_j - \sum_{\lambda=1}^{k} Z_j^\lambda), \quad (222)$$

$k = 1, \ldots, n \cdot |\mathcal{N}'|$. Then Equation 219 can be re-written as

$$\max_{k \leq \tau^n} U^{i,k} \leq n \cdot \sum_{j=1}^{L} b_j^i \cdot \mathcal{N}_j, \; i \in \{m+1, \ldots, L\}. \quad (223)$$

From Observation 4 and Equations 221-222 we have that, for $i \in \{m+1, \ldots, L\}$,

$$\begin{aligned} \sum_{\lambda=1}^{\tau^n} Z_{j(i)}^\lambda &= \sum_{\lambda=1}^{\tau^n} \sum_{j \in \mathcal{B}} b_j^i Z_j^\lambda - U^{i,\tau^n} && (224) \\ &= n \cdot \sum_{j \in \mathcal{B}} b_j^i \cdot \mathcal{N}_j - G^{i,\tau^n} - U^{i,\tau^n}. && (225) \end{aligned}$$

Then, Equation 220 can be re-written as

$$n \cdot \sum_{j \in \mathcal{B}} b^i_j \cdot \mathcal{N}_j - G^{i,\tau^n} - U^{i,\tau^n} \leq n \cdot \mathcal{N}_{j(i)},\ i \in \{m+1, \ldots, L\}, \tag{226}$$

or

$$G^{i,\tau^n} + U^{i,\tau^n} \leq n \cdot \sum_{j=1}^{L} b^i_j \cdot \mathcal{N}_j,\ i \in \{m+1, \ldots, L\}. \tag{227}$$

Hence, E^n is the intersection of the events given by Equations 223 and 227. Since, from the non-degeneracy hypothesis, $\sum_{j=1}^{L} b^i_j \cdot \mathcal{N}_j > 0,\ i = 1, \ldots, L$, we can assume the existence of a constant $\bar{b} > 0$ such that $\min_{1 \leq i \leq L} \sum_{j=1}^{L} b^i_j \cdot \mathcal{N}_j > \bar{b}$. Then the probability on the complement E'^n can be written as:

$$\begin{aligned}
P(E'^n) &\leq \sum_{i=m+1}^{L} P(\max_{k \leq \tau^n} U^{i,k} > n \cdot \sum_{j=1}^{L} b^i_j \cdot \mathcal{N}_j) && (228)\\
&+ \sum_{i=m+1}^{L} P(G^{i,\tau^n} + U^{i,\tau^n} > n \cdot \sum_{j=1}^{L} b^i_j \cdot \mathcal{N}_j) && (229)\\
&\leq \sum_{i=m+1}^{L} P(\max_{k \leq \tau^n} U^{i,k} > n \cdot \bar{b}) && (230)\\
&+ \sum_{i=m+1}^{L} P(G^{i,\tau^n} + U^{i,\tau^n} > n \cdot \bar{b}) && (231)\\
&\leq \sum_{i=m+1}^{L} P(\max_{k \leq \tau^n} U^{i,k} > n \cdot \bar{b}) && (232)\\
&+ \sum_{i=m+1}^{L} P(G^{i,\tau^n} > n \cdot \frac{\bar{b}}{2}) + \sum_{i=m+1}^{L} P(U^{i,\tau^n} > n \cdot \frac{\bar{b}}{2}) && (233)\\
&\leq 2 \sum_{i=m+1}^{L} P(\max_{k \leq \tau^n} U^{i,k} > n \cdot \frac{\bar{b}}{2}) && (234)\\
&+ \sum_{i=m+1}^{L} P(G^{i,\tau^n} > n \cdot \frac{\bar{b}}{2}) && (235)
\end{aligned}$$

In the next two sections, we concentrate on characterizing the probabilities on the right hand side of Equation 235 involving the quantities $U^{i,k}$ and $G^{i,k}$.

4.6.2 A closer look at the quantities $G^{i,k}$

Before we proceed with the analysis of the quantities $G^{i,k}$, we prove that for the most part of the ONV-2 problem, the optimal basis is $\mathbf{B}$ and the visitation requirements of the nodes $j \in \mathcal{B}$ are positive, as $n \cdot \mathcal{N}'$ is scaled to infinity. This result is of similar nature and essentially strengthens the result of Equation 186, where we prove that for the most time, the optimal basis is $\mathbf{B}$. The result is given by the following lemma:

Lemma 7

$$E[n \cdot |\mathcal{N}'| - \tau^n] < \infty \tag{236}$$

as $n \rightarrow \infty$.

Proof Before we proceed with the proof, we remind the reader that we observe the time τ^n on the evolution of the ONV-2 problem. Furthermore, we remind the reader the definition of the random vectors $Y^\lambda \in \mathbb{Z}^L, \lambda = 0, \ldots, n \cdot |\mathcal{N}'|$, given in the proof of Proposition 6: $Y^\lambda \equiv W^{\lambda-1} - W^\lambda,\ \lambda = 1, \ldots, n \cdot |\mathcal{N}'|$, i.e., the vector Y^λ characterizes the λ^{th} macro-state transition, associating with each terminal node $x \in X^L$, a random variable Y_x^λ, such that $Y_x^\lambda = 1$ if there is a requirement reduction at terminal node x, during the λ^{th} macro-state transition, and $Y_x^\lambda = 0$ otherwise.

Then, if T_0^n is the first time for the ONV-2 problem, when the visitation requirement for some node $j \in \mathcal{B}$ reduces to zero, we can write

$$\begin{aligned} T_0^n &= \min\{k \leq n \cdot |\mathcal{N}'| :\ W_i^k \leq 0,\ \text{for some } i \in \mathcal{B}\} \\ &= \min\{k \leq n \cdot |\mathcal{N}'| :\ n \cdot \mathcal{N}_i - \sum_{\lambda=1}^{k} Y_i^\lambda \leq 0,\ \text{for some } i \in \mathcal{B}\} \\ &= \min\{k \leq n \cdot |\mathcal{N}'| :\ \sum_{\lambda=1}^{k} Y_i^\lambda \geq n \cdot \mathcal{N}_i,\ \text{for some } i \in \mathcal{B}\}. \end{aligned} \tag{237}$$

If we consider the vectors $c^i \in \mathbb{Z}^L,\ i = 1, \ldots, L$, such that $c_i^i = 1$ and $c_j^i = 0,\ j \neq i$, and let

$$\bar{c}^i = \sum_{j \in \mathcal{B}} c_j^i \cdot \frac{\mathcal{N}_j}{|\mathcal{N}'|},\ i \in \mathcal{B}, \tag{238}$$

and

$$X_0^{i,\lambda} = \sum_{j\in\mathcal{B}} c_j^i \cdot Y_j^\lambda,\ i \in \mathcal{B},\ 1 \le \lambda \le n \cdot |\mathcal{N}'|,$$

then,

$$\begin{aligned} T_0^n &= \min\{k \le n \cdot |\mathcal{N}'| :\ \sum_{\lambda=1}^{k} \sum_{j\in\mathcal{B}} c_j^i \cdot Y_j^\lambda \ge n \cdot \sum_{j\in\mathcal{B}} c_j^i \cdot \mathcal{N}_j,\ \text{for some } i \in \mathcal{B}\} \\ &= \min\{k \le n \cdot |\mathcal{N}'| :\ \sum_{\lambda=1}^{k} \sum_{j\in\mathcal{B}} c_j^i \cdot Y_j^\lambda \ge n \cdot |\mathcal{N}'| \cdot \sum_{j\in\mathcal{B}} c_j^i \cdot \frac{\mathcal{N}_j}{|\mathcal{N}'|},\ \text{for some } i \in \mathcal{B}\} \\ &= \min\{k \le n \cdot |\mathcal{N}'| :\ \sum_{\lambda=1}^{k} X_0^{i,\lambda} \ge n \cdot |\mathcal{N}'| \cdot \bar{c}^i,\ \text{for some } i \in \mathcal{B}\} \qquad (239) \end{aligned}$$

Similarly, if $T_{\mathcal{B}}^n$ is the first time there is an optimal basis change for the ONV-2 problem, then

$$T_{\mathcal{B}}^n = \min\{k \le n \cdot |\mathcal{N}'| :\ \sum_{j\in\mathcal{B}} b_j^i \cdot (n \cdot \mathcal{N}_j - \sum_{\lambda=1}^{k} Y_j^\lambda) \le 0,\ \text{for some } i \in \{1,\dots,L\}\}$$

If we define

$$X_{\mathcal{B}}^{i,\lambda} = \sum_{j\in\mathcal{B}} b_j^i \cdot Y_j^\lambda,\ i = 1,\dots,L,\ \lambda = 0,\dots,n \cdots |\mathcal{N}'|, \qquad (240)$$

and

$$\bar{b}^i = \sum_{j\in\mathcal{B}} b_j^i \cdot \frac{\mathcal{N}_j}{|\mathcal{N}'|},\ i = 1,\dots,L, \qquad (241)$$

then, after some algebra, $T_{\mathcal{B}}^n$ can be written

$$T_{\mathcal{B}}^n = \min\{k \le n \cdot |\mathcal{N}'| :\ \sum_{\lambda=1}^{k} {X_{\mathcal{B}}}^{i,\lambda} \ge n \cdot |\mathcal{N}'| \cdot \bar{b}^i,\ \text{for some } i \in \{1,\dots,L\}\}.$$

We now have that

$$\tau^n = \min\{T_0^n, T_{\mathcal{B}}^n\}. \qquad (242)$$

Hence, τ^n is the minimum $k \in \{1,\dots,n \cdot |\mathcal{N}'|\}$ such that $\sum_{\lambda=1}^{k} X_0^{i,\lambda} \ge n \cdot |\mathcal{N}'| \cdot \bar{c}^i$ for some $i \in \mathcal{B}$, or $\sum_{\lambda=1}^{k} {X_{\mathcal{B}}}^{j,\lambda} \ge n \cdot |\mathcal{N}'| \cdot \bar{b}^j$ for some $j \in \{1,\dots,L\}$. It is evident that up to time τ^n, the processes ${X_0}^{i,\lambda}$ and ${X_{\mathcal{B}}}^{i,\lambda}$ are a sampling without replacement and from Proposition 7 given in Appendix A, we have that

$$\lim_{n\to\infty} E[n \cdot |\mathcal{N}'| - \tau^n] < \infty. \qquad (243)$$

■

Now, we have that

$$\begin{aligned} E[|G^{i,\tau^n}|] &= E[|\sum_{j\in\mathcal{B}} b_j^i \cdot (n\cdot\mathcal{N}_j - \sum_{\lambda=1}^{\tau^n} Z_j^\lambda)|] && (244)\\ &\leq E[\max_{i,j}\{b_j^i\}\cdot\sum_{j\in\mathcal{B}}(n\cdot\mathcal{N}_j - \sum_{\lambda=1}^{\tau^n} Z_j^\lambda)] && (245)\\ &= \max_{i,j}\{b_j^i\}\cdot E[n\cdot|\mathcal{N}'| - \tau^n] && (246)\end{aligned}$$

where the first equality is a result of Equation 222, and the last equality is a direct result of the definition of the random variables Z_j^k. Hence, from Markov inequality we get that

$$P(|G^{i,\tau^n}| \geq n\cdot\frac{\bar{b}}{2}) \leq \frac{2}{n\cdot\bar{b}}\cdot E[|G^{i,\tau^n}|] \qquad (247)$$

Finally from Equations 246 and 247 we get that

$$P(G^{i,\tau^n} \geq n\cdot\frac{\bar{b}}{2}) = O(n^{-1}). \qquad (248)$$

4.6.3 A closer look at the quantities $U^{i,k}$

In this section we take a closer look at the quantities $U^{i,k}$. Recall from the Definition 3 that, at the k^{th} macro-state of the ONV-2 problem, the terminal node fluid pattern is given by the vector $f^k = (f_1^k, \ldots, f_L^k)$. Let $f_{\mathcal{B}}^k = \sum_{j\in\mathcal{B}} f_j^k$. Under the assumption that the terminal nodes $j \in \mathcal{B}$ have positive remaining visitation requirements, i.e., $\sum_{\lambda=1}^k Z_j^\lambda < n\cdot\mathcal{N}_j,\ j\in\mathcal{B}$, we have that

$$E[Z_j^k] = \frac{f_j^k}{f_{\mathcal{B}}^k},\ j = 1,\ldots,L. \qquad (249)$$

Notice that the fluid volume reaching the set of terminal nodes is determined by the amount of fluid routed through each of the probability vectors $\phi^i,\ i = 1,\ldots,M$, that belong to the optimal basis of the relaxing-LP, expressed by Equations 114-116. Hence, while the optimal basis is $\mathbf{B}$, the terminal node fluid vector, f^k, is a linear combination of the basis vectors $\phi^{B(i)},\ i = 1,\ldots,m$. From Equation 167 of Observation 4, we have that $(b^i)^T \cdot \phi^{B(j)} = 0,\ i = m+1,\ldots,L,\ j = 1,\ldots,m$. Hence, we must also have $(b^i)^T \cdot f^k = 0,\ i = m+1,\ldots,L$. Therefore,

$$f^k_{j(i)} = \sum_{j \in \mathcal{B}} b^i_j \cdot f^k_j,\ i = m+1,\ldots,L. \tag{250}$$

In the light of the above equation we have that:

$$\begin{aligned} E[\sum_{j=1}^{L} b^i_j \cdot Z^k_j] &= E[\sum_{j \in \mathcal{B}} b^i_j \cdot Z^k_j - Z^k_{j(i)}] && (251) \\ &= \sum_{j \in \mathcal{B}} b^i_j \cdot \frac{f^k_j}{f_{\mathcal{B}}} - \frac{f^k_{j(i)}}{f_{\mathcal{B}}} && (252) \\ &= 0,\ i = m+1,\ldots,L. && (253) \end{aligned}$$

Remember that Equation 253 holds under the condition that (i) the optimal basis is $\mathbf{B}$, and (ii) the remaining visitation requirement for every node $j \in \mathcal{B}$ is positive. These two conditions hold for the first macro-state of the ONV-2 problem, and, furthermore, τ^n is the first time one of the above two conditions is violated. Therefore Equation 253 holds for all k's such that $0 \leq k \leq \tau^n$. From the definition of the processes $U^{i,k}$, given by Equation 221, and Equation 253, we have the following lemma:

Lemma 8 *For $i = m+1,\ldots,L$, the process $U^{i,k},\ 0 \leq k \leq \tau^n$, is a martingale.*

Since τ^n is a stopping time, then $U^{i,k\wedge\tau^n},\ 0 \leq k \leq n \cdot |\mathcal{N}'|$, is a *stopped martingale* [38]. A stopped martingale is a martingale [38] and, therefore, if we re-define $U^{i,k} = U^{i,k\wedge\tau^n},\ 0 \leq k \leq n \cdot |\mathcal{N}'|$, then $U^{i,k},\ 0 \leq k \leq n \cdot |\mathcal{N}'|$ is a martingale. In order to quantify the probability involving the processes $U^{i,k}$ in the right hand side of Equation 235, we use a martingale inequality given by the following theorem:

Theorem 12 *[12] Let U be a martingale satisfying*

1. $\mathsf{Var}(U^k|U^{k-1}) \leq \sigma^2_k$, *for* $1 \leq k \leq n$;

2. $U^k - U^{k-1} \leq \Lambda_k$, *for* $1 \leq k \leq n$.

Then, we have

$$P(U^n - U^0 \geq \lambda) \leq e^{-\frac{\lambda^2}{2\sum_{i=1}^{n}(\sigma_i^2+\Lambda_i^2)}} \tag{254}$$

In order to make use of Theorem 12 we have to establish that its conditions are satisfied by the processes $U^{i,k}$. The first condition is established by the following lemma:

Lemma 9 *There is a constant $\sigma > 0$ such that*

$$\mathsf{Var}(U^{i,k}|U^{i,k-1}) \leq \sigma^2 \tag{255}$$

for all $i = m+1, \ldots, L,\ k = 0, \ldots, n \cdot |\mathcal{N}'|$.

Proof We have that

$$\begin{aligned}\mathsf{Var}(U^{i,k}|U^{i,k-1}) &= E[(U^{i,k} - E[U^{i,k}|U^{i,k-1}])^2|U^{i,k-1}] &(256)\\ &= E[(U^{i,k} - U^{i,k-1})^2|U^{i,k-1}] &(257)\end{aligned}$$

We also have that

$$U^{i,k} - U^{i,k-1} = \begin{cases}\sum_{j=1}^{L} b_j^i \cdot Z_j^k, & k \leq \tau^n\\ 0, & k > \tau^n.\end{cases} \tag{258}$$

The quantity $\sum_{j=1}^{L} b_j^i \cdot Z_j^k$ can be alternatively written as $\sum_{j\in\mathcal{B}} b_j^i \cdot Z_j^k - Z_{j(i)}^k$ and takes values in the set $\{b_j^i - m :\ j \in \mathcal{B},\ m = 0, 1, \ldots\}$. Each such value $b_j^i - m$ is taken with probability $(\frac{f_{j(i)}^k}{f_{j(i)}^k + f_{\mathcal{B}}^k})^m \cdot \frac{f_j^k}{f_{\mathcal{B}}^k}$. Let $p^{i,k} = \frac{f_{j(i)}^k}{f_{j(i)}^k + f_{\mathcal{B}}^k}$ and $q_j^k = \frac{f_j^k}{f_{\mathcal{B}}^k}$, then

$$\begin{aligned}\mathsf{Var}(U^{i,k}|U^{i,k-1}) &= \sum_{j\in\mathcal{B}}\sum_{m=0}^{\infty}(b_j^i - m)^2 \cdot (\frac{f_{j(i)}^k}{f_{j(i)}^k + f_{\mathcal{B}}^k})^m \cdot \frac{f_j^k}{f_{\mathcal{B}}^k} &(259)\\ &= \sum_{j\in\mathcal{B}}\sum_{m=0}^{\infty}(b_j^i - m)^2 \cdot (p^{i,k})^m \cdot q_j^k &(260)\\ &= \sum_{j\in\mathcal{B}} q_j^k \cdot \sum_{m=0}^{\infty}(m^2 - 2 \cdot m \cdot b_j^i + (b_j^i)^2) \cdot (p^{i,k})^m &(261)\\ &= \sum_{j\in\mathcal{B}} q_j^k \cdot (\frac{p^{i,k}(1+p^{i,k})}{(1-p^{i,k})^3} - 2b_j^i\frac{p^{i,k}}{(1-p^{i,k})^2} + (b_j^i)^2\frac{1}{1-p^{i,k}}) &(262)\\ &\leq \sum_{j\in\mathcal{B}} \frac{p^{i,k}(1+p^{i,k}) - 2b_j^i p^{i,k}(1-p^{i,k}) + (b_j^i)^2(1-p^{i,k})^2}{(1-p^{i,k})^3} &(263)\end{aligned}$$

There is a constant $C > 0$ such that $p^{i,k}(1+p^{i,k})-2b_j^i p^{i,k}(1-p^{i,k})+(b_j^i)^2(1-p^{i,k})^2 \leq C$. Hence, Equation 263 now becomes

$$\mathsf{Var}(U^{i,k}|U^{i,k-1}) \quad \leq \quad \frac{C \cdot L}{(1-p^{i,k})^3} \tag{264}$$

It can be proved that there is an $\epsilon > 0$ such that $1 - p^{i,k} \geq \epsilon$ and, hence,

$$\mathsf{Var}(U^{i,k}|U^{i,k-1}) \leq \frac{C \cdot L}{\epsilon^3} \tag{265}$$

for all $i = m+1, \ldots, L,\ k = 0, \ldots, n \cdot |\mathcal{N}'|$. Set $\sigma^2 = \frac{C \cdot L}{\epsilon^3}$ and Lemma 9 is now proved. ▮

The second condition of Theorem 12 is implied by the fact that

$$\begin{aligned} U^{i,k} - U^{i,k-1} \quad &\leq \quad \sum_{j=1} b_j^i \cdot Z_j^k &(266)\\ &\leq \quad \max_{i,j} b_j^i. &(267) \end{aligned}$$

Hence, if we set $\Lambda = \max_{i,j} b_j^i$, Theorem 12, Lemma 9 and Equation 267 imply that for all $a > 0$,

$$P(U^{i,k} \geq \alpha \cdot n) \leq e^{-\frac{\alpha^2 \cdot n^2}{2k(\sigma^2+\Lambda^2)}} \tag{268}$$

and hence,

$$\begin{aligned} P(\max_{0 \leq k \leq \tau^n} U^{i,k} \geq \alpha \cdot n) \quad &= \quad P(\max_{0 \leq k \leq n \cdot |\mathcal{N}'|} U^{i,k} \geq \alpha \cdot n) &(269)\\ &\leq \quad \sum_{k=0}^{n \cdot |\mathcal{N}'|} P(U^{i,k} \geq \alpha \cdot n) &(270)\\ &\leq \quad \sum_{k=0}^{n \cdot |\mathcal{N}'|} e^{-\frac{\alpha^2 \cdot n^2}{2k(\sigma^2+\Lambda^2)}} &(271)\\ &\leq \quad n \cdot |\mathcal{N}'| \cdot e^{-\frac{\alpha^2 \cdot n}{2(\sigma^2+\Lambda^2)}} &(272) \end{aligned}$$

Therefore,

$$P(\max_{0 \leq k \leq \tau^n} U^{i,k} \geq \alpha \cdot n) = O(e^{-n}) \tag{273}$$

Now we are ready for the main result of this chapter.

4.6.4 Bringing everything together

From Equations 248 and 273, Equation 235 becomes

$$1 - P(E^n) = O(n^{-1}) \tag{274}$$

and, hence, in the light of the introductory discussion of section 4.6,

$$V^{\pi^{adrel}}(n \cdot \mathcal{N}) - V^{\pi^{adrel}}(n \cdot \mathcal{N}') < \infty \tag{275}$$

as $n \to \infty$. Finally,

Theorem 13 *Assume a requirement vector $\mathcal{N}$ such that the corresponding solution of the relaxing-LP is non-degenerate. Then,*

$$\lim_{n\to\infty} (V^{\pi^{adrel}}(n \cdot \mathcal{N}) - V^{*}(n \cdot \mathcal{N})) < \infty. \tag{276}$$

Proof Theorem 13 follows from Theorem 11, Equation 275 and the properties $V_{rel}(n \cdot \mathcal{N}') = V_{rel}(n \cdot \mathcal{N})$ and $V^{*}(n \cdot \mathcal{N}) \geq V_{rel}(n \cdot \mathcal{N})$. ▮

The question that naturally arises after Theorem 13, is what happens when the relaxing-LP basis is degenerate. Up to this date we do not have a clear cut result regarding this case. However, we have numerical evidence that in the case of degeneracy, the tight performance bound suggested by Theorem 13 may not hold, as we see in the following example.

Example 5 In this example we examine the relaxing-LP for the acyclic graph given in Figure 5 of Section 3.1.1. For a requirement vector $\mathcal{N}= \begin{bmatrix} \mathrm{N}_1 \\ \mathrm{N}_2 \end{bmatrix}$, the relaxing-LP is expressed by the following formulation:

$$\min\{x_1 + x_2\}$$

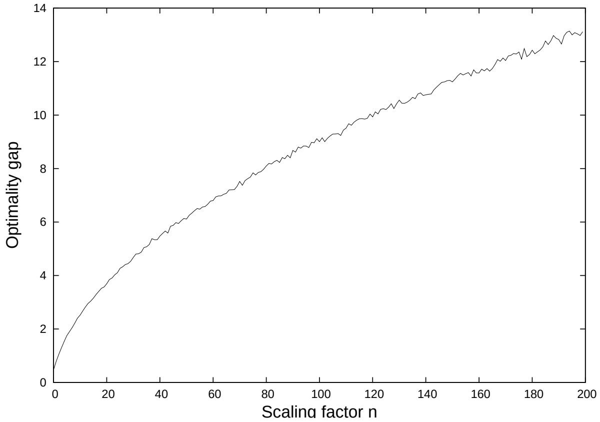

Figure 15: The optimality gap $V^{\pi^{adrel}}(n\cdot\mathcal{N})-V^*(n\cdot\mathcal{N})$ against the scaling factor $n\in\mathbb{Z}_+$, for the relaxing-LP of Example 5.

s.t.

$$x_1\phi^1+x_2\cdot\phi^2-x_3\cdot r^1-x_4\cdot r^2 \;=\; \mathcal{N} \tag{277}$$

$$x_i \;\geq\; 0,\; i=1,\ldots,4 \tag{278}$$

where

$$\phi^1{=}\begin{bmatrix}0.5\\0.5\end{bmatrix},\ \phi^2{=}\begin{bmatrix}0.3\\0.7\end{bmatrix},\ r^1{=}\begin{bmatrix}1\\0\end{bmatrix}\text{ and } r^2{=}\begin{bmatrix}0\\1\end{bmatrix}.$$

For the requirement vector $\mathcal{N}{=}\begin{bmatrix}2\\2\end{bmatrix}$, the optimal solution is $(x_1^*,x_2^*,x_3^*,x_4^*)=(4,0,0,0)$ and it is degenerate. We conducted a simulation-based evaluation of $V^{\pi^{adrel}}(n\cdot\mathcal{N})$ and we obtained the optimality gap $V^{\pi^{adrel}}(n\cdot\mathcal{N})-V^*(n\cdot\mathcal{N}),\ n\in\mathbb{Z}_+$, as illustrated in Figure 15. The optimality gap seems to be an increasing function of the scaling factor $n\in\mathbb{Z}_+$, providing evidence that policy π^{adrel} may not be $O(1)$ from the optimal in case of degeneracy. It is worthwhile to notice that the requirement vectors for which the relaxing-LP has a degenerate basis, are of the form:

$$n\cdot\begin{bmatrix}1\\1\end{bmatrix},\ n\cdot\begin{bmatrix}3\\7\end{bmatrix},\ n\in\mathbb{Z}_+.$$

or in other words, the requirement vectors that belong to the boundary of the cone defined by the vectors ϕ^1 and ϕ^2. For any other vector, the expected cost of policy π^{adrel} should be within $O(1)$ from the optimal, as it is uniformly scaled to infinity. □

Another interesting question concerns the requirement vectors $\mathcal{N}$ for which the relaxing-LP has a degenerate optimal basis. We already argued that, for a requirement vector $\mathcal{N}$ with a relaxing-LP optimal basis $\mathbf{B}$, the non-degeneracy condition implies $\mathbf{B}^{-1} \cdot \mathcal{N} > 0$. From a geometric viewpoint, notice that the inverse matrix of the optimal basis, $\mathbf{B}^{-1}$, defines a polyhedral cone

$$\mathcal{P}_{\mathbf{B}} = \{b \in Z_+^n | \ \mathbf{B}^{-1} \cdot b \geq 0\}. \tag{279}$$

Obviously, there are finitely many bases and, hence, finitely many polyhedral cones that cover the space $\mathbb{Z}_+^L$. The non-degeneracy condition for the vector $\mathcal{N}$, implies that $\mathcal{N}$ lies strictly within some cone $\mathcal{P}_{\mathbf{B}}$. For those vectors, the expected cost of policy π^{adrel} should be within $O(1)$ from the optimal, as this vector is uniformly scaled to infinity.

4.7 Discussion

In this chapter, we treated the suboptimal but computationally tractable policy π^{adrel}. Our analysis was based on an alternative characterization of the relaxing-LP and the partitioning of the state space into the state space subsets defined by a common visitation requirement vector. The undertaken approach led to a profound understanding of the dynamics governing the evolution of the ONV problem under π^{adrel}, and delivered the main result of this chapter, that for a large number of requirement vector choices the expected performance of π^{adrel} is $O(1)$ from the optimal, as the visitation requirement vector is uniformly scaled to infinity. In the next chapter, we extend our results on the ONV problem, to some new variations that are important for the effective usage of the ONV problem in the application context that motivated it.

CHAPTER V

OPTIMAL NODE VISITATION IN ACYCLIC STOCHASTIC DIGRAPHS WITH MULTI-THREADED TRAVERSALS AND INTERNAL VISITATION REQUIREMENTS

In this chapter, we extend the results of Chapter 3 to some new variations of the ONV problem, that can be obtained from the addition of the following two assumptions in the original definition: Under the first assumption, the tokens traversing the graph can "split" during certain transitions to a number of (sub-)tokens, allowing, thus, for multi-threaded graph traversals and the satisfaction of many visitation requirements during a single traversal; the resulting problem variation will be referred to as ONV-I. Under the second assumption, there are additional visitation requirements attached to the internal graph nodes, which, however, can be served only when the visitation requirements of their successors have been fully met; this new problem variation will be referred to as ONV-II. Beyond its theoretical interest, the considered extension of the ONV problem to these new variations is crucial for the effective utilization of the relevant results in the application context of Chapter 2 that motivated the ONV problem in the first place.

From an analytical standpoint, the ONV problem variations present more complicated dynamics compared to the dynamics underlying the original problem definition. More specifically, the token "splitting" effect introduced to the ONV-I variation necessitates the tracing, during a single graph traversal, of a number of tokens that can be *exponentially* large with respect to the size of the underlying graph, and thus, it adds another element of complexity to that of the original problem. Similarly, in the

ONV-II variation, an additional dimension to the problem complexity arises from the partial ordering imposed on the visitation requirements. These complications prevent the extension of the methodological framework originally developed in Chapter 3 to the new problem contexts, rendering, thus, nontrivial the extension of the results derived in that chapter to the new problem variations, and warranting a systematic re-investigation.

In the light of the above remarks, the key developments and contributions of the work presented in this chapter can be summarized as follows:

1. It is shown that the ONV-I problem variation can also be modeled as an SSP problem and that, similar to the original ONV case, its fluid relaxation can provide the basis for a suboptimal randomized policy that is computationally tractable and asymptotically optimal. Using *renewal theory* [38] arguments, we also establish bounds for the divergence of the performance of the aforementioned policy from the performance of the optimal policy, as the posed visitation requirements are scaled by a factor n that grows to infinity. Furthermore, this analysis has revealed a number of cases of considerable practical significance where the aforementioned divergence remains bounded.

2. On the other hand, the optimization problem resulting from the fluid relaxation of the ONV-II problem is of limited computational tractability. Hence, in order to obtain computationally efficient policies for this ONV variation, we confine our analysis within a class of randomized policies that are easily implementable, and we provide a fluid relaxation that leads to a policy which is asymptotically optimal within the scope of the considered policies.

The rest of the chapter is organized as follows: Section 5.1 addresses the ONV-I problem (i.e., the ONV problem with multi-threaded traversals), discussing its SSP formulation, the fluid relaxation, and an asymptotically optimal policy that can be defined on the basis of this relaxation. Subsequently, Section 5.2 introduces the analysis of the ONV-II problem within a class of randomized policies that are easily implementable, and we provide a fluid relaxation that leads to a policy which is asymptotically optimal within the scope of the considered policies.

5.1 The ONV problem with multi-threaded traversals

5.1.1 Problem description and its MDP formulation

A formal description of the ONV-I problem An instance of the problem considered in this section is completely defined by a quadruple $\mathcal{E} = (X, \mathcal{A}, \mathcal{P}, \mathcal{N})$, where

- X is a finite set of *nodes*, that is partitioned into a sequence of *"layers"*, X^0, X^1, ..., X^L. $X^0 = \{x^0\}$ defines the *source* or *root node*, while nodes $x \in X^L$ are the *terminal* or *leaf* nodes.

- $\mathcal{A}$ is a set function defined on X, that maps each $x \in X$ to the finite, non-empty set $\mathcal{A}(x)$, comprising all the *decisions / actions* that can be executed by the control agent at node x. It is further assumed that for $x \neq x'$, $\mathcal{A}(x) \cap \mathcal{A}(x') = \emptyset$.

- $\mathcal{P}$ is the *transition function*, defined on $\bigcup_{x \in X \setminus X^L} \mathcal{A}(x)$, that associates with every action a in this set a discrete probability distribution $p(\cdot; a)$. The support sets, $\mathcal{S}(a)$, of the distributions $p(\cdot; a)$ consist of *multi-sets*[1] that satisfy the following property: For any given action $a \in \mathcal{A}(x)$ with $x \in X^i$ for some $i = 0, \ldots, L-1$, the elements of $\mathcal{S}(a)$ are multi-sets defined on $\bigcup_{j=i+1}^{L} X^j$.

- $\mathcal{N}$ is the *visitation requirement vector*, that associates with each node $x \in X^L$ a visitation requirement $\mathcal{N}_x \in \mathbb{Z}_0^+$. The *support* $||\mathcal{N}||$ of $\mathcal{N}$ is defined by the nodes $x \in X^L$ with $\mathcal{N}_x > 0$; we shall refer to nodes $x \in ||\mathcal{N}||$ as the problem *"target"* nodes.

[1] We remind the reader that a multi-set defined on a set X is essentially a vector of dimensionality $|X|$ and with elements belonging to $\mathbb{Z}_0^+$.

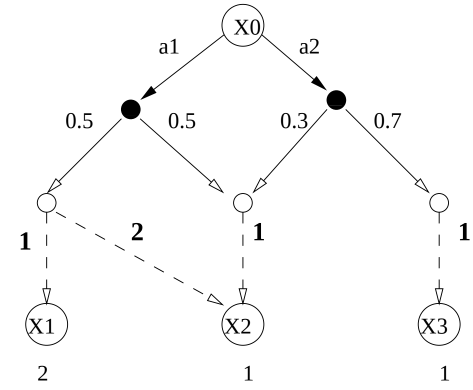

Figure 16: The stochastic graph for the problem instance considered in Example 6.

- Finally, we define the *instance size* $|\mathcal{E}| \equiv |X| + |\bigcup_{x \in X} \mathcal{A}(x)| + |\mathcal{N}|$, where application of the operator $|\ |$ on a set returns the cardinality of this set, while application on a vector returns its l_1 norm.

In the subsequent discussion we shall employ the variable vector $\mathcal{N}^c$ to denote the *vector of the remaining visitation requirements.* The control agent starts at period $t = 0$, by placing a *token* at node x^0, sets $\mathcal{N}^c := \mathcal{N}$, and at every consecutive period $t = 1, 2, 3, \ldots$, it (i) observes the current *configuration* g, i.e. the number and position of the tokens in the set $X \backslash\ X^L$, and the vector of remaining visitation requirements, $\mathcal{N}^c$, (ii) selects an action $a \in \mathcal{A}(x)$ and commands its execution on a single token at node x, (iii) generates the new tokens at the nodes indicated by the multi-set selected according to the probabilities $p(\cdot, a)$, (iv) updates $\mathcal{N}^c_x$ to $(\mathcal{N}^c_x - k)^+$ when k tokens reach one of the terminal nodes, $x \in X^L$, and finally, when the last token exits the set $X \backslash X^L$, (v) *resets* itself by placing a token at the initial node x^0, in order to start another traversal. The entire operation terminates when all the node visitation requirements have been reduced to zero. Our intention is to determine an action selection scheme – or, a *policy* – π, that maps each configuration g to an action $\pi(g) \in \bigcup_{x \in X \backslash X^L} \mathcal{A}(x)$ in a way that minimizes the expected number of graph traversals until $\mathcal{N}^c = \mathbf{0}$.

Example 6 As an example, we consider the problem instance depicted in Figure 16. In this case, there are two actions, a^1 and a^2, emanating from the root node x^0 and three leaf nodes, x^1, x^2 and x^3. The set $\mathcal{S}(a^1)$ consists of two multi-sets $\nu_{a^1,1} = [1,2,0]$ and $\nu_{a^1,2} = [0,1,0]$ whereas $\mathcal{S}(a^2)$ consists of $\nu_{a^2,1} = [0,1,0]$ and $\nu_{a^2,2} = [0,0,1]$. Furthermore, $p(\nu_{a^1,1}; a^1) = 0.5$, $p(\nu_{a^1,2}; a^1) = 0.5$, $p(\nu_{a^2,1}; a^2) = 0.3$ and $p(\nu_{a^2,2}; a^2) = 0.7$. In words, for each token emanating from x^0 through a^1, either one copy is generated at leaf node x^1 and two copies at leaf node x^2 with probability 0.5 or a single copy is generated at leaf node x^2 with probability 0.5. On the other hand, for each token emanating from x^0 through a^2, either one copy is generated at leaf node x^2 with probability 0.3 or one copy is generated at leaf node x^3 with probability 0.7. Finally we assume the requirement vector $\mathcal{N} = [2,1,1]$. □

The induced MDP problem The problem defined above can be further abstracted to a Discrete Time Markov Decision Process, $\mathcal{M} = (S, A, t, c)$, where

- S is the finite set of states, identified with tuples $(\mathcal{X}, \mathcal{N}^c)$, where (i) $\mathcal{X}$ denotes a vector of dimensionality $|X| - |X^L|$ with each component $\mathcal{X}_x$ denoting the number of tokens at node $x \in X \setminus X^L$ and (ii) $\mathcal{N}^c \in \prod_{x \in X^L} \{0, \ldots, \mathcal{N}_x\}$ denotes the remaining visitation requirements.

- A is a set function defined on S that maps each state $s \in S$ to the finite, non-empty set $A(s)$, comprising all the actions that are feasible in s. More specifically, for $s = (\mathcal{X}, \mathcal{N}^c), \mathcal{X} > \mathbf{0}$, $A(s)$ coincides with $\bigcup_{x \in X \setminus X^L : \mathcal{X}_x > 0} \mathcal{A}(x)$. Furthermore, for all states $s = (\mathcal{X}, \mathcal{N}^c)$ with $\mathcal{X} = \mathbf{0}$ and $\mathcal{N}^c \neq \mathbf{0}$, $A(s)$ consists of the single "resetting" action β.

- $t : S \times \bigcup_{s \in S} \mathcal{A}(s) \times S \longrightarrow [0,1]$ is the MDP *state transition* function, i.e., a function on all triplets (s, a, s') with $t(s, a, s')$ being the probability to reach state s' from state s on action a. More specifically, for $s = (\mathcal{X}, \mathcal{N}^c)$, $a \in A(s)$

and $s' = (\mathcal{X}', \mathcal{N}^{c'}$,

$$t(s,a,s') = \begin{cases} p(\nu_{a,i}; a), & \text{if } a \neq \beta,\ \mathcal{X}'_y = \mathcal{X}_y - 1 \geq 0,\ a \in \mathcal{A}(y), \\ & \mathcal{X}'_x = \mathcal{X}_x + \nu^x_{a,i},\ \forall x \in X/X^L \\ & \text{with } x \neq y, \text{ and } \mathcal{N}^{c'}_x = (\mathcal{N}^c_x - \nu^x_{a,i})^+, \\ & \forall x \in X^L,\ 1 \leq i \leq |\mathcal{S}(a)|; \\ 1, & \text{if } a = \beta,\ \mathcal{X} = \mathbf{0},\ \mathcal{X}' = \mathbf{1}^0; \\ 0, & \text{otherwise.} \end{cases} \tag{280}$$

In Equation 280, $\mathbf{1}^0$ denotes the unit vector with all its components equal to zero except for the one corresponding to x^0.

- $c : S \longrightarrow \{0, 1\}$ is the *cost function*, where for $s = (\mathcal{X}, \mathcal{N}^c)$,

$$c(s) = \begin{cases} 1, & \text{if } \mathcal{X} = \mathbf{0},\ \mathcal{N}^c \neq \mathbf{0}, \\ 0, & \text{if otherwise.} \end{cases} \tag{281}$$

Similar to the case of the original ONV problem, the set of states $s = (\mathcal{X}, \mathcal{N})$ with $\mathcal{N}^c = \mathbf{0}$ constitute a *closed* class which is also cost-free, i.e., once the process enters this class of states it will remain in it and there will be no more cost accumulation. We shall represent this entire class of states with a single aggregate state, s^T, which we shall refer to as the problem *terminal state*; clearly, s^T is *absorbing* and *cost-free* under any policy π. In order to ensure the reachability of s^T, from the initial state s^0, it is further assumed that for every node $x \in X^L$, with $\mathcal{N}_x > 0$, there exists at least one sequence $\xi(x) = a^{(0)}s^{(0)}a^{(1)}s^{(1)} \ldots a^{(k(x))}s^{(k(x))}$ such that (i) $a^{(0)} \in A(s^0)$ with $t(s^0, a^{(0)}, s^{(0)}) > 0$, (ii) $\forall i = 1, \ldots, k(x)$, $a^{(i)} \in A(s^{(i-1)})$ with $t(s^{(i-1)}, a^{(i)}, s^{(i)}) > 0$, and (iii) $s^{k(x)} = (\mathcal{X}, \mathcal{N}^c)$ with $\mathcal{N}^c_x < \mathcal{N}_x$; we shall refer to this sequence as an *action path* from node x^0 to node x.

In the following, we are especially interested in a policy π^*, that, starting from the *initial state* $s^0 \equiv (\mathbf{1}^0, \mathcal{N})$, will drive the underlying process to the terminal state s^T

with the minimum expected total cost. Let $V_\pi(s^0) = E_\pi[\sum_{t=0}^{\infty} c(s_t)|s_0 = s^0]$, where π is some given policy from the policy set Π, and the expectation $E_\pi[\cdot]$ is taken over all possible realizations under π. Then π^* is formally defined by

$$\pi^* = \arg\min_{\pi\in\Pi} V_\pi(s^0). \tag{282}$$

It is easy to see that under the aforestated assumptions, this problem is well defined, and therefore, according to [2]:

Theorem 14 *There exists a unique vector $V^*(s)$, $s \in S$, with $V^*(s^T) = 0$ and with its remaining components satisfying the Bellman equation*

$$V^*(s) = \min_{a\in A(s)} \{c(s) + \sum_{s'\in S} t(s,a,s') \cdot V^*(s')\} \tag{283}$$

Furthermore, the vector $V^*(s)$ defines an optimal policy π^*, by setting for all $s \in S \backslash \{s^T\}$,

$$\pi^*(s) := \arg\min_{a\in A(s)} \{c(s) + \sum_{s'\in S} t(s,a,s') \cdot V^*(s')\} \tag{284}$$

5.1.2 A computationally efficient and asymptotically optimal policy for the ONV-I problem

In Section 5.1.1, the ONV-I problem was treated by formulating it as an equivalent Dynamic Programming problem. In that formulation, the problem state was defined by the vector $(\mathcal{X}, \mathcal{N}^c)$, whose first component denotes the number and the position of the tokens in the problem-defining graph $\mathcal{G}$, and the second component the vector of the remaining visitation requirements. However, the cardinality of both, the state space and the action sets associated with the different states is an exponential function of the problem size $|\mathcal{E}|$, rendering this particular approach intractable, for most problem instances. In Chapter 6 we establish that the intractability of the Dynamic Programming approach is due to the inherent difficulty of the considered problem rather than a deficiency of the particular method. We are thus motivated to study and develop a class of efficient, yet computationally tractable, policies that we shall refer to as *asymptotically optimal*, since the ratio of their expected performance to V^* converges to unity as the node visitation requirement vector, $\mathcal{N}$, is scaled to infinity.

The "Relaxing LP" and the policy π^{rel} Next, we introduce and analyze the performance of a particular randomized policy that is obtained through a continuous – or *"fluid"* – relaxation of the original MDP problem. We shall refer to this policy as π^{rel}, and its definition relies on the optimal solution of the following LP formulation, that will be called the *"relaxing LP"*:

$$\min \sum_{a \in \mathcal{A}(x^0)} \chi_a \tag{285}$$

s.t.

$$\sum_{a \in \bigcup_{y \in X \setminus X^L} \mathcal{A}(y)} \sum_{1 \le i \le |\mathcal{S}(a)|} p(\nu_{a,i}; a) \cdot \nu^x_{a,i} \cdot \chi_a = \sum_{a \in \mathcal{A}(x)} \chi_a, \tag{286}$$

$$\forall x \in X \setminus (\{x^0\} \cup X^L) \tag{287}$$

$$\sum_{a \in \bigcup_{y \in X \setminus X^L} \mathcal{A}(y)} \sum_{1 \le i \le |\mathcal{S}(a)|} p(\nu_{a,i}; a) \cdot \nu^x_{a,i} \cdot \chi_a \ge \mathcal{N}_x, \ \forall x \in X^L \tag{288}$$

$$\chi_a \ge 0, \ \forall x \in X \setminus X^L, \ \forall a \in \mathcal{A}(x) \tag{289}$$

A natural interpretation of an optimal solution, χ^*, of the relaxing LP, is that it constitutes a *generalized* flow pattern that can satisfy the flow requirements for the terminal nodes $x \in X^L$ expressed by the visitation requirement vector, $\mathcal{N}$, while minimizing the total amount of flow induced into the graph. In particular, the generalized nature of the flow is expressed by the fact that, according to Equations 287-288, the flow leaving a node, x, is magnified by the gains defined by the multi-sets $\nu_{a,i}, \ 1 \le i \le |\mathcal{S}(a)|$.

Example 7 Consider the problem instance described in Example 6 and depicted in

Figure 16. Then the relaxing LP is expressed by the following linear program:

$$\min \;\; \chi_{a^1} + \chi_{a^2}$$

s.t.

$$\begin{aligned} 0.5 \cdot 1 \cdot \chi_{a^1} &\geq 2 \\ 0.5 \cdot 2 \cdot \chi_{a^1} + 0.5 \cdot 1 \cdot \chi_{a^1} + 0.3 \cdot 1 \cdot \chi_{a^2} &\geq 1 \\ 0.7 \cdot 1 \cdot \chi_{a^2} &\geq 1 \end{aligned}$$

$$\chi_{a^1} \geq 0, \;\; \chi_{a^2} \geq 0.$$

□

Given an optimal solution $\chi^* = \{\chi^*_a |\ a \in \bigcup_{x \in X \setminus X^L} \mathcal{A}(x)\}$ of the LP defined by Equations 285-289, policy π^{rel} assigns to a state $s = (\mathcal{X}, \mathcal{N}^c)$ with $\mathcal{X} \neq \mathbf{0}$ an action $\pi^{rel}(\mathcal{X}, \mathcal{N}^c)$ by (i) randomly picking a node $x \in X \setminus X^L$ with $\mathcal{X}_x > 0$ and (ii) executing an action $\pi^{rel}(\mathcal{X}, \mathcal{N}; x) \in \mathcal{A}(x)$ on a single token according to the probability distribution

$$\text{Prob}(\pi^{rel}(\mathcal{X}, \mathcal{N}^c; x) = a) = \frac{\chi^*_a}{\sum_{a \in \mathcal{A}(x)} \chi^*_a}, \;\; a \in \mathcal{A}(x). \tag{290}$$

For states $s = (\mathcal{X}, \mathcal{N})$, with $\mathcal{X} = \mathbf{0}$ and $\mathcal{N}^c > 0$, the policy executes with certainty the unique action $\beta \in A(s)$. Clearly the deployment of the aforestated policy π^{rel} is of polynomial complexity with respect to the problem size $|\mathcal{E}|$.

The optimal value of the relaxing LP as a lower bound to V^* Let e^{rel}_j denote the amount of flow reaching leaf node x^j when a unit amount of flow is induced into the graph and it is conveyed according to the flow pattern defined by the routing probabilities of policy π^{rel} (cf. Eq. 290). Then e^{rel}_j is equal to the expected number of tokens reaching node x^j during a single graph traversal under policy π^{rel}, and we have the following theorem:

Theorem 15 *Given an ONV-I problem instance $\mathcal{E} = (X, \mathcal{A}, \mathcal{P}, \mathcal{N})$, let V^*_{rel} and χ^* respectively denote the optimal value and an optimal solution of the relaxing LP. Also, let e^{rel}_j, $x^j \in X^L$, be defined on the basis of χ^* as indicated in the previous paragraph. Then,*

$$V^*_{rel} = \max_{j:\mathcal{N}_j>0}\{\frac{\mathcal{N}_j}{e^{rel}_j}\} \leq V^* \tag{291}$$

The proof of this theorem is similar to the proof of Theorem 6 in Chapter 3, and it is omitted.

Establishing the asymptotic optimality of π^{rel} Next we proceed to prove the asymptotic optimality of π^{rel}. For this, consider the problem sequence, $\{\mathcal{E}(n)\}$, that is induced by a problem instance $\mathcal{E} = (X, \mathcal{A}, \mathcal{P}, \mathcal{N})$ through the scaling of the visitation requirement vector, $\mathcal{N}$, by a factor $n \in \mathbb{Z}^+$. Also, in the following, we shall let $\{V^*_{rel}(n)\}$ denote the sequence of the optimal objective values of the relaxing LP implied by the problem sequence $\{\mathcal{E}(n)\}$, and $\{V^*(n)\}$ denote the sequence of the corresponding optimal expected total costs. Finally, we define $\{V^{\pi^{rel}}(n)\}$ as the sequence of the expected costs incurred by the application of the randomized policy π^{rel} to the problem instances $\mathcal{E}(n)$. Before we proceed, we present a technical lemma that is necessary in the subsequent derivations.

Lemma 10 *Let $X_1, X_2, \ldots$ be i.i.d. random variables such that $0 \leq X \leq K, a.s.$ and $\mu = E[X_1]$. Set $S_0 = 0$, $S_k = \sum_{i=1}^k X_i$ and define $\psi_n = \max\{k : S_k \leq n \cdot c\}$. Then*

$$\{n^{-r/2}(\psi_n - \frac{n \cdot c}{\mu})^r, n \geq 1\} \tag{292}$$

is uniformly integrable for every $r \geq 1$.

Proof: Let $\psi'_n = \min\{k : S_k > n \cdot c\}$. Then ψ'_n is a stopping time and, from Lemma 2.3 of [21], we have that

$$E[(\sum_{i=1}^{\psi'_n}(X_i - \mu))^r] \quad \leq \quad C(r, E[X^r]) \cdot E[(\psi'_n)^{r/2}] \tag{293}$$

where $C(r, E[X^r])$ is a constant depending only on r and $E[X^r]$. Equation 293 further implies that

$$E[n^{-r/2} \cdot (\sum_{i=1}^{\psi'_n}(X_i - \mu))^r] \quad \leq \quad C(r, E[X^r]) \cdot E[(\frac{\psi'_n}{n})^{r/2}] \tag{294}$$

From Equation 294 and Theorem 2.3 of [21], we get

$$\sup_{n\geq 1} E[n^{-r/2} \cdot (\sum_{i=1}^{\psi'_n}(X_i - \mu))^r] < \infty \tag{295}$$

which implies the uniform integrability of $\{n^{-r/2} \cdot (\sum_{i=1}^{\psi'_n}(X_i - \mu))^r, n \geq 1\}$ [9].

By the definition of the renewal process ψ'_n,

$$n^{-1/2} \cdot \sum_{i=1}^{\psi'_n}(X_i - \mu) - n^{-1/2} \cdot K \leq n^{-1/2} \cdot (n \cdot c - \mu \cdot \psi'_n) \leq n^{-1/2} \cdot \sum_{i=1}^{\psi'_n}(X_i - \mu) + n^{-1/2} \cdot K \tag{296}$$

which implies

$$|n^{-1/2} \cdot (n \cdot c - \mu \cdot \psi'_n)| \leq |n^{-1/2} \cdot \sum_{i=1}^{\psi'_n}(X_i - \mu)| + n^{-1/2} \cdot K \tag{297}$$

and based on the inequality $(a + b)^r \leq 2^{r-1} \cdot (|a|^r + |b|^r),\ a, b \in R$, we also get,

$$|n^{-1/2}(n \cdot c - \mu \cdot \psi'_n)|^r \leq 2^{r-1} \cdot (|n^{-1/2} \sum_{i=1}^{\psi'_n}(X_i - \mu)|^r + n^{-r/2} \cdot K^r) \tag{298}$$

Hence, the uniform integrability of $\{n^{-r/2} \cdot (\sum_{i=1}^{\psi'_n}(X_i - \mu))^r, n \geq 1\}$ and Equation 298 imply the uniform integrability of $\{n^{-r/2} \cdot (n \cdot c - \mu \cdot \psi'_n)^r, n \geq 1\}$. Since $\psi'_n = \psi_n + 1$ we have that,

$$n^{-1/2} \cdot (n \cdot c - \mu \cdot \psi_n) = n^{-1/2} \cdot (n \cdot c - \mu \cdot \psi'_n) + n^{-1/2} \cdot \mu \tag{299}$$

which gives

$$n^{-r/2} \cdot |n \cdot c - \mu \cdot \psi_n|^r \leq 2^{r-1} \cdot (n^{-r/2} \cdot |n \cdot c - \mu \cdot \psi'_n|^r + n^{-r/2} \cdot \mu^r) \tag{300}$$

and implies the uniform integrability of $\{n^{-r/2} \cdot (n \cdot c - \mu \cdot \psi_n)^r,\ n \geq 1\}$. □

Then, we have the following theorem:

Theorem 16 *Given an ONV-I problem instance $\mathcal{E} = (X, \mathcal{A}, \mathcal{P}, \mathcal{N})$, consider the problem sequence $\mathcal{E}(n)$ that is obtained through the uniform scaling of the visitation requirement vector $\mathcal{N}$ by a factor $n \in \mathbb{Z}^+$. Then, as $n \rightarrow \infty$,*[2]

$$V^{\pi^{rel}}(n) - V^*_{rel}(n) = O(\sqrt{n}) \tag{301}$$

Furthermore, if there exists a target leaf node x^k such that, for any other target leaf node x^j, $\frac{\mathcal{N}_k}{e^{rel}_k} > \max_{j \neq k}\{\frac{\mathcal{N}_j}{e^{rel}_j}\}$, then, as $n \rightarrow \infty$,

$$V^{\pi^{rel}}(n) - V^*_{rel}(n) = O(1). \tag{302}$$

Proof: Let X^i_j denote the random number of tokens ending at leaf node x^j during the i^{th} graph traversal under π^{rel} and $\sigma^2_j = Var(X^i_j)$. Also, let $\{\psi^n_j,\ n \geq 0\}$ be a *renewal process* [38] associated with the sequence $\{X^i_j\}$ defined as

$$\psi^n_j = \max\{k :\ \sum_{i=1}^{k} X^i_j \leq n \cdot \mathcal{N}_j\}, \tag{303}$$

with $\psi^n_j = 0$ if $X^1_j > n \cdot \mathcal{N}_j,\ j : \mathcal{N}_j > 0$. Then the performance of the policy π^{rel} satisfies

$$V^{\pi^{rel}}(n) \leq E[\max_{j:\mathcal{N}_j>0}\{1 + \psi^n_j\}]. \tag{304}$$

Hence,

$$\begin{aligned} V^{\pi^{rel}}(n) - V^*_{rel}(n) &\leq 1 + E[\max_{j:\mathcal{N}_j>0}\{\psi^n_j\}] - \max_{j:\mathcal{N}_j>0}\{\frac{n\mathcal{N}_j}{e^{rel}_j}\} & (305)\\ &\leq 1 + E[\max_{j:\mathcal{N}_j>0}\{|\psi^n_j - \frac{n\mathcal{N}_j}{e^{rel}_j}|\}] & (306)\\ &\leq 1 + \sum_{j:\mathcal{N}_j>0} E[|\psi^n_j - \frac{n\mathcal{N}_j}{e^{rel}_j}|] & (307) \end{aligned}$$

[2] *We remind the reader that $f(n) = O(g(n)) \Rightarrow \exists c, n_0$ s.t. $0 \leq f(n) \leq c \cdot g(n),\ \forall n \geq n_0$.*

where the first inequality is the result of Equation 304 and Theorem 15, and the second inequality is the result of the following property:

$$\begin{aligned} & \forall a_i, b_i \in \mathbb{R}, i = 1, \ldots, n, \\ & |\max\{a_1, a_2, \ldots, a_n\} - \max\{b_1, b_2, \ldots, b_n\}| && (308) \\ \leq\ & \max\{|a_1 - b_1|, |a_2 - b_2|, \ldots, |a_n - b_n|\}. && (309) \end{aligned}$$

Also , from the *renewal central limit theorem* [38] we get that

$$\frac{1}{\sqrt{n}} \cdot (\psi_j^n - \frac{n\mathcal{N}_j}{e_j^{rel}}) \Rightarrow N(0, \frac{\sigma_j^2 \cdot \mathcal{N}_j}{(e_j^{rel})^3}),\ j : \mathcal{N}_j > 0. \qquad (310)$$

as $n \to \infty$. But then, Equation 310, when combined with Lemma 10 and the Continuous Mapping Theorem, imply that

$$\frac{1}{\sqrt{n}} E[|\psi_j^n - \frac{n\mathcal{N}_j}{e_j^{rel}}|] \longrightarrow E[|N(0, \frac{\sigma_j^2 \cdot \mathcal{N}_j}{(e_j^{rel})^3})|],\ j : \mathcal{N}_j > 0. \qquad (311)$$

as $n \to \infty$. Equation 301 now follows from Equation 307 when combined with Equation 311.

To prove Equation 302 we proceed as follows: Assume that $\max_{j:\mathcal{N}_j>0}\{\frac{n\mathcal{N}_j}{e_j^{rel}}\} = \frac{n\cdot\mathcal{N}_1}{e_1^{rel}}$; then,

$$\begin{aligned} V^{\pi^{rel}}(n) - V_{rel}^*(n) &\leq 1 + E[\max_{j:\mathcal{N}_j>0}\{\psi_j^n\}] - \max_{j:\mathcal{N}_j>0}\{\frac{n\mathcal{N}_j}{e_j^{rel}}\} && (312) \\ &= 1 + E[\max_{j:\mathcal{N}_j>0}\{\psi_j^n\}] - E[\psi_1^n] + E[\psi_1^n] - \frac{n\mathcal{N}_1}{e_1^{rel}} && (313) \\ &= 1 + E[\max_{j:\mathcal{N}_j>0}\{\psi_j^n - \psi_1^n\}] + E[\psi_1^n] - \frac{n\mathcal{N}_1}{e_1^{rel}} && (314) \\ &\leq 1 + \sum_{j\neq 1:\mathcal{N}_j>0} E[(\psi_j^n - \psi_1^n)^+] + E[\psi_1^n] - \frac{n\mathcal{N}_1}{e_1^{rel}} && (315) \end{aligned}$$

Since, for every $n \geq 1$, $\psi_j^n + 1$ is a *stopping time* with respect to $\{X_j^i\}$, with $E[\psi_j^n] < \infty$, we can write [38]

$$\begin{aligned} E[\sum_{i=1}^{\psi_j^n+1} X_j^i] &= E[\psi_j^n + 1]E[X_j^1] && (316) \\ &= e_j^{rel} \cdot (E[\psi_j^n] + 1) && (317) \end{aligned}$$

Let K denote the maximum number of tokens that can be generated during a single graph traversal. Then, by definition of $\psi_j^n + 1$,

$$n \cdot \mathcal{N}_j \leq \sum_{i=1}^{\psi_j^n + 1} X_j^i \leq n \cdot \mathcal{N}_j + K, \; j : \mathcal{N}_j > 0. \tag{318}$$

Equations 317 and 318 imply that

$$0 \leq E[\psi_j^n] + 1 - \frac{n \cdot \mathcal{N}_j}{e_j^{rel}} \leq \frac{K}{e_j^{rel}} \tag{319}$$

Next, we prove that

$$E[(\psi_j^n - \psi_1^n)^+] \rightarrow 0, \; \forall j : \mathcal{N}_j > 0 \tag{320}$$

as $n \rightarrow \infty$. Indeed, for $r \geq 1$, $a_j^n = \frac{1}{\sqrt{n}}(\psi_j^n - \frac{n \cdot \mathcal{N}_j}{e_j^{rel}})$ and $c_j = \frac{\mathcal{N}_1}{e_1^{rel}} - \frac{\mathcal{N}_j}{e_j^{rel}} > 0$, we have that

$$\begin{aligned}
& E[(\psi_j^n - \psi_1^n)^+] && (321)\\
= \;& E[(\psi_j^n - \psi_1^n) \cdot I(\psi_j^n \geq \psi_1^n)] && (322)\\
\leq \;& E[\psi_j^n \cdot I(\psi_j^n \geq \psi_1^n)] && (323)\\
\leq \;& \sqrt{E[(\psi_j^n)^2] \cdot P(\psi_j^n \geq \psi_1^n)} && (324)\\
= \;& \sqrt{E[(\psi_j^n)^2] \cdot P((\psi_j^n - \frac{n \cdot \mathcal{N}_j}{e_j^{rel}}) - (\psi_1^n - \frac{n \cdot \mathcal{N}_1}{e_1^{rel}}) \geq \frac{n \cdot \mathcal{N}_1}{e_1^{rel}} - \frac{n \cdot \mathcal{N}_j}{e_j^{rel}})} && (325)\\
= \;& \sqrt{E[(\psi_j^n)^2] \cdot P(a_j^n - a_1^n \geq \sqrt{n} \cdot c_j)} && (326)\\
\leq \;& \sqrt{E[(\psi_j^n)^2] \cdot \frac{1}{c_j^r \cdot n^{r/2}} \cdot E[(a_j^n - a_1^n)^r]} && (327)\\
\leq \;& \sqrt{E[(\psi_j^n)^2] \cdot \frac{2^{r-1}}{c_j^r \cdot n^{r/2}} \cdot E[|a_j^n|^r + |a_1^n|^r]} && (328)
\end{aligned}$$

where the second inequality is an application of Schwarz inequality, the third inequality an application of Markov inequality, and the last inequality is a direct consequence of $(a+b)^r \leq 2^{r-1} \cdot (|a|^r + |b|^r), \; a, b \in R$. Furthermore, from Theorem 2.3 of [21] we have that

$$E[(\psi_j^n)^2] = O(n^2) \tag{329}$$

and if we choose r such that $\frac{r}{2} > 2$, then Equations 311, 328, 329 and Lemma 10 imply Equation 320.

Finally, Equation 302 follows immediately from Equation 315 when combined with Equations 319 and 320. □

An immediate implication of Theorem 16 is the asymptotic optimality of policy π^{rel}, formally stated and proven in the following corollary:

Corollary 3 *Given an ONV-I problem instance $\mathcal{E} = (X, \mathcal{A}, \mathcal{P}, \mathcal{N})$, consider the problem sequence $\mathcal{E}(n)$ that is obtained through the uniform scaling of the visitation requirement vector $\mathcal{N}$ by a factor $n \in \mathbb{Z}^+$. Then, as $n \to \infty$,*

$$\frac{V^{\pi^{rel}}(n)}{V^*(n)} \longrightarrow 1 \tag{330}$$

Proof: The combination of Theorems 16 and 15 implies that $\lim_{n\to\infty} \frac{V^{\pi^{rel}}(n)}{V^*(n)} \leq 1$, while the definition of V^* implies that $V^{\pi^{rel}}(n) \geq V^*(n)$, $\forall n \in \mathbb{Z}^+$. □

We conclude this section by noticing that the results of Theorem 16 and their derivation imply that, under the condition that there exists a k such that $\frac{\mathcal{N}_k}{e_k^{rel}} > \max_{j\neq k}\{\frac{\mathcal{N}_j}{e_j^{rel}}\}$, the performance of π^{rel} and π^* will differ from the lower bound $V^{rel}(n)$ by at most $\frac{K}{e_k^{rel}}$, as the scaling factor n grows to infinity. An intuitive interpretation of this result can be obtained by considering the ratio $\frac{\mathcal{N}_j}{e_j^{rel}}$ to be a "measure of difficulty" of the visitation requirement of the j^{th} leaf node. As n grows to infinity, the differences $\frac{n\cdot\mathcal{N}_k}{e_k^{rel}} - \frac{n\cdot\mathcal{N}_j}{e_j^{rel}}$ are also growing, hence the solution of the relaxing LP contains enough information in order to bias the system behavior towards the optimal solution. On the other hand, when the number of leaf nodes corresponding to the maximal ratio $\frac{n\cdot\mathcal{N}_k}{e_k^{rel}}$ are more than one, π^{rel} will treat those nodes as equally difficult targets. Furthermore, the *static* nature of this policy will not allow it to exploit the dynamics of the future problem states, where the original ties will have been resolved. This last observation motivates the consideration of *adaptive* implementations of π^{rel}, where the routing probabilities that define the new policy are revised at every change of the vector $\mathcal{N}^c$. The specification details of these policies are similar to those discussed in Chapter 4 with respect to the the original ONV problem, and they are omitted.

5.2 Adding the Internal Visitation Requirements

The new ONV problem version In this section we consider the extension of the ONV problem, that is obtained by the introduction of visitation requirements for the internal nodes of the stochastic graph that underlies the problem definition. An instance of this new ONV problem is defined again by a quadruple $\mathcal{E} = (X, \mathcal{A}, \mathcal{P}, \mathcal{N})$, where all the components remain the same as in the case of Section 5.1, except for the visitation requirement vector $\mathcal{N}$, which now is defined as follows:

- $\mathcal{N}$ associates with each node $x \in X$ a visitation requirement $\mathcal{N}_x \in \mathbb{Z}_0^+$. The support $||\mathcal{N}||$ of $\mathcal{N}$ is defined by the nodes $x \in X$ with $\mathcal{N}_x > 0$. Furthermore, it is implicitly assumed that the visitation requirements of a node $x \in X$ will start to be satisfied only after the complete satisfaction of the visitation requirements of all its successor nodes.

The new problem described above can be further abstracted to an MDP, $\mathcal{M} = (S, A, t, c)$, where all the components remain the same as in the MDP definition of the ONV problem addressed in Section 5.1, except for the remaining visitation requirement vector $\mathcal{N}^c$ and its updating through the transition function t. More specifically, in this new problem context, $\mathcal{N}^c$ is an $|X|$-dimensional vector initialized at $\mathcal{N}$. Furthermore, given a state $s = (\mathcal{X}, \mathcal{N}^c) \in S$ with $\mathcal{X}_y > 0$, and a decision $a \in A(y)$, we compute the state $s' = (\mathcal{X}', \mathcal{N}^{c'})$, that results from the execution of a in s through its outcome defined by the multi-set $\nu_{a,i}$, according to the following procedure:

1. $\mathcal{X}'_y := \mathcal{X}_y - 1$;

2. $\forall x \in X \backslash X^L, \;\; \mathcal{X}'_x := \mathcal{X}_x + \nu^x_{a,i}$;

3. $\forall l = L, L-1, \ldots, 0, \; \forall x \in X^l$,

   `if` $\sum_{q \in Succ(x)} \mathcal{N}^c_q = 0$ `then` $\mathcal{N}^{c'}_x := (\mathcal{N}^c_x - \nu^x_{a,i})^+$ `else` $\mathcal{N}^{c'}_x := \mathcal{N}^c_x$;

The notation $Succ(x)$ appearing in the above specification denotes the *immediate* successors of node x in the problem-defining graph $\mathcal{G}$.[3] For states $s = (\mathcal{X}, \mathcal{N}^c) \in S$ with $\mathcal{X} = 0$, the process "resets" itself in the spirit expressed by Equation 280 in Section 5.1. Finally, defining the cost function $c(s)$ and the terminal state s^T as discussed in Section 5.1, and expressing the problem objective by

$$\pi^* = \arg\min_{\pi \in \Pi} E_\pi[\sum_{t=0}^{\infty} c(s_t) | s_0 = s^0] \tag{331}$$

we obtain a well-defined SSP problem whose optimal solution is characterized by Theorem 14. In the following, we shall use the notation $V^*(s)$ and $\pi^*(s)$, $s \in S \backslash \{s^T\}$, in order to characterize the optimal value function and an optimal policy for this SSP.

Problem restriction An exact fluid relaxation of the ONV problem with internal visitation requirements is provided in Appendix B, but its practical value is rather limited. In the following, we constrain the solution of the considered ONV problem over the class of *static* randomized policies, which are simpler in their characterization and evaluation, and more easily implementable. Hence, let Π^S denote the class of static randomized policies and V^*_S denote the optimal value of the considered problem when resticted in the policy space Π^S. Then, in a spirit similar to that adopted in Section 5.1, we define a fluid relaxation and an induced randomized policy for the ONV variation considered in this section, and we show that the proposed fluid

[3] Obviously, for nodes $x \in X^L$, $Succ(x) = \emptyset$ and the condition in the "if" statement of item (3) is immediately satisfied.

relaxation provides a lower bound for V_S^*, while the induced randomized policy is asymptotically optimal for the problem restriction in the policy space Π^S.

A computationally efficient and asymptotically optimal policy for the restricted problem The problem relaxation employed in the subsequent analysis is described by the following mathematical programming (MP) formulation:

$$\min V_{x^0} \tag{332}$$

s.t.

$$\sum_{a \in \mathcal{A}(x^0)} \chi_a = 1 \tag{333}$$

$$\forall x \in X \backslash \ (\{x^0\} \cup X^L),$$

$$\sum_{a \in \bigcup_{y \in X \backslash X^L} \mathcal{A}(y)} \sum_{1 \leq i \leq |\mathcal{S}(a)|} p(\nu_{a,i}; a) \nu_{a,i}^x \chi_a = \sum_{a \in \mathcal{A}(x)} \chi_a \tag{334}$$

$$e_{x^0}^{rel} = 1 \tag{335}$$

$$\forall x \in X \backslash \{x^0\},$$

$$e_x^{rel} = \sum_{a \in \bigcup_{y \in X \backslash X^L} \mathcal{A}(y)} \sum_{1 \leq i \leq |\mathcal{S}(a)|} p(\nu_{a,i}; a) \nu_{a,i}^x \chi_a \tag{336}$$

$$\forall x \in X \backslash \{x^0\} \text{ with } \mathcal{N}_x > 0, \quad e_x^{rel} > 0 \tag{337}$$

$$\forall x \in X^L, \ V_x = \frac{\mathcal{N}_x}{e_x^{rel}} \tag{338}$$

$$\forall x \in X \backslash X^L, \ V_x = \max_{y \in Succ(x)} \{V_y\} + \frac{\mathcal{N}_x}{e_x^{rel}} \tag{339}$$

$$\forall x \in X \backslash X^L, \ \forall a \in \mathcal{A}(x), \quad \chi_a \geq 0 \tag{340}$$

Variables χ_a in the above formulation denote the (generalized) flow routed through the arcs corresponding to the different actions $a \in \mathcal{A}$, for each unit of flow induced to the problem-defining graph $\mathcal{G}$ through its root node x^0 (c.f., Constraints 333, 334).

In a similar spirit, variables e_x^{rel} denote the amount of (generalized) flow reaching each node $x \in X$, for each unit of flow induced in $\mathcal{G}$ through node x^0 (c.f., Constraints 335, 336). Furthermore, Constraint 337 requests that any feasible solution of this formulation routes a positive amount of flow to every node x with non-zero visitation requirements. Finally, variables V_x denote the minimum amount of flow required in order to satisfy the corresponding node visitation requirements, under the routing scheme described by variables χ_a, e_x^{rel}, and the precedence constraints expressed by the underlying graph $\mathcal{G}$ (c.f., Constraints 338, 339). From a practical computational standpoint, the solution of the above formulation can be further facilitated by replacing Constraint 339 with the following constraint:

$$\forall x \in X \backslash X^L,\ \forall y \in Succ(x), \quad V_x \geq V_y + \frac{\mathcal{N}_x}{e_x^{rel}} \tag{341}$$

The resulting formulation is convex, and it can be easily addressed through standard techniques borrowed from convex optimization [3].

Given an optimal flow, χ^*, for the MP formulation defined by Equations 332-340, the definition and execution of the proposed randomized policy follows exactly the guidelines described in Section 5.1 for the definition of the policy π^{rel} from the fluid relaxation of the ONV problem addressed in that section. To emphasize this affinity between the two policies, we shall keep referring to the new policy defined in this section as the policy π^{rel}, while the MP formulation of Equations 332-340 will be called the *relaxing MP*. The following theorem is the counterpart of Theorem 15 for this new problem context:

Theorem 17 *Given an instance $\mathcal{E} = (X, \mathcal{A}, \mathcal{P}, \mathcal{N})$ of the ONV problem with internal visitation requirements, let V_{rel}^* and (χ^*, e^{rel^*}, V^*) respectively denote the optimal value and an optimal solution of the corresponding relaxing MP. Then,*

$$V_{rel}^* = V_{x^0}^* \leq V_S^* \tag{342}$$

where V_S^* denotes the optimal solution of the considered problem instance when restricted to the space of static randomized policies.

The proof of Theorem 17 can be obtained through a technique similar to that applied in the proof of Theorem 6, and it is omitted. The next theorem and the accompanying corollary establish the asymptotic optimality of π^{rel} in the considered problem context.

Theorem 18 *Given an instance $\mathcal{E} = (X, \mathcal{A}, \mathcal{P}, \mathcal{N})$ of the ONV problem with internal visitation requirements, consider the problem sequence, $\mathcal{E}(n)$, that is obtained through the uniform scaling of the visitation requirement vector $\mathcal{N}$ by a factor $n \in \mathbb{Z}^+$. Then, as $n \to \infty$,*

$$V^{\pi^{rel}}(n) - V^*_{rel}(n) = O(\sqrt{n}) \tag{343}$$

Proof: Let X_x^i denote the random number of tokens traversing node $x \in X$ during the i^{th} graph traversal under π^{rel} and $\sigma_x^2 = Var(X_x^i)$. Also, let $\{\psi_x^n, n \geq 1\}$ be the renewal process associated with the sequence $\{X_x^i\}$, defined as

$$\psi_x^n = \max\{k : \sum_{i=1}^{k} X_x^i \leq n \cdot \mathcal{N}_x\} \tag{344}$$

with $\psi_x^n = 0$ if $X_x^1 > n \cdot \mathcal{N}_x,\ x : \mathcal{N}_x > 0$. Finally, define

$$\Psi_x^n = \psi_x^n + 1,\ x \in X^L \tag{345}$$

$$\Psi_x^n = \max_{y \in Succ(x)} \{\Psi_y^n\} + \psi_x^n + 1,\ x \in X \backslash X^L \tag{346}$$

Then, the performance of policy π^{rel} satisfies

$$V^{\pi^{rel}}(n) \leq E[\Psi_{x^0}^n] \tag{347}$$

Equation 347, when combined with Theorem 17, imply that, in order to prove the result of Theorem 18, it suffices to show that

$$\forall x \in X, \quad E[|\Psi_x^n - V_x^*(n)|] = O(\sqrt{n}) \tag{348}$$

where $V_x^*(n)$ denotes the optimal value of variable V_x in the relaxing MP formulated for problem instance $\mathcal{E}(n)$.

We proceed to prove this result through an induction on the number of graph layers, l. The base case, for $l = L$, is immediately obtained from the results in the proof of Theorem 16. Next, we consider an l such that $0 \leq l < L$, and assume that Equation 348 holds for all $x \in \bigcup_{l+1 \leq i \leq L} X^i$. Then, for $x \in X^l$, we have that

$$\begin{aligned} E[|\Psi_x^n - V_x^*(n)|] &= \\ E[|\max_{y \in Succ(x)} \{\Psi_y^n\} + \psi_x^n + 1 - \max_{y \in Succ(x)} \{V_y^*(n)\} - \frac{n \cdot \mathcal{N}_x}{e_x}|] &\leq \\ E[\max_{y \in Succ(x)} \{|\Psi_y^n - V_y^*(n)|\}] + E[|\psi_x^n + 1 - \frac{n \cdot \mathcal{N}_x}{e_x}|] &\leq \\ \sum_{y \in Succ(x)} E[|\Psi_y^n - V_y^*(n)|] + E[|\psi_x^n + 1 - \frac{n \cdot \mathcal{N}_x}{e_x}|] & \end{aligned} \tag{349}$$

Each term of the summation appearing in Equation 349 is $O(\sqrt{n})$ from the induction hypothesis, while the fact that

$$E[|\psi_x^n + 1 - \frac{n \cdot \mathcal{N}_x}{e_x}|] = O(\sqrt{n}) \tag{350}$$

follows immediately from the results in the proof of Theorem 16. But then, the whole quantity appearing in Equation 349 is $O(\sqrt{n})$, establishing the result of Equation 348, and, through that, the result of the Theorem. □

The next corollary derives from Theorems 17 and 18, and it is the counterpart of Corollary 3 for the restricted ONV problem variation considered in this paragraph.

Corollary 4 *Consider an instance $\mathcal{E} = (X, \mathcal{A}, \mathcal{P}, \mathcal{N})$ of the ONV problem with internal visitation requirements, restricted in the space of static randomized policies Π_S. Also, consider the problem sequence $\mathcal{E}(n)$ that is obtained through the uniform scaling of the visitation requirement vector $\mathcal{N}$ by a factor $n \in \mathbb{Z}^+$. Then, as $n \to \infty$,*

$$\frac{V^{\pi^{rel}}(n)}{V_S^*(n)} \longrightarrow 1 \tag{351}$$

5.3 Discussion

The work presented in this chapter, extended the previous results on the ONV problem to some new problem variations, that are important for the effective usage of the ONV problem in the application that motivated it. Furthermore, by using techniques similar to those presented in this chapter, our current results on the ONV problem have been extended even to cases where the problem-defining digraph possesses cyclical structure. The extension is non-trivial, and the relevant results can be found in [10]. In the next chapter, we take a more systematic look at the computational complexity of the various ONV variations introduced in this work. Our results will position these variations in the problem hierarchy established by the theory of computational complexity.

CHAPTER VI

THE COMPUTATIONAL COMPLEXITY OF THE ONV PROBLEM VARIATIONS

This chapter takes a systematic look at the computational complexity of the ONV problems considered in Chapters 3 and 5. A key result offered in this direction is that the introduction of the token splitting effect renders the ONV-I problem PSPACE-hard [32]. On the other hand, we have not been able to provide a clear-cut complexity characterization neither for the original ONV problem nor for ONV-II. But as an intermediary step to the complexity analysis of these two last cases, we provide an additional result that establishes that the ONV-II problem is at least as difficult as the *"Poisson-tree" scheduling* problem, a well known and, to the best of our knowledge, still open problem in the relevant literature on computational complexity [31, 32]. Beyond assisting with positioning the ONV and ONV-II problems in the relevant hierarchy of the computational complexity theory, this last result also reveals a connection between the ONV problem(s) and the problems addressed by the more classical scheduling theory.

6.1 The computational complexity of the ONV-I problem

In this section we address the question of the computational complexity of the ONV-I problem by studying its *decision version*, which corresponds to a relevant "*yes*" or "*no*" problem. In particular, this decision problem addresses the following question: Given a problem instance $\mathcal{E}$ described by the MDP $\mathcal{M} = (S, A, t, c)$ and an integer K, is there a policy π such that $V_\pi(s^0) < K$?

For a general introduction to the theory of Computational Complexity we refer to [32]. The primary tool for classifying computational problems has been the notion of *reduction.* Given two decision problems A and B, we say that A *polynomially reduces* to B, if there is a mapping f from the instances of size n of A to the instances of size at most $p(n)$ of B, such that, (i) $p(n)$ is a polynomial depending on A and B, (ii) the mapping f can be computed in polynomial time, and (iii) an instance x of A is a "yes" instance if and only if $f(x)$ is a "yes" instance of B. Let A be a problem, and $\mathcal{C}$ be a class of problems, such as P, NP, PSPACE. We say that A is *$\mathcal{C}$-complete* if (i) A is a member of $\mathcal{C}$, and (ii) every member B of $\mathcal{C}$ is reducible to A. We say that problem B is *$\mathcal{C}$-hard* if there is a $\mathcal{C}$-complete problem A such that A is reducible to B.

The complexity class relevant to the ONV-I problem, is the class of problems solvable in polynomial space, PSPACE. The best known PSPACE-complete problem is that of *quantified satisfiability (QSAT).* A *quantified boolean formula* consists of a series of existential and universal quantifiers and a boolean formula ϕ in conjunctive normal form. In the QSAT problem we are given a quantified boolean formula with alternating quantifiers, $\exists x_1 \forall x_2 \exists x_3 \dots \forall x_n \phi(x_1, \dots, x_n)$ and we seek to determine whether this formula is *satisfiable*, that is, whether there is a truth value for x_1 such that for all truth values of x_2, etc. there is a truth value of x_n, such that ϕ comes out true. Next, we use QSAT in order to establish the following theorem:

Theorem 19 *The decision version of the ONV-I problem is PSPACE-hard.*

Proof: As mentioned above, to show PSPACE-hardness, we reduce QSAT to the considered problem. For any quantified formula ϕ with n variables and m clauses, we construct an ONV-I problem instance, $\mathcal{E}(X, \mathcal{A}, \mathcal{P}, \mathcal{N}; \phi)$, that involves an acyclic graph with n decision and $m+1$ terminal nodes, and its optimal policy has a cost of 1 if and only if the original QSAT problem is satisfiable.

We now proceed into the details of the construction (cf. Figure 17 for a concrete example). The acyclic graph consists of n decision nodes, partitioned in n consecutive

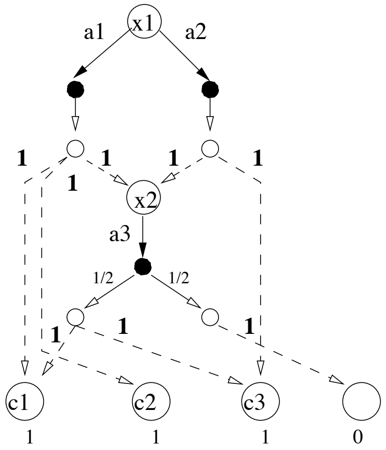

Figure 17: The acyclic graph corresponding to the boolean formula ϕ with two variables x_1, x_2 and three clauses $c_1 = x_1 \vee x_2, c_2 = x_1$ and $c_3 = \overline{x}_1 \vee x_2$. The dashed lines indicate the multi-sets corresponding to each decision.

layers, corresponding to the n variables $x_1, \dots, x_n$. A decision node corresponding to an existential variable has two emanating decision arcs whereas a decision node corresponding to a universal variable has one. Furthermore, we assume $m+1$ leaf nodes, with the first m corresponding to the m clauses $c_1, \dots, c_m$ of the boolean formula ϕ.

Next, we describe the decisions, the routing probabilities and the relevant multi-sets. Each decision arc emanating from an existential node corresponds to a truth assignment of the corresponding variable. Each such decision arc leads with certainty to a multi-set that (i) drives tokens to the leaf nodes corresponding to the satisfied clauses or, if no clause is satisfied, a token to the $(m+1)^{th}$ leaf node, and (ii) drives one more token to the decision node in the subsequent layer. On the other hand, the single decision arc that corresponds to a universal node leads to two distinct multisets with probability $\frac{1}{2}$. Each such multiset corresponds to a truth assignment for the corresponding universal variable, and is constructed in a similar fashion as before. Finally, we assign a unit requirement to the first m leaf nodes and a requirement of zero to the last leaf node of the acyclic graph.

We claim that the optimal expected cost of $\mathcal{E}(X, \mathcal{A}, \mathcal{P}, \mathcal{N}; \phi)$ is equal to one if and only if formula ϕ is satisfiable. Suppose that the optimal expected cost is 1. In other words, we can choose a decision at the first decision node such that for any multiset chosen in the second node, there is a decision in the third node e.t.c. such that all leaf nodes satisfy their unit requirement. Then it is obvious that this policy defines a truth assignment for the first existential variable x_1 such that for every truth assignment of the second variable x_2, there is a truth assignment to x_3 etc, such that all the clauses are satisfied.

Conversely, if the quantified formula $\exists x_1 \forall x_2 \exists x_3 \ldots \forall x_n \phi$ is true, there is a truth assignment for x_1, such that for every truth assignment of x_2 there is a truth assignment for x_3 etc, such that ϕ comes out true. This last statement can be translated into a policy for choosing the appropriate decisions so that at least one token reaches every one of the first m leaf nodes in a single traversal of the corresponding graph, thus resulting in an optimal expected cost of one.□

As mentioned in the introduction of this chapter, currently we lack a clear-cut result regarding the complexity of the original ONV and the ONV-II problem versions. As an intermediary step to the development of such a complexity characterization for the ONV-II problem, in the next section we show that the well known problem of *"Poisson-tree" scheduling* [31] reduces polynomially to the ONV-II problem, and therefore, the latter is at least as difficult as the former. Beyond assisting with positioning the ONV-II problem in the broader landscape of the computational complexity theory, the provided reduction will also reveal the underlying affinity of the ONV problem to the problems addressed by the more classical stochastic scheduling theory.

6.2 A complexity result for the ONV-II problem

We proceed with the development of this section, by providing first a brief description of the "Poisson-tree" scheduling problem, borrowed from [31]. This scheduling problem is defined by a triplet $\Theta = (m, \tau, \Gamma)$, where:

- m denotes the number of the identical processors that are available in the system.
- $\tau = \{T_1, \ldots, T_n\}$ denotes a finite set of tasks that must be processed by the system processors. It is further assumed that the processing time of each task is exponentially distributed with rate equal to one.
- $\Gamma = (\tau, \Omega)$ is a *rooted in-tree* – i.e., a directed acyclic graph with out-degree of at most one – that expresses a set of *precedence constraints* imposed on the task set τ.

The problem objective is to identify a *schedule* – i.e., a policy for assigning tasks to the available processors – that minimizes the expected *makespan* – i.e., the expected completion time of the last task – while respecting the imposed precedence constraints.

The *memoryless property* possessed by the exponential distribution [38] implies that (i) the natural decision epochs for this scheduling problem are determined by the task completion times, and that (ii) the uncompleted tasks can be scheduled preemptively at those points. We shall refer to the interval between two consecutive decision epochs as a *processing cycle*. The uniformly unit-valued task processing rates imply that (i) a processing cycle involving k processors has an expected duration of $1/k$, and that (ii) the probability for any of the k processed tasks to finish first is also equal to $1/k$. Next we exploit these insights in order to provide a polynomial reduction of a k-processor "Poisson-tree" scheduling problem to an ONV-II problem. For ease of exposition, we restrict our discussion to the 3-processor case; however, all the key ideas underlying the following developments carry over to any k-processor version of the problem. Furthermore, the provided reduction for the 3-processor case has its own significant value, since, to the best of our knowledge, it remains an open problem in terms of the formal characterization of its computational complexity.

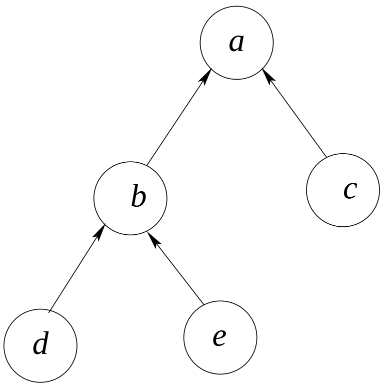

Figure 18: The rooted in-tree modelling the precedence constraints for the tasks of the "Poisson-tree" scheduling problem Θ considered in this example.

Table 1: A tabular characterization of the stochastic graph $\mathcal{G}$ and the visitation requirement vector $\mathcal{N}$ corresponding to the ONV-II problem instance $\mathcal{E}(\Theta)$.

Node	Action	Outcomes and their Distribution	Visitation Req.
x^0	a_1	$(a, 1/3), (x^l, 2/3)$	0
	a_2	$(b, 1/3), (x^l, 2/3)$	
	a_3	$(c, 1/3), (x^l, 2/3)$	
	a_4	$(d, 1/3), (x^l, 2/3)$	
	a_5	$(e, 1/3), (x^l, 2/3)$	
	a_6	$(b, 1/3), (c, 1/3), (x^l, 1/3)$	
	a_7	$(c, 1/3), (d, 1/3), (x^l, 1/3)$	
	a_8	$(c, 1/3), (e, 1/3), (x^l, 1/3)$	
	a_9	$(d, 1/3), (e, 1/3), (x^l, 1/3)$	
	a_{10}	$(c, 1/3), (d, 1/3), (e, 1/3)$	
a	a_{11}	$(b, 1)$	1
	a_{12}	$(c, 1)$	
b	a_{13}	$(d, 1)$	1
	a_{14}	$(e, 1)$	
c	$\emptyset$		1
d	$\emptyset$		1
e	$\emptyset$		1
x^l	$\emptyset$		0

Theorem 20 *The decision version of the 3-processor "Poisson-tree" scheduling problem reduces polynomially to the decision version of the ONV-II problem.*

Proof: Given an instance $\Theta = (m, \tau, \Gamma)$ of the "Poisson-tree" scheduling problem, the corresponding instance $\mathcal{E}(\Theta) = (X, \mathcal{A}, \mathcal{P}, \mathcal{N})$ of the ONV-II problem is defined as follows (the reader is referred to Figure 18 and Table 1 for a more concrete example of this construction):

- $X = \tau \cup \{x^0, x^\lambda\}$. In the graph $\mathcal{G}$ of the constructed ONV-II problem, x^0 will play the role of the root node, while x^λ is a terminal node with zero requirements that will enable the modelling of the losses resulting from the under-utilization of the system processors.
- The action set $\mathcal{A}$ is defined as follows:
  - For each node $T_i \in \tau$, the action set $\mathcal{A}(T_i)$ is defined by the set of its incoming arcs in graph Γ.
  - The actions set $\mathcal{A}(x^0)$ is defined by all the single, two and three-element subsets of the task set τ, which do not contain pairs of tasks associated through the precedence relationship defined by Γ.
  - Finally, $\mathcal{A}(x^\lambda) = \emptyset$ (as already mentioned, x^λ is a terminal node).
- The transition function $\mathcal{P}$ establishes the following connectivity:
  - For each node $T_i \in \tau$, the action corresponding to an incoming arc (T_j, T_i) leads deterministically to node T_j.
  - The action at node x^0 corresponding to a task set $\{T_i\}$ leads to node T_i with probability 1/3, and to node x^λ with probability 2/3. On the other hand, the action corresponding to a task set $\{T_i, T_j\}$ leads to each of these two nodes with respective probability 1/3, and to node x^λ with the remaining

probability. Finally, an action corresponding to a triplet $\{T_i, T_j, T_k\}$ leads to each of these three nodes with respective probability 1/3.

- The visitation requirement vector $\mathcal{N}$ assigns a *unit* visitation requirement to each node $T_i \in \tau$ and a zero visitation requirement to x^0 and x^λ.

Clearly, the above construction of $\mathcal{E}(\Theta)$ can be performed in polynomial time with respect to the size of the defining elements of problem Θ. Furthermore, a scheduling decision d applied during a processing cycle of the original problem Θ, can be simulated in the context of the ONV-II problem $\mathcal{E}(\Theta)$ through the selection of the action $a \in \mathcal{A}(x^0)$ that corresponds to the tasks selected by d, and the resulting outcomes will have the same transition structure in each problem context. At the same time, the deterministic[1] policies applied during any single traversal of the graph $\mathcal{G}$ in problem $\mathcal{E}(\Theta)$ also have a mapping decision in the original problem Θ, with the same transition structure for the resulting outcomes. More specifically, given a state $(x^0, \mathcal{N}^c)$ for problem $\mathcal{E}(\Theta)$, the application over a single traversal of the graph $\mathcal{G}$ of a policy π that, starting from node x^0, selects the action corresponding to a single task T_i and once in the subtree emanating from node T_i follows deterministically a path leading to an active target node T_j, can be interpreted as the scheduling decision of processing only the available task T_j during the corresponding processing cycle of problem Θ. Also, similar interpretations apply to policies π that select actions at state x^0 corresponding to two or three tasks, and subsequently, they reach deterministically one of the target nodes in the resulting subtree. Hence, it is possible to simulate any policy π of Θ on $\mathcal{E}(\Theta)$ and vice versa.

To conclude the proof, it suffices to show that the value functions for any pair of policies π, π' related through the aforementioned simulation, satisfy $V^\pi / V^{\pi'} = a$, for some pre-determined constant a (since, then, there will exist a policy π for Θ with

[1] Confining this analysis to the set of deterministic policies is enabled by the relevant MDP/SSP theory that guarantees the existence of a deterministic optimal policy.

$V^{\pi} < K$ *iff* there exists a policy π' for $\mathcal{E}(\Theta)$ with $V^{\pi'} < K/a$). Next we show, through an induction on $|\tau|$, that $a = 1/3$. Indeed, for the base case of $|\tau| = 1$, there will be only one busy processor during the relevant processing cycle, and therefore, $V^{\pi} = 1$, while the simulation of the corresponding decision in the $\mathcal{E}(\Theta)$ context will result in $V^{\pi'} = 3$. For a problem Θ with $|\tau| > 1$, consider that the aforestated relationship holds true for all "Poisson-tree" scheduling problems involving a number of tasks less than or equal to $|\tau| - 1$. Furthermore, let τ^1 denote the set of tasks scheduled by π during the first processing cycle, and also let $\Theta \backslash T_i$ denote the "Poisson-tree" scheduling problem resulting from Θ through the removal from the task set τ of task $T_i \in \tau^1$. Then, it is easy to see that

$$V^{\pi}(\Theta) = (\text{Expected duration of first processing cycle}) + \frac{1}{|\tau^1|}\sum_{T_i \in \tau^1} V^{\pi}(\Theta \backslash T_i) \quad (352)$$

and a similar equation applies to $V^{\pi'}(\mathcal{E}(\Theta))$, i.e.,

$$V^{\pi'}(\mathcal{E}(\Theta)) = (\text{Expected duration until the first visitation}) + \frac{1}{|\tau^1|}\sum_{T_i \in \tau^1} V^{\pi'}(\mathcal{E}(\Theta \backslash T_i)) \quad (353)$$

The induction hypothesis implies that $V^{\pi}(\Theta \backslash T_i)/V^{\pi'}(\mathcal{E}(\Theta \backslash T_i)) = 1/3$ for every task $T_i \in \tau^1$, and the reader can easily verify that the ratio of the first terms in the right-hand-sides of Equations 352 and 353 is also equal to 1/3. Hence, in this case, $V^{\pi}(\Theta)/V^{\pi'}(\mathcal{E}(\Theta)) = 1/3$, as well. □

6.3 Discussion

The work presented in this chapter positioned some of the variations of the ONV problem considered in Chapters 3 and 5, in the problem hierarchy established by the theory of computational complexity. It was, thus, shown that the introduction of the splitting effect in the ONV problem renders it PSPACE-hard. On the other hand, the ONV problem with internal visitation requirements was shown to be at least as hard as the "Poisson-tree" scheduling problem, a well known problem in scheduling theory whose computational complexity remains an open issue. However, the computational complexity of the original ONV problem remains unresolved and constitutes an open problem.

CHAPTER VII

A PRACTICAL IMPLEMENTATION OF THE PROPOSED PAC-LEARNING ALGORITHM AND ITS EMPIRICAL EVALUATION

The work presented in this chapter integrates the PAC-learning algorithm of Chapter 2 with the ONV results of Chapters 3 and 4 into a practical RL algorithm, and evaluates its empirical performance. The proposed algorithm is shown to be a substantial improvement of the original algorithm developed in Chapter 2, in terms of, both, the involved computational effort and the attained performance, where the latter is measured by the accumulated reward. The new algorithm also leads to a robust performance gain over the typical Q-learning implementations for the considered problem context.

7.1 The need for efficient routing policies for the proposed PAC-learning algorithm

The reader should recall from the development of the PAC-learning algorithm of Chapter 2, that the maximum number of episodes, N, that can be executed by the algorithm before its termination, is determined before the initiation of the algorithm, on the basis of the specified parameters δ and ϵ, and some of the parameters involved in the definition of the problem structure $\mathcal{E}$. Hence, the algorithm described in that chapter can fail either (i) because it did not manage to determine a complete policy $\hat{\pi}(x)$, $\forall x \in X$, within the specified episode budget, N, or (ii) because the chosen policy $\hat{\pi}(x)$, $\forall x \in X$, had an error $\texttt{err}(\hat{\pi}) > \varepsilon$. In Chapter 2 it was shown that a total success probability of $1 - \delta$ can be guaranteed by respectively limiting the probability of failure according to each of the above two failure modes to δ^I and δ^{II}, such that

$$\delta^I + \delta^{II} = \delta \tag{354}$$

The allocation of the sampling effort across all state-action pairs and the determination of an apparent optimum action at every actively explored state, in a way that guarantees the PAC requirement, was based on the embedding into the algorithm of results coming from the statistical inference area of R&S. In particular, the total number of obsevations that the learning agent is required to collect for every state-action pair, (x, a), is obtained from Equation 11, by substituting: (i) k with $|\mathcal{A}(x)|$; (ii) $\bar{v}$ with $(L - l + 1)\bar{v}$, where l is the level of node x; (iii) ε with $\varepsilon(x) = \varepsilon/(L+1)$; and (iv) δ with $\delta(x) = \delta^{II}/|X| = \delta/(2|X|)$. Hence, for every actively explored state $x \in X$ and every action $a \in \mathcal{A}(x)$, the learning agent is required to collect

$$n(x, a) = \lceil \frac{4\bar{v}(x)^2}{\varepsilon(x)^2} \ln(\frac{|A(x)| - 1}{\delta(x)}) \rceil \tag{355}$$

samples of the cumulative reward that results by taking action a in state x and following the pre-determined policy in the remaining nodes, until the completion of the running episode.

On the other hand, in order to control failure according to mode (i), the algorithm must employ a routing policy that will enable the collection of the state-action samples, within the specified episode budget N, with probability δ^I. Furthermore, in order for the resulting scheme to be *efficient* according to the definitions of computational learning theory, the specified episode budget N must be polynomially related to $1/\delta$, $1/\epsilon$ and the other problem-defining parameters. The existence of an episode budget N and of a corresponding routing policy presenting the aforementioned properties, was resolved in Chapter 2 by recognizing that the collection of a single observation during a single episode, constitutes a Bernoulli trial with its success probability bounded from below by $\underline{q}$.

However, it should be clear from the above discussion that, while efficient according to the relevant CLT definition, the routing scheme underlying the specification of the episode budget N in Chapter 2 is very simplistic, since it tries to collect the requested samples one at a time, while ignoring completely the interdependencies and complementarities that are implied by the structure of the underlying state space. It is natural to expect that the consideration of these additional dynamics, and the exploitation of any available information about them through a pertinently designed routing policy, can result in substantial gains with respect to the number of episodes, N, that is required for the successful completion of the underlying sampling process. This topic is systematically addressed in the next section, that discusses an enhanced version of the original PAC-learning algorithm of Chapter 2.

7.2 An enhanced PAC-learning algorithm

This section discusses a series of enhancements for the PAC-learning algorithm developed in Chapter 2 that can lead to (i) a much more expedient execution of the sampling process that is employed by that algorithm, and consequently, (ii) to substantial increases of the total reward that will be accumulated by the algorithm over any given time-span. The first of these enhancements concerns the development of a more pertinent routing policy by employing the ONV problem results of Chapters 3 and 4 in a heuristic attempt to govern the sampling process, and it is addressed in the following paragraph.

Integrating the results on the ONV problem to the PAC-learning algorithm of Chapter 2 We remind the reader that, according to the description of the PAC-learning algorithm provided in Chapter 2, during any single episode, the states of the underlying acyclic state space are partitioned into unexplored, actively explored and (fully) explored states, and the learning agent traverses this state space, starting from the single initial state, and seeking to sample the cumulative reward resulting from a partially explored action of some actively explored state. It should be clear that during the execution of the n-th task episode by the learning agent, the subset of all the unexplored states plus the states that can be reached from them through a single transition, constitute an acyclic connected digraph $\mathcal{G}^n$, with a single source node, x^0, and with the set of terminal nodes containing all the actively explored states. Furthermore, each actively explored state has associated with it a remaining number of samples that must be collected until it becomes fully explored, and it can be perceived as a visitation requirement for that state. The visitation requirements for the entire set of actively explored states are collectively expressed by vector $\mathcal{N}^n$. Hence, the first enhancement of the PAC-learning algorithm of Chapter 2 proposed in this section, routes the learning agent to collect the next sample by (i) solving the relaxing LP corresponding to the ONV problem instance defined by $\mathcal{G}^n$ and $\mathcal{N}^n$, and (ii) using the resulting randomized policy $\pi^{adrel}(n)$ as the corresponding routing policy. Furthermore, in the solution of the aforementioned ONV problem, the algorithm uses the estimates of the branching probabilities $p(x; a)$ that are computed on the basis of the experienced history $< x_0, a_0, x_1, a_1, \ldots, x_t, a_t >$. A more detailed description of this logic is as follows:

- The algorithm updates the empirical estimates $\hat{p}(\cdot; a),\ x \in X, a \in \mathcal{A}(x)$ of the branching probabilities $p(\cdot; a), x \in X, a \in \mathcal{A}(x)$, every time that an action $a \in \cup_{x \in X} \mathcal{A}(x)$ is exercised.

- At the beginning of the n-th episode, the algorithm solves the relaxing LP defined by (i) the running instance $\mathcal{G}^n$ of the acyclic digraph, (ii) the set of empirical branching probabilities $\hat{p}(\cdot; a),\ x \in X, a \in \mathcal{A}(x)$, and (iii) the visitation requirement vector $\mathcal{N}^n$. It subsequently derives the randomized policy $\pi^{rel}(n)$ defined on the set of unexplored states.

- The algorithm traverses the subset of unexplored states by applying the randomized policy $\pi^{rel}(n)$ until it reaches an actively explored or a fully explored state.

We shall refer to the resulting algorithm as the *PAC2-learning algorithm.*

Expediting the sample collection through reuse of the historical experience
A second heuristical modification of the PAC-learning algorithm of Chapter 2, that can lead to very dramatic reductions in the number of episodes that are necessary for the collection of the samples indicated by Equation 355, is based on the capture of the transitional dynamics and the rewards experienced by the algorithm in a pertinent set of data structures, and on the exploitation of this information for reconstructing a part of the requested samples, every time that a new state becomes actively explored. More specifically, in addition to the information extracted by the algorithmic versions discussed in the earlier parts of this chapter, this new variation maintains the following two data structures:

- A vector Ω defined over the set of actions $a \in \cup_{x \in X \setminus X^L} \mathcal{A}(x)$. The component $\Omega(a)$ of this vector expresses the total immediate reward that has resulted from the execution of action a over the entire history of the learning process.

- Another vector W defined over the tuples (a, x) with $a \in \cup_{x \in X \setminus X^L} \mathcal{A}(x)$ and $x \in \mathcal{S}(a)$. The component $W(a, x)$ of this vector expresses the number of times that the execution of action a resulted in state $x \in \mathcal{S}(a)$, during the entire history of the learning process.

Every time that a state $x \in X \setminus X^L$ becomes actively explored, each action $a \in \mathcal{A}(x)$ is pre-assigned $\sum_{x' \in \mathcal{S}(a)} W(a, x')$ samples with a total observed reward equal to

$$\Omega(a) + \sum_{x' \in \mathcal{S}(a)} W(a, x') \cdot \hat{V}^{\hat{\pi}}(x') \tag{356}$$

Table 2: The intervals defining the *uniform* distributions of the immediate rewards that result from the different actions

a_1	[0.1,0.4]	a_2	[0.2,0.5]	a_3	[-0.2, 0.6]	a_4	[0.1, 0.5]
a_5	[-0.2, 0.7]	a_6	[0.2, 0.5]	a_7	[0.1, 0.6]	a_8	[-0.1,0.4]
a_9	[-0.5, 0.8]	a_{10}	[-0.5, -0.1]	a_{11}	[-0.5, 0.2]	a_{12}	[-0.6,-0.2]
a_{13}	[-0.2,0.5]	a_{14}	[-0.3, 0.1]	a_{15}	[-0.2,0.4]	a_{16}	[-0.1,0.1]
a_{17}	[-1.0,1.2]	a_{18}	[-0.4,0.4]	a_{19}	[-0.1, 0.1]	a_{20}	[0.1,0.5]
a_{21}	[-0.1,0.4]	a_{22}	[-0.2,0.7]	a_{23}	[1.0,1.4]	a_{24}	[-0.2,0.7]
a_{25}	[-0.1,0.8]	a_{26}	[0.1,0.2]	a_{27}	[-0.2,0.4]	a_{28}	[-0.1,0.1]
a_{29}	[-0.1, 0.0]						

The quantities $\hat{V}^{\hat{\pi}}(x')$, that appear in the right-hand-side of Equation 356, denote the estimated values of the explored states $x' \in \mathcal{S}(a)$ under the fixed policy $\hat{\pi}$, and they are obtained during the exploration of the corresponding states. In this way, the additional samples to be collected by the algorithm with respect to each action $a \in \mathcal{A}(x)$, until state x becomes fully explored, are reduced to

$$[n(x,a) - \sum_{x' \in \mathcal{S}(a)} W(a,x')]^+ \tag{357}$$

The algorithmic variation obtained by the integration of the logic expressed by Equations 356 and 357 in the PAC2-learning algorithm, will be referred to as the *PAC3-learning algorithm*. It is worth-noticing that Equations 356 and 357 imply that the number of episodes required for the collection of the samples indicated by Equation 355 can be even less than $\sum_{x \in X} \sum_{a \in \mathcal{A}(x)} n(x,a)$, an effect that is actually manifested in the computational study reported in the next section.

We conclude this section by providing in Figures 19 and 20 a complete description of the PAC3-learning algorithm, based on the original developments of Chapter 2 and on the two modifications introduced in this section.

7.3 A computational study of the proposed algorithm

In this section we present the results of a series of experiments that intend (i) to highlight the gains attained by the PAC2 and PAC3-learning algorithms with respect

```
**Input:** L; $X^l,\ l = 0, \dots, L$; $\mathcal{A}(x), \forall x \in X$; $\bar{v}$; $\underline{q}$; $\varepsilon$; $\delta^I$; $\delta^{II}$
**Output (under successful completion):** $\hat{\pi}(x), \forall x \in X$

  I. **Initialization**

     (a) Compute $X \equiv \bigcup_{l=0}^{L} X^l$; $|X|$; $|\mathcal{A}(x)|, \forall x \in X$;

     (b) Set
         $\bar{v}(x) := (L - l + 1)\bar{v},\ \forall l = 0, \dots, L,\ \forall x \in X^l$;
         $\varepsilon(x) := \varepsilon/(L + 1),\ \forall x \in X$;
         $\delta(x) := \delta^{II}/|X|,\ \forall x \in X$;
         $n(x) := \lceil \frac{4\bar{v}(x)^2}{\varepsilon(x)^2} \ln(\frac{|A(x)|-1}{\delta(x)}) \rceil, \forall x \in X$;
         $\sigma := \sum_{x \in X} |\mathcal{A}(x)| n(x)$;
         $N := \sigma \lceil (1/\underline{q}) \ln(\sigma/\delta^I) \rceil$;
         $\hat{p}(x; a) := 1/|\mathcal{S}(a)|,\ \forall x \in \mathcal{S}(a),\ a \in \mathcal{A}(x),\ x \in X \backslash X^L$;
         $Q(x, a) := 0, \forall x \in X, \forall a \in \mathcal{A}(x)$;
         $O(x, a) := 0, \forall x \in X, \forall a \in \mathcal{A}(x)$;
         $\Omega(a) := 0, \forall a \in \cup_{x \in X \backslash X^L} \mathcal{A}(x)$;
         $W(a, x) := 0, \forall a \in \cup_{x \in X \backslash X^L} \mathcal{A}(x),\ \forall x \in \mathcal{S}(a)$;
         $AE := X^L$; $UE := \bigcup_{l=0}^{L-1} X^l$;
         $i := 1$
```

Figure 19: The proposed PAC3-learning algorithm for the RL problem considered in this work: Initialization

```
II. Policy Computation
    while ($AE \neq \emptyset \wedge i \leq N$) do
      (a) Initiate a new episode by placing the agent at the initial state, $x^0$, solve the relaxing LP corresponding to the current ONV problem instance, and route the agent to an actively explored state, $x \in AE$, by picking actions according to the implied randomized policy. For every exercised action $a \in \bigcup_{x \in X \setminus X^L} \mathcal{A}(x)$, update the corresponding empirical probabilities $\hat{p}(\cdot; a)$ and the data structures $\Omega$ and $W$.
      (b) If an actively explored state $x \in AE$ is successfully reached,
          i. select an action $a \in \mathcal{A}(x)$ for which $O(x,a) < n(x)$;
          ii. obtain an observation $\Psi(x,a)$, by accumulating the total reward obtained by exercising action $a$ at state $x$, and subsequently following the pre-computed policy $\hat{\pi}$ until the termination of the current episode;
          iii. $Q(x,a) := Q(x,a) + \Psi(x,a)$;  $O(x,a) := O(x,a) + 1$;
          iv. If ($O(x,a) = n(x)$)
              • $Q(x,a) := Q(x,a)/n(x)$;
              • If ($\forall a' \in \mathcal{A}(x), O(x,a') = n(x)$)
                  – $\hat{\pi}(x) := \arg\max_{a \in \mathcal{A}(x)}\{Q(x,a)\}$;
                  – remove state $x$ from $AE$;
                  – Remove from $UE$ every state $x'$ that does not have any immediate successor states in $AE \cup UE$, and add it to $AE$. Initialize the sampling process for each such state $x'$ according the logic of Equations 356 and 357.
      (c) $i := i + 1$
    endwhile
III. Exit
    If ($AE = \emptyset$) return $\hat{\pi}(x), \forall x \in X$, else report failure
```

Figure 20: The proposed PAC3-learning algorithm for the RL problem considered in this work: Main Iteration and Exit

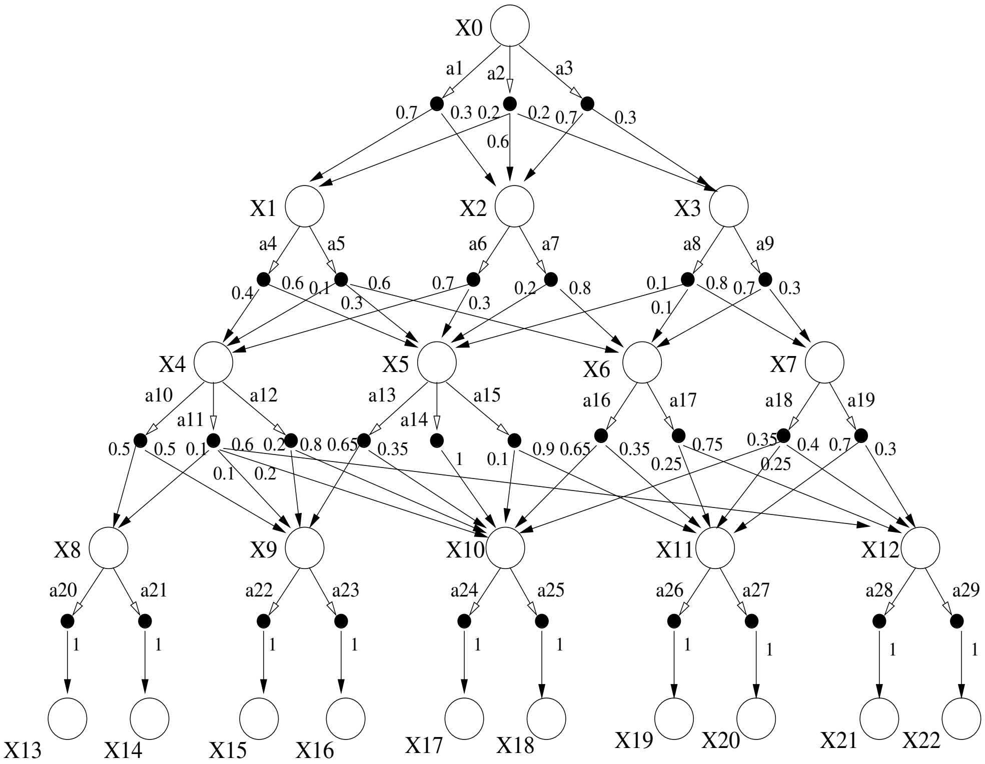

Figure 21: The stochastic acyclic digraph used in the presented experiments

to the original PAC-learning algorithm presented in Chapter 2 and the standard Q-learning algorithm, and (ii) to provide some insights on the comparative advantages of the proposed approach. These insights also suggest a set of guidelines for the effective implementation of the proposed algorithm. The presented experiments are based on the problem instance defined by the graph structure presented in Figure 21 and the distributions of the action immediate rewards listed in Table 2. But the reported findings reflect our experiences with a broader set of graph structures and parameterizations. We also notice, for further reference, that the optimal policy for the RL problem defined by Figure 21 and Table 2, has a value equal to 1.322.

Characterizing the gains attained by the enhanced sampling process of the PAC2 and PAC3-learning algorithms Our first experiment seeks to register the computational gains that are attained by the enhancements for the sampling

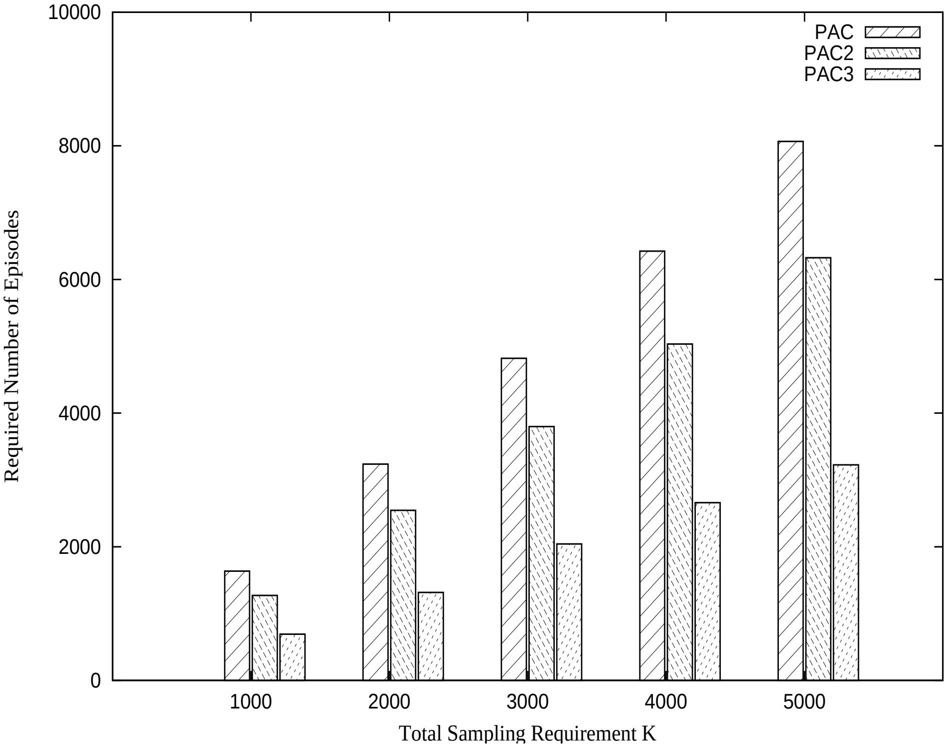

Figure 22: Characterizing the gains attained by the enhanced sampling process of the PAC2 and PAC-3 learning algorithms

process of the original PAC-learning algorithm that were introduced in Section 7.2. To this end, we measured the number of episodes that are required by the original PAC-learning algorithm of Chapter 2, as well as by the PAC2 and PAC3-learning algorithms defined in Section 7.2, in order to perform a total number of K visitations to the different nodes $x \in X$ of the graph depicted in Figure 21.[1] More specifically, these K visitations were apportioned to the different graph nodes $x \in X$ according to the proportions $n(x)/\sigma$, where $n(x)$ and σ were calculated from the formula of Equation 355, by using $\epsilon = 0.1$, $\delta^{II} = 0.2$ and $\bar{v} = 1.5$. Figure 22 reports the obtained results when K was set equal to 1000, 2000, 3000, 4000 and 5000. Each of the depicted values is the average of 100 replications for the corresponding case. It is obvious from

[1] We remind the reader that in the context of the proposed learning algorithm, each node visitation corresponds to the collection of a sample $\Psi(x, a)$ for some action $a \in \mathcal{A}(x)$.

the provided data that each of the proposed enhancements leads to a substantial reduction of the number of episodes required for the coverage of the posed visitation requirements, and that the PAC3-learning algorithm presents the best performance. Finally, it is worth-noticing that for all cases presented in Figure 22, the PAC3-learning algorithm manages to complete the requested sampling process in a number of episodes that is substantially smaller than the requested number of samples K, and this gain increases with the value of K. This effect results from the ability of this algorithm to reconstruct some of the requested samples from the historical data captured in the data structures Ω and W, and it is in agreement with the relevant remarks made in Section 7.2.

Comparing the performance of the PAC3-learning algorithm to the performance of the Q-learning algorithm The results of the previous experiment clearly demonstrate that among the original PAC-learning algorithm of Chapter 2 and the PAC2 and PAC3-learning algorithms proposed in this work, the PAC3-learning algorithm results in the fastest execution of the sampling process, and therefore, to the most expedient learning of an optimized acting policy. Hence, in this experiment we seek to compare the performance of the PAC3-learning algorithm against the performance of the more classical Q-learning algorithm, and develop some insights about the relative merits of these two algorithms. In the considered problem setting, this comparison can be based on the tracking of the cumulative reward that is collected by each of these two algorithms over the execution of a certain number of episodes.

We begin the detailed discussion of the pursued experiment by reminding the reader that the classical Q-learning algorithm updates the Q-factor estimate $Q(x,a)$ upon the t-th execution of action $a \in \mathcal{A}(x)$, $x \in X$, by setting [40]

$$Q(x,a) := (1-\gamma_t(x,a)) \cdot Q(x,a) + \gamma_t(x,a) \cdot (r_t + Q(y)) \tag{358}$$

In the recursion of Equation 358, r_t denotes the experienced immediate reward, drawn from the distribution $\mathcal{D}(\mu(a), v(a))$, $y \in \mathcal{S}(a)$ denotes the state that resulted from the execution of action a, $Q(y) = \max_{a \in \mathcal{A}(y)}\{Q(y, a)\}$, and $\gamma_t(x, a) \in (0, 1)$ is the *learning rate* applied during the t-th execution of this recursion. Furthermore, the works of [42, 5] have shown that if (i) the sequence $\{\gamma_t(x, a),\ t = 1, 2, \ldots\}$ is chosen such that

$$\sum_{t=1}^{\infty} \gamma_t = \infty \ \wedge \ \sum_{t=1}^{\infty} \gamma_t^2 < \infty \quad a.s. \tag{359}$$

and (ii) every state-action pair $(x, a),\ x \in X \backslash X^L,\ a \in \mathcal{A}(x)$, is exercised an infinite number of times, then the algorithm estimates, $Q(x, a)$, will converge to the optimal values, $Q^*(x, a)$, with probability 1, irrespectively of the initial values of the $Q(x, a)$ estimates.

A practical way to guarantee the requirements of Condition (i) above, is by having the learning rates $\gamma_t(x, a)$, decrease asymptotically to zero, according to the schedule

$$\gamma_t(x, a) := c_1/(c_2 + t) \tag{360}$$

where c_1 and c_2 are positive constants. In the considered experiment, we regulated the learning rates involved in our Q-learning implementations by applying the schedule of Equation 360 with $c_1 = c_2 = 1.0$.

On the other hand, Condition (ii) for the convergence of the Q-learning algorithm is frequently satisfied by introducing a small positive parameter $\theta \in (0, 1)$, and adopting an action selection scheme that, at every node $x \in X$, selects an action a corresponding to a maximal $Q(x, a)$ estimate with probability $1 - \theta$, and an alternative random action a' with probability θ. Clearly, larger values of the randomizing probability θ can result in more aggressive exploration, and therefore, more expedient learning of the correct Q-values, but at the same time, they compromise the ability of the learning agent to benefit from this enhanced information, by forcing it to select a suboptimal action more frequently. Hence, it is customary that the pricing of the randomizing probability θ is reduced towards zero through a number of stages, where the agent emphasis shifts incrementally from “exploration” to “exploitation”. In the presented experiment this idea is implemented by partitioning the execution of the Q-learning algorithm into two phases, I and II: Phase I runs over a predetermined number of episodes, K, with the parameter θ fixed on a preselected value. Phase II runs over the remaining episodes until the termination of the algorithm, and during this phase, the agent selects a suboptimal action at each visited node $x \in X$ with probability $1/(1 + \nu(x))$, where $\nu(x)$ denotes the number of times that node x has been visited during the entire execution of the algorithm. We shall indicate the aforementioned dependence on the parameter θ by characterizing the resulting implementation as the $Q(\theta)$-learning algorithm.

We perceive the PAC3-learning algorithm proposed in this paper, as an alternative mechanism for implementing Phase I in the Q-learning implementations on RL problems that evolve episodically over acyclic state spaces.[2] When viewed from such an implementational standpoint, the computational results presented in the following essentially demonstrate that the explicit consideration and facilitation of the informational flow that underlies the design of the PAC3-learning algorithm can lead to a more efficient learning process and to higher reward accumulations compared to the Q-learning implementations described in the previous paragraph, where the control of the algorithm exploration is done only through the pricing of the randomizing probability θ. This is especially true for problem instances where the optimal Q-values of the different actions at any given state are quite close to each other and the distributions / spread of the corresponding rewards present significant overlap.

At the same time, our experiments have also revealed that the sampling requirements expressed by Equation 355 are overly conservative, leading to an unnecessarily long (exploration) Phase I. In order to remedy this drawback of the PAC3-learning

[2] Indeed, as discussed in Chapter 2, it is pertinent to complement these algorithms with a Q-learning-based Phase II, similar to that described in the previous paragraph, since such an augmentation can compensate for potential erroneous choices made by the PAC-learning algorithm and provides robustness to non-stationarities.

algorithm while maintaining the efficiencies that can result from the explicit consideration of the underlying informational flow, we propose the discounting of the original (cumulative) sample size $\sigma(\epsilon, \delta^{II})$, that is computed for some imposed PAC requirements ϵ and δ^{II}, to some empirically selected value K,[3] and the apportioning of this cumulative sampling requirement to the different node-action pairs (x, a) according to the proportions $n(x, a; \epsilon, \delta^{II})/\sigma(\epsilon, \delta^{II})$. Then, according to the formulae provided in Figure 19, the sample sizes allocated to any two nodes x and x', that belong respectively to the graph layers $l(x)$ and $l(x')$, will satisfy:

$$\frac{\hat{n}(x)}{\hat{n}(x')} = \frac{n(x)}{n(x')} = \left(\frac{L - l(x) + 1}{L - l(x') + 1}\right)^2 \cdot \frac{\ln(1/\delta^{II}) + \ln(|X|) + \ln(|A(x)| - 1)}{\ln(1/\delta^{II}) + \ln(|X|) + \ln(|A(x')| - 1)} \tag{361}$$

Furthermore, when $\ln(1/\delta^{II}) + \ln(|X|) \gg \ln(|A(x)| - 1)$, $\forall x \in X$, Equation 361 simplifies to

$$\frac{\hat{n}(x)}{\hat{n}(x')} \approx \left(\frac{L - l(x) + 1}{L - l(x') + 1}\right)^2 \tag{362}$$

Equations 361 and 362 reveal that the PAC parameters ϵ and δ^{II} are fundamental for the determination of the overall sampling effort to be pursued by the proposed PAC-learning algorithms, but, in most cases, they do not affect the distribution of this effort across the different nodes. In the context of the practical implementation of the PAC3-learning algorithm outlined above, Equations 361 and 362 imply that the pricing of the aforementioned parameter K to some arbitrarily selected value is tantamount to the selection of certain pricing(s) for the parameters ϵ and δ^{II}. On the other hand, once the pricing of the parameter K has been determined, the parameters ϵ and δ^{II} become inconsequential for the subsequent execution of the algorithm. Hence, it can be concluded from the above remarks that the algorithmic variation that results from the introduction of the parameter K and the suggested apportioning of the overall sampling effort to the different decision nodes, is coherent

[3] Notice that the role of K in this implementation of the PAC3-learning algorithm is similar to its role in the Q-learning implementation described in the previous paragraphs; i.e., the determination of the length of the exploration Phase I.

and consistent with the informational flow that is sought be the original PAC-learning algorithms.

Finally, we notice, for completeness, that as manifested in the first experiment reported in this section, it is quite possible that the proposed implementation of the PAC3-learning algorithm will collect the requested K samples in a number of episodes substantially smaller than K. In order to cover this possibility, we stipulate that the algorithm will switch to Phase II either upon the execution of K episodes, or as soon as it completes the selection of the required K samples, whichever event occurs first.

Figures 23 and 24 present the results of a computational experiment that compares the performance of the practical, two-phased implementation of the PAC3-learning algorithm that was described in the previous paragraphs, with that of the $Q(\theta)$-learning algorithm. This experiment was based on the learning task defined by the graph of Figure 21 and the reward distributions of Table 2. Each of the six pairs of plots presented in Figures 23 and 24 depict the evolution of the rewards accumulated by the PAC3 and the $Q(\theta)$-learning algorithms when the length of the exploration phase, K, was respectively set to 1000, 2000, 3000, 4000, 5000 and 6000. Furthermore, in each of these six cases, the reported performance of the $Q(\theta)$-learning algorithm is the best observed when θ was varied over the interval $(0, 1)$ with an increment of 0.05. Finally, each curve presented in these plots averages the results of 300 replications for the corresponding case.

A first observation that can be made on the provided results is that, in all of the presented cases, the PAC3-learning algorithm has a long-term performance that is better than or comparable to the corresponding optimized performance of the $Q(\theta)$-learning algorithm. A closer study of the provided plots reveals that the enhanced performance of the PAC3-learning algorithm results indeed from its ability to provide a more balanced solution to the "exploration vs. exploitation" dilemma. More specifically, the provided plots indicate that low values of K – i.e., a short exploration

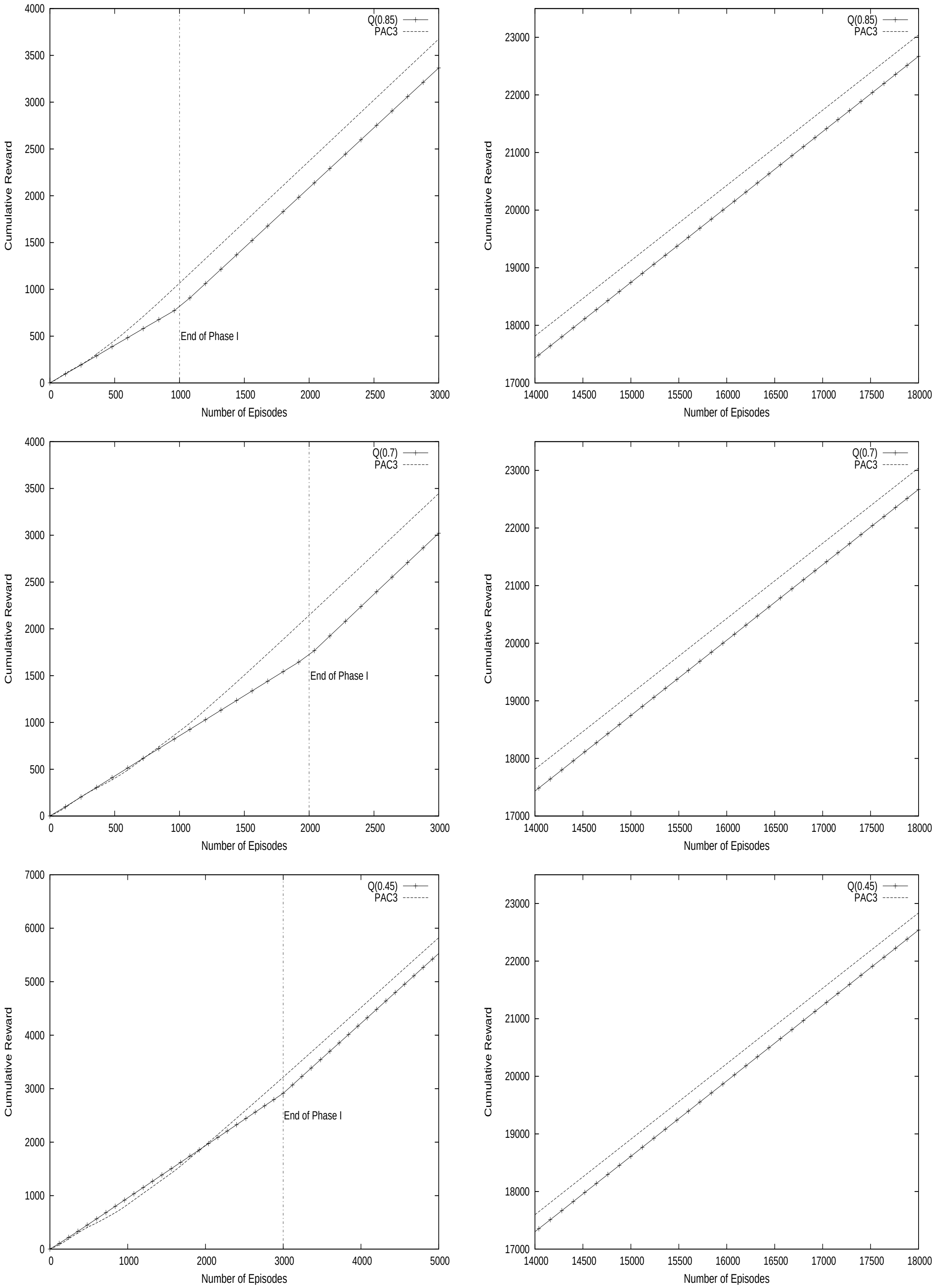

Figure 23: Relative performance of the $Q(\theta)$ and PAC3-learning algorithms for different selections of K and an optimized selection of the parameter θ

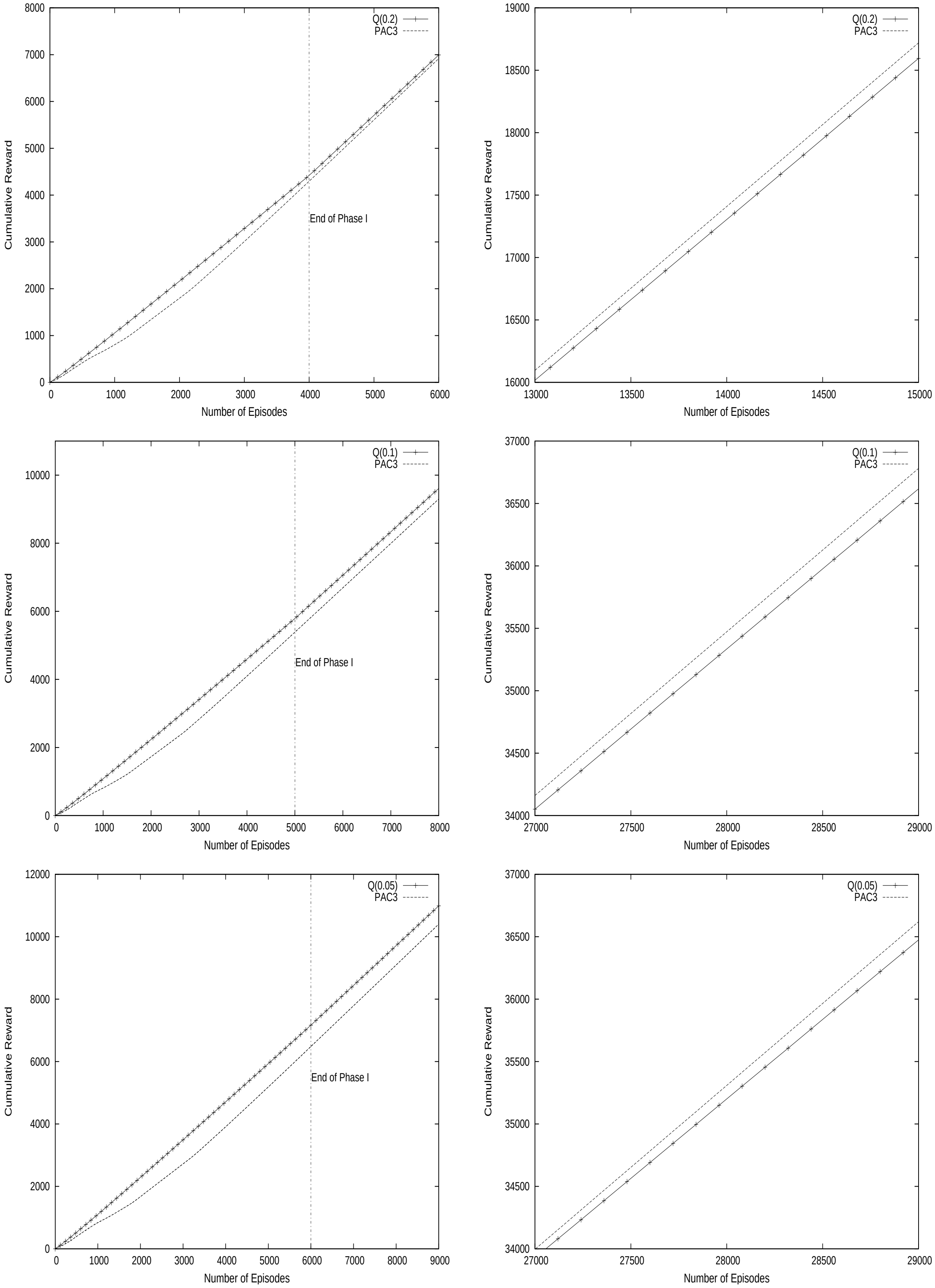

Figure 24: Relative performance of the $Q(\theta)$ and PAC3-learning algorithms for different selections of K and an optimized selection of the parameter θ (cont.)

Phase I – force the $Q(\theta)$-learning algorithm to adopt an over-agressive exploration attitude, manifested by the high optimized values for the randomizing probability θ. Such an approach seems to result in a correct identification of the optimal policy by the $Q(\theta)$-learning algorithm in Phase I, but at the same time, it incurs substantial reward losses in that phase, due to over-exploration, from which the algorithm is unable to recover in the subsequent phase. On the other hand, higher values of K allow the $Q(\theta)$-learning algorithm to be more conservative with respect to the applied exploration scheme during Phase I, as manifested by the lower optimized values for the parameter θ. Such a conservative selection of θ enables the $Q(\theta)$-learning algorithm to even out-perform the PAC3-learning algorithm in the earlier stages, since it places an early emphasis on exploitation. But the PAC3-learning algorithm is eventually able to catch up and even dominate the $Q(\theta)$-learning algorithm in the long run, apparently due to the better quality of the learned policy.

The above remarks are supported by our broader experimentation with additional learning tasks defined by different graph structures and distributions for the immediate rewards. In all the performed experiments the proposed PAC3-learning algorithm demonstrated a long-run performance that was comparable to or better than the performance of the $Q(\theta)$-learning algorithm. Even more importantly, this enhanced performance was attained in a robust and straightforward manner, since the proposed implementation of the PAC3-learning algorithm does not necessitate the empirical tuning of any unspecified parameters, like the parameter θ of the Q-learning algorithm. Finally, as already mentioned, the identified dominance of the PAC3 over the $Q(\theta)$-learning algorithm seems to increase with the difficulty of the underlying learning task, where the latter is determined by the proximity of the value of the optimal action at each problem state to the values of the suboptimal actions at that state.

7.4 Discussion

The work presented in this chapter provided a practical, customized learning algorithm for reinforcement learning tasks that evolve episodically over acyclic state spaces. The presented results essentially complement the earlier developments on the considered problem, that were presented in Chapter 2. Extensive computational experimentation established that the proposed algorithm is a substantial improvement of the original algorithm developed in Chapter 2, in terms of, both, the involved computational effort, and the attained performance, where the latter is measured by the accumulated rewards. The presented algorithm also leads to a robust performance gain over the typical Q-learning algorithm.

We should mention at this point that the proposed PAC3-learning algorithm is easily extensible to episodic tasks that involve multi-threaded traversals of the underlying acyclic state space. Such a multi-threading effect results, for instance, in the context of the ODP problem that was presented in the introductory section, by the disassembly operations involved in it. The necessary modifications for applying the PAC3-learning algorithm in this new operational context pertain to the specification of the reward accumulation process and the employment of the ONV-I and ONV-II problem variations of Chapter 5.

On the other hand, an issue that remains an open challenge with respect to the developments presented in this manuscript, is the *"optimum"* pricing of the parameter K that is involved in the practical implementation of the PAC3-learning algorithm proposed in Section 7.3. Given the apparent intractability of this problem, a possible way to deal with it is through the substitution of the sampling scheme presented in Figure 19, by an *on-line*, adaptive sampling scheme, that determines the required sample sizes incrementally, on the basis of the already accumulated information on the values of the assessed actions and their variance. The development of such an incremental sampling process can be based on the adaptation and extension of a similar set of results that have been developed in the area of R&S and involve single-stage decision making and normally distributed rewards.[4] The investigation of this possibility and the potential gains that can result from it defines an interesting line for future research work on this problem.

[4] e.g., cf. [28]; also, another set of results that can provide a base for such a development are those presented in [16]

CHAPTER VIII

CONCLUDING REMARKS AND FUTURE WORK

In this work we presented an efficient PAC-learning algorithm for episodic tasks over acyclic state spaces. The defining characteristic of our algorithm is that it takes explicitly into consideration the structure of the underlying state space thus rendering it easy to implement. Furthermore, we move one step further from characterizing the sampling requirements by expediting the sampling process through the introduction of efficient routing policies that will help the algorithm complete its exploration program. This last requirement gave rise to a family of novel stochastic control problems that are characterized as the Optimal Node Visitation problems. A large part of this work concerns the systematic modelling and analysis of the ONV problem variations. The last part of this research program explores the computational merits obtained by heuristical implementations that result from the integration of the ONV problem developments into the PAC-algorithms developed in the first part of this work. The work presented in this last part reinforces and confirms the driving assumption of this research, i.e., that one can design customized RL algorithms of enhanced performance if the underlying problem structure is taken into account.

A first line of future research would seek to identify a tighter bound for the episode budget, N , of the PAC-learning algorithm presented in Chapter 2. The presented developments sought to explicitly establish the ability of the proposed algorithm to guarantee the PAC requirement, within a number of episodes that is polynomially related to the parameters of interest, rather than provide the tightest possible bound for such an episode budget. We shall consider the possibility of replacing the R&S criterion of Theorem 1 with other R&S criteria that will employ sampling techniques of more sequential nature, e.g., similar to those discussed in [16, 28]. Regarding the ONV problem, a future line of research would seek to extend the analysis of the ONV and ONV-I problems to the ONV-II problem variation. In particular, the remaining open issues that present further research opportunities are (i) the existence of tractable fluid relaxations that capture more thoroughly the ONV-II problem dynamics implied by the introduction of the internal visitation requirements with precedence constraints, and, (ii) whether those relaxations can become the basis for the design of suboptimal control schemes with guaranteed performance bounds. Along this line, future work will seek capitalize upon further insights and results from stochastic scheduling theory, in order to identify additional structure and suboptimal policies for the ONV problem variations. Finally, another line of future work will seek the integration of the results on the ONV-I,II problem variations with the multi-threading property, in the application context that motivated this work at the first place.

APPENDIX A

A STOPPING TIME RESULT FOR RANDOM VARIABLES WITH A PERMUTATION DISTRIBUTION

The result provided in this appendix is useful for the analysis provided in Chapter 4. We define a sequence of random variables that resemble a sampling without replacement process, and prove a result for an appropriately defined, associated stopping time. This result is relevant to the analysis of Chapter 4, because a part of the requirement reduction process under π^{adrel} resembles a sampling without replacement process.

A.1 Random variables with a permutation distribution

Given a vector $\mathcal{N} \in \mathbb{Z}_+^L$, assume a sequence of random vectors $Z^\lambda \in \mathbb{Z}^L,\ \lambda = 1, \ldots, |\mathcal{N}|$, that take values on the set of the L-dimensional unit vectors $r^1, \ldots, r^L$. If Z_j^λ denotes the j^{th} component of $Z^\lambda,\ \ j = 1, \ldots, L$ the distribution of Z^λ's is given by

$$P(Z^1 = r^j) = \frac{\mathcal{N}_j}{|\mathcal{N}|}, \tag{363}$$

and

$$P(Z^{\lambda+1} = r^j) = \frac{\mathcal{N}_j - \sum_{k=1}^{\lambda} Z_j^k}{|\mathcal{N}| - \lambda}, \tag{364}$$

$j = 1, \ldots, L,\ \lambda = 1, \ldots, |\mathcal{N}|$. Notice that the random sequence $Z^1, \ldots, Z^{|\mathcal{N}|}$ can be viewed as a sequencial sampling procedure without replacement from a population of $|\mathcal{N}|$ objects. The population consists of L different types of objects, with $\mathcal{N}_j$ objects of each type. Then, the random vector Z^λ indicates the object type of the λ^{th} draw.

Now for any vector $d \in \mathbb{R}^L$, consider the random variables

$$G^\lambda = \sum_{j=1}^{L} d_j \cdot Z_j^\lambda, \; \lambda = 1, \ldots, |\mathcal{N}|. \tag{365}$$

Then, in the light of the above interpretation for the random vectors Z^λ, the random variables G^λ have the following interpretation: Suppose a finite population $\mathcal{D}$ consists of $|\mathcal{N}|$ values that belong to $\{d_1, \ldots, d_L\}$ with $\mathcal{N}_j$ values equal to d_j. Then, the sequence $G^1, \ldots, G^{|\mathcal{N}|}$ represents the sample values obtained by randomly sampling the population $\mathcal{D}$ without replacement. Consider the vector $\mathcal{N}$ when it is scaled by a factor $n \in \mathbb{Z}_+$ and the corresponding sampling without replacement expressed by the sequence $G^1, \ldots, G^{n \cdot |\mathcal{N}|}$. If $\bar{d} = \sum_{j=1}^{L} d_j \frac{\mathcal{N}_j}{|\mathcal{N}|}$, then we have the following proposition:

Proposition 7 *Let* $T^n = \min\{k \leq n|\mathcal{N}| : \; \sum_{\lambda=1}^{k} G^\lambda > n \cdot |\mathcal{N}| \cdot \bar{d}\}$*. If* $\bar{d} > 0$*, then*

$$\lim_{n \to \infty} E[n \cdot |\mathcal{N}| - T^n] < \infty. \tag{366}$$

The proof of this proposition, is provided at the end of this appendix. In order to derive the proof, we need to take a closer look at the random sequence $G^1, \ldots, G^{n \cdot |\mathcal{N}|}$. First, let us consider more closely the case of the unscaled requirement vector $\mathcal{N}$. Since the G^i's represent the sample values obtained by randomly sampling the population $\mathcal{D}$ without replacement, the sequence $G^1, \ldots, G^{|\mathcal{N}|}$, possess a *permutation* distribution [24]. That is, if $g = (g_1, \ldots, g_k)$ is a set of real numbers, then the joint distribution of $(G^1, \ldots, G^k)$ takes as values all the $k!$ permutations of g with equal probabilities. Some basic properties of random variables with a permutation distribution are summarized below.

Property P1 [34] The r.v's $G^\lambda, \; \lambda = 1, \ldots, L$, are *equidistributed* as $P(G^\lambda = d_j) = \frac{\mathcal{N}_j}{|\mathcal{N}|}, \; j = 1, \ldots, L$.

Property P2 [34] The r.v's $G^\lambda, \; \lambda = 1, \ldots, L$ are *exchangeable*. That is, the joint distribution of any k of the r.v. G^λ is the same as that of the first k of them.

It is evident that the variables $G^1, \ldots, G^{|\mathcal{N}|}$ are dependent and someone would expect that given the first k sample values $G^1, \dots, G^k$, the sample G^{k+1} would be negatively correlated with the first k samples. This idea is formalized in [24], where it is proved that:

Property P3 [24]. Random variables with a permutation distribution are *Negatively Associated* (NA).

In general, the random variables $X_1, \dots, X_n$ are said to be NA if for every pair of disjoint subsets A_1, A_2 of $\{1, \dots, n\}$,

$$\mathsf{Cov}\{f_1(X_i, i \in A_1), f_2(X_j, j \in A_2)\} \leq 0 \tag{367}$$

whenever f_1 and f_2 are increasing.

The NA property of a sequence of random variables is stronger than that of negative correlation. In the following, we present an interesting result that concerns a sequence of NA random variables, and will be useful in the sequel:

Property P4 [39] For a sequence of NA random variables $X_1, \dots, X_n$ and a sequence of independent random variables $X_1^*, \dots, X_n^*$ such that $X_i =^{st} X_i^*, i = 1, \dots, n$, ("$=^{st}$" denotes equality in distribution) we have that

$$E[f(\max_{1 \leq k \leq n}\{\sum_{i=1}^{k} X_i)\}] \leq E[f(\max_{1 \leq k \leq n}\{\sum_{i=1}^{k} X_i^*\})] \tag{368}$$

for any increasing convex function f such that the above expectations exist.

Next, we assume that the vector $\mathcal{N}$ is scaled by a factor $n \in \mathbb{Z}_+$. In the following lemma we prove a property of the random sequence $G^1, \dots, G^{n \cdot |\mathcal{N}|}$ that will be useful in the proof of Proposition 7.

Lemma 11 *For any $a > 0$, we have*

$$\lim_{n \to \infty} E[\max\{k \leq n|\mathcal{N}| : \sum_{\lambda=1}^{k}(-G^\lambda + \bar{d}) > \alpha \cdot k\}] < \infty \tag{369}$$

Proof Let

$$t^n = \max\{k \leq n|\mathcal{N}| : \sum_{\lambda=1}^{k}(-G^\lambda + \bar{d}) > \alpha \cdot k\} \tag{370}$$

From the above equation we have that $\sum_{\lambda=1}^{t^n}(-G^\lambda + \bar{d}) - \alpha \cdot t^n \geq 0$. Hence, we can write

$$\begin{aligned} \frac{\alpha}{2}t^n &\leq \sum_{\lambda=1}^{t^n}(-G^\lambda + \bar{d}) - \alpha \cdot t^n + \frac{\alpha}{2}t^n && (371) \\ &= \sum_{\lambda=1}^{t^n}(-G^\lambda + \bar{d}) - \frac{\alpha}{2}t^n && (372) \\ &\leq \max_{k \leq n|\mathcal{N}|}\{\sum_{\lambda=1}^{k}(-G^\lambda + \bar{d}) - \frac{\alpha}{2}k\} && (373) \\ &= \max_{k \leq n|\mathcal{N}|}\{\sum_{\lambda=1}^{k}(-G^\lambda + \bar{d} - \frac{\alpha}{2})\} && (374) \end{aligned}$$

Hence from Equation 374 we get that

$$\frac{\alpha}{2}E[t^n] \leq E[\max_{k \leq n|\mathcal{N}|}\{\sum_{\lambda=1}^{k}(-G^\lambda + \bar{d} - \frac{\alpha}{2})\}] \tag{375}$$

We already know that the variables G^λ have a permutation distribution and hence are NA. It is trivial to see that the variables $-G^\lambda + \bar{d} - \frac{\alpha}{2},\ \lambda = 1, \ldots, n|\mathcal{N}|$, also have a permutation distribution and, hence, are equi-distributed and NA. Assume the i.i.d. random variables G^{λ^*} such that

$$P(G^{\lambda^*} = d_j) = \frac{\mathcal{N}_j}{|\mathcal{N}|},\ j = 1, \ldots, L. \tag{376}$$

$\lambda = 1, \ldots, n|\mathcal{N}|$. From Property 1, we know that $G^\lambda =^{st} G^{\lambda^*},\ \lambda = 1, \ldots, n \cdot |\mathcal{N}|$. Furthermore, from Property 4 we can claim that

$$E[\max_{k \leq n|\mathcal{N}|}\{\sum_{\lambda=1}^{k}(-G^\lambda + \bar{d} - \frac{\alpha}{2})\}] \leq E[(\max_{k \leq n|\mathcal{N}|}\{\sum_{\lambda=1}^{k}(-G^{\lambda^*} + \bar{d} - \frac{\alpha}{2})\}] \tag{377}$$

Notice that the random variables $-G^{\lambda^*} + \bar{d} - \frac{\alpha}{2}$ are a.s. bounded random variable with a negative expectation, i.e., $E[-G^{\lambda^*} + \bar{d} - \frac{\alpha}{2})] = -\frac{\alpha}{2} < 0$. Hence, from Theorem 1 of [23], we get

$$E[\max_{k}\{\sum_{\lambda=1}^{k}(-G^{\lambda^*} + \bar{d} - \frac{\alpha}{2})\}] \ < \ \infty \tag{378}$$

and therefore,

$$\begin{aligned} \lim_{n\to\infty} E[\max_{k\leq n\cdot|\mathcal{N}|}\{\sum_{\lambda=1}^{k}(-G^{\lambda^*}+\bar{d}-\frac{\alpha}{2})\}] &\leq E[\max_{k}\{\sum_{\lambda=1}^{k}(-G^{\lambda^*}+\bar{d}-\frac{\alpha}{2})\}] && (379)\\ &< \infty. && (380) \end{aligned}$$

Finally, Lemma 11 follows from Equations 370, 375, 377, and 380 ∎

Proof of Proposition 7. We have that

$$\begin{aligned} & E[n\cdot|\mathcal{N}|-T^n] && (381)\\ = \;& E[(n\cdot|\mathcal{N}|-\min\{k\leq n|\mathcal{N}|:\ \sum_{\lambda=1}^{k}G^{\lambda}>n\cdot|\mathcal{N}|\cdot\bar{d}\}] && (382)\\ = \;& E[\max_{k\leq n|\mathcal{N}|}\{n\cdot|\mathcal{N}|-k:\ \sum_{\lambda=1}^{k}G^{\lambda}>n\cdot|\mathcal{N}|\cdot\bar{d}\}] && (383)\\ = \;& E[\max_{k\leq n|\mathcal{N}|}\{n\cdot|\mathcal{N}|-k:\ \sum_{\lambda=1}^{k}(G^{\lambda}-\bar{d})>\bar{d}\cdot(n\cdot|\mathcal{N}|-k)\}] && (384) \end{aligned}$$

From the definition of G^{λ}, Z^{λ} and $\bar{d}$, we notice that:

$$\begin{aligned} \sum_{\lambda=1}^{n|\mathcal{N}|}(G^{\lambda}-\bar{d}) &= \sum_{\lambda=1}^{n|\mathcal{N}|}G^{\lambda}-\sum_{\lambda=1}^{n|\mathcal{N}|}\bar{d} && (385)\\ &= \sum_{\lambda=1}^{n|\mathcal{N}|}\sum_{j=1}^{L}d_j\cdot Z_j^{\lambda}-n\cdot\sum_{j=1}^{L}d_j\cdot\mathcal{N}_j && (386)\\ &= \sum_{j=1}^{L}d_j\sum_{\lambda=1}^{n|\mathcal{N}|}\cdot Z_j^{\lambda}-n\cdot\sum_{j=1}^{L}d_j\cdot\mathcal{N}_j && (387)\\ &= \sum_{j=1}^{L}d_j\cdot n\cdot\mathcal{N}_j-n\cdot\sum_{j=1}^{L}d_j\cdot\mathcal{N}_j && (388)\\ &= 0. && (389) \end{aligned}$$

Hence,

$$\sum_{\lambda=1}^{k}(G^{\lambda}-\bar{d})=0-\sum_{\lambda=k+1}^{n|\mathcal{N}|}(G^{\lambda}-\bar{d}) \qquad (390)$$

Recall that, according to Property 2, the random variables G^{λ} are exchangeable. Then, the joint distribution of $(G^1,\ldots,G^{n|\mathcal{N}|-k})$ is the same as the joint distribution of $(G^{k+1}, \ldots, G^{n|\mathcal{N}|})$ for all $k = 0, \ldots, n|\mathcal{N}|$. Hence, we have that

$$\sum_{\lambda=1}^{n|\mathcal{N}|-k} G^{\lambda} =^{st} \sum_{\lambda=k+1}^{n|\mathcal{N}|} G^{\lambda}, \tag{391}$$

$k = 0, \ldots, n|\mathcal{N}|$. From Equations 390 and 391 we get that

$$\sum_{\lambda=1}^{k}(G^{\lambda} - \bar{d}) =^{st} - \sum_{\lambda=1}^{n|\mathcal{N}|-k} (G^{\lambda} - \bar{d}),\ k = 0, \ldots, n|\mathcal{N}|. \tag{392}$$

Then, from Equation 392, Equation 384 becomes

$$\begin{aligned} & E[\max_{k \le n \cdot |\mathcal{N}|} \{n \cdot |\mathcal{N}| - k :\ \sum_{\lambda=1}^{k}(G^{\lambda} - \bar{d}) > \bar{d} \cdot (n \cdot |\mathcal{N}| - k)\} & (393) \\ = \ & E[\max_{k \le n \cdot |\mathcal{N}|} \{n \cdot |\mathcal{N}| - k :\ -\sum_{\lambda=1}^{n|\mathcal{N}|-k} (G^{\lambda} - \bar{d}) > \bar{d} \cdot (n \cdot |\mathcal{N}| - k)\}] & (394) \\ = \ & E[\max\{k \le n \cdot |\mathcal{N}| :\ -\sum_{\lambda=1}^{k}(G^{\lambda} - \bar{d}) > \bar{d} \cdot k\}] & (395) \end{aligned}$$

Now, from Lemma 11, we have that

$$\lim_{n \to \infty} E[\max\{k \le n \cdot |\mathcal{N}| :\ -\sum_{\lambda=1}^{k}(G^{\lambda} - \bar{d}) > \bar{d} \cdot k\}] < \infty \tag{396}$$

From Equations 384, 395 and 396 the proposition follows. ∎

APPENDIX B

A FLUID RELAXATION FOR THE ONV-II PROBLEM

In this appendix we provide with a fluid-based relaxation of the ONV-II problem of Chapter 5. Remember that the ONV-II version is an extension of the ONV problem that is obtained by the introduction of visitation requirements for the internal nodes of the stochastic digraph. The fluid relaxation must take into account the ONV-II problem property that the visitation requirements of a node $x \in X$ will start to be satisfied only after the complete satisfaction of the visitation requirements of the successor nodes. In order to accommodate this requirement, the fluid relaxation constitutes a continuous-time flow control problem and it is described in the next section.

B.1 The fluid relaxation for the ONV-II problem as a continuous time flow control problem

The flow control problem that will serve as a fluid based relaxation for the ONV-II problem concerns the transferring of some required amounts of fluid to different nodes of an acyclic graph, while minimizing the potential losses that are incurred by (i) the presence of nodes with uncontrollable routing, and (ii) the imposition of precedence constraints on the satisfaction of the fluid requirements at these nodes. This problem can be described as follows: We are given a network, modeled by an acyclic, connected digraph $G = (V, E)$, where V and E denote respectively the sets of the graph nodes and edges. Furthermore, V is partitioned to two node classes, V^c and V^u. The sets of source and leaf nodes of graph G are respectively denoted by ${}^{\bullet}V$ and $V^{\bullet}$, and it is further assumed that ${}^{\bullet}V = \{v_0\}$.[1] To facilitate the subsequent discussion, we also use the following notation: $E^{\bullet}(v)$ will denote the set of edges emanating from node v, ${}^{\bullet}E(v)$ will denote the set of edges leading to node v, and ${}^{\bullet}E^{\bullet}(v)$ will denote the entire set of edges incident on v. Finally, $v^{\bullet}$ will denote the set of the immediate successors of node v, i.e., $v^{\bullet}$ collects all the terminating nodes of the edges $e \in E^{\bullet}(v)$. A fluid is pumped into this network through its source node v_0, at a constant rate of one unit of fluid per unit of time. Flow reaching a node $v \in V^u$ is distributed to its outgoing edges according to an uncontrollable and time invariant distribution $d_v =< d_v(e),\ e \in E^{\bullet}(v) >$. On the other hand, the distribution of the flow reaching a node $v \in V^c$ to its emanating edges is controllable and it can be varied over time. Finally, each node $v \in V$ has a fluid requirement $\bar{F}(v)$ associated with it, and it is also stipulated that node v can begin accumulating the incoming flow in order to fulfill its requirement $\bar{F}(v)$ only after all of its successor nodes in graph G, $v' \in V$, have fulfilled their own requirements, $\bar{F}(v')$. A node v that can proceed to the accumulation of its fluid requirement, $\bar{F}(v)$, will be characterized as *activated*. It will be further characterized as *completed*, when the accumulated amount of fluid reaches the designated level $\bar{F}(v)$. The control problem considered in this work is the determination of a (time-dependent) routing policy for nodes $v \in V^c$ that will enable the completion of all the nodal requirements $\bar{F}(v),\ v \in V$, in *minimal time* (or equivalently, while pumping the minimal amount of fluid into the network). An example problem instance is presented in Figure 25. In the depicted digraph, uncontrollable nodes are represented by black circles. The nodal fluid requirements are reported by the numbers in bold, on the right side of each node, and the distributions characterizing the routing pattern at the uncontrollable nodes are reported by the numbers on the left of the edges emanating from these nodes.

[1] It should be obvious from the subsequent discussion that the assumption $|{}^{\bullet}V| = 1$ is not restrictive at all, since any problem instance with $|{}^{\bullet}V| > 1$ can be easily reduced to a problem instance with $|{}^{\bullet}V| = 1$ through the addition of a new dummy node; the details are left to the reader.

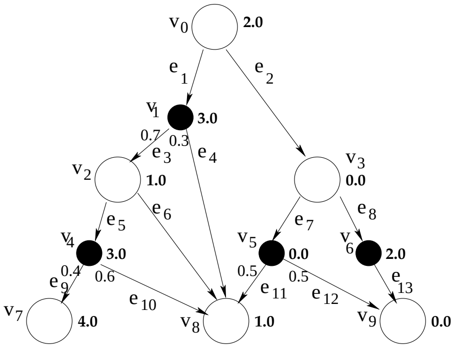

Figure 25: An example problem instance

B.2 A succinct representation of the flow control problem

The problem introduced in the previous section can now be naturally formulated as continuous-time optimal control problem. In this modeling framework, the key "decision variables" are the functionals $f_e(t),\ t \in [0, \infty)$, defining a flow profile for each edge $e \in E$. We shall also use the functionals $F_v(t),\ t \in [0, \infty)$, to denote the evolution of the fluid accumulation at node $v \in V$. Then, the considered problem can be succinctly expressed as follows:

$$\min \int_0^\infty I_{\{F_{v_0}(t) \leq \bar{F}(v_0)\}} dt \tag{397}$$

s.t.

$$\sum_{e \in E^\bullet(v_0)} f_e(t) = 1.0 \tag{398}$$

$$\forall v \in V^c \backslash (V^\bullet \cup^\bullet V),\ \forall t \in [0, \infty),$$

$$\sum_{e \in E^\bullet(v)} f_e(t) = \sum_{e' \in {}^\bullet E(v)} f_{e'}(t) \tag{399}$$

$$\forall v \in V^u \backslash V^\bullet,\quad \forall e \in E^\bullet(v),\ \forall t \in [0, \infty),$$

$$f_e(t) = d_v(e) \cdot \sum_{e' \in {}^\bullet E(v)} f_{e'}(t) \tag{400}$$

$$\forall v \in V,\ \forall t \in [0, \infty),\quad F_v(t) =$$

$$\int_0^t \Big(\sum_{e \in {}^\bullet E(v)} f_e(\tau)\Big) \cdot I_{\{\forall v' \in v^\bullet,\ F_{v'}(t) \geq \bar{F}(v')\}} d\tau \tag{401}$$

$$\forall e \in E,\ \forall t \in [0, \infty), \quad f_e(t) \geq 0 \tag{402}$$

Constraint 398 in the above formulation expresses the finiteness of the ingress capacity of the considered network and establishes the equivalence between the cumulative amount of fluid entering this network and the passage of time. Constraints 399 and 400 impose a flow balance requirement for the set of nodes $V \backslash ({}^{\bullet}V \cup V^{\bullet})$, with Constraint 400 further communicating the uncontrollable nature of the flow routing that takes place at nodes in V^u. Constraint 401 expresses the accumulation of the fluid required by the different nodes $v \in V$, and it ensures that it is in agreement with the precedence constraint defined in the introductory section. Finally, the problem objective function seeks the completion of all the nodal fluid requirements in minimal time (or in the light of Constraint 398, with a minimal loss of fluid).

B.3 A structural property of the considered optimal control problem

While the formulation of Equations 397-402 offers a succinct characterization of the considered problem, it is very cumbersome from a computational standpoint. However, next we present a structural property that will enable its transformation to a mixed integer program, which is readily solvable through canned optimization software [43]. The main essence of this property is that the restriction of the original problem to flows $< f_e(t),\ e \in E,\ t \in [0, \infty) >$ that maintain a constant distribution at all nodes $v \in V$ between two consecutive completions of some fluid requirements, does not compromise the optimality of the derived solution. This result can be stated and proven as follows:

Proposition 8 *Let* $< f_e(t),\ e \in E;\ F_v(t),\ v \in V;\ t \in [0, \infty) >$ *denote a feasible solution for the formulation of Equations 397-402, and consider a time interval* $[t_1, t_2]$ *such that*

$$\forall v \in V, \quad I_{\{F_v(t_1) < \bar{F}(v)\}} = I_{\{F_v(t_2) < \bar{F}(v)\}} \tag{403}$$

Then, there exists a solution $< f'_e(t),\ e \in E;\ F'_v(t),\ v \in V;\ t \in [0,\infty) >$ *such that*

$$\forall e \in E,\ \forall t \in [t_1, t_2],\ f'_e(t) = c_e \tag{404}$$

and

$$\int_0^\infty I_{\{F'_{v_0}(t) \le \bar{F}(v_0)\}} dt = \int_0^\infty I_{\{F_{v_0}(t) \le \bar{F}(v_0)\}} dt \tag{405}$$

Proof Consider the flow $< f'_e(t),\ e \in E;\ t \in [0,\infty) >$ that is defined by flow f as follows:

$$\forall e \in E,\quad f'_e(t) = \begin{cases} \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} f_e(\tau) d\tau, & \text{if } t \in [t_1, t_2] \\ f_e(t), & \text{otherwise} \end{cases} \tag{406}$$

Clearly, the flow f' defined by Equation 406 satisfies the condition of Equation 404 and it also satisfies Constraint 402 of the problem formulation provided in Section B.2. Furthermore, the definition of f', together with the linearity of the integral, imply that f' is also feasible with respect to Constraints 398-400 of that formulation. Next we consider the fluid accumulations $< F'_v(t),\ v \in V,\ t \in [t, \infty) >$, that are induced by f' through the integral of Equation 401, and we establish that

$$\forall t \in \{t_1, t_2\},\ \forall v \in V,\quad F'_v(t) = F_v(t) \tag{407}$$

The validity of Equation 407 for $t = t_1$ follows immediately from the definitions of f' and F' (c.f., Equations 406 and 401). The validity of Equation 407 for $t = t_2$ can be established inductively as follows: First consider the set of leaf nodes and notice that for any such node $v \in V^{\bullet}$, $I_{\{\forall v' \in v^{\bullet},\ F_{v'}(t) \geq \bar{F}(v')\}} = 1,\ \forall t \in [0, \infty)$. Hence,

$$\begin{aligned} \forall v \in V^{\bullet},\quad F'_v(t_2) &= \\ F'_v(t_1) + \int_{t_1}^{t_2} \sum_{e \in {}^{\bullet}E(v)} f'_e(\tau) d\tau &= \\ F_v(t_1) + \int_{t_1}^{t_2} [\sum_{e \in {}^{\bullet}E(v)} \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} f_e(s) ds] d\tau &= \\ F_v(t_1) + \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} d\tau \int_{t_1}^{t_2} \sum_{e \in {}^{\bullet}E(v)} f_e(s) ds &= \\ F_v(t_1) + \int_{t_1}^{t_2} \sum_{e \in {}^{\bullet}E(v)} f_e(s) ds &= \\ F_v(t_2) & \end{aligned} \tag{408}$$

For the inductive step, consider a node $v \in V \backslash V^{\bullet}$, and suppose that

$$\forall v' \in v^{\bullet},\quad F'_{v'}(t_2) = F_{v'}(t_2) \tag{409}$$

Then, if there exists a node $v' \in v^{\bullet}$ with $F'_{v'}(t_2) = F_{v'}(t_2) < \bar{F}(v')$, Constraint 401 implies that

$$F'_v(t_2) = F_v(t_2) = 0 \tag{410}$$

In the opposite case, $F'_{v'}(t_2) = F_{v'}(t_2) \geq \bar{F}(v')$, $\forall v' \in v^{\bullet}$, which combined with Equation 403 and the established validity of Equation 407 for $t = t_1$, further implies that

$$\begin{aligned} \forall t \in [t_1, t_2],\quad I_{\{\forall v' \in v^{\bullet},\ F'_{v'}(t) \geq \bar{F}(v')\}} &= \\ I_{\{\forall v' \in v^{\bullet},\ F_{v'}(t) \geq \bar{F}(v')\}} &= 1 \end{aligned} \tag{411}$$

But then, the equality of $F'_v(t_2)$ and $F_v(t_2)$ can be established through a computation similar to that presented in Equation 408. Finally, Equation 405 follows from the definition of f' (c.f. Equation 406) and the application of Equations 403 and 407 to node v_0.∎

As already explained, Proposition 8 implies that we can restrict the search for an optimal control law into the class of control laws that allow for a switch of the applied routing scheme only at the time points corresponding to the completion of some fluid requirement. In the light of the above discussion, it is possible to provide a MIP-based formulation of the considered optimal control problem as described in the next section.

B.4 The MIP formulation

The flow dynamics generated by this restricted class of control laws can be modelled using the set of "*control modes*", $\mathcal{V}$, that is defined by all the possible partitions of the node set V, that (i) split it into two subsets, one containing the nodes that have their flow requirements completed, and its complement, and (ii) they are in agreement with the precedence constraints expressed by Equation 401. In order to characterize for each mode $\nu \in \mathcal{V}$, all the possible evolution patterns, towards the fulfillment of the corresponding fluid requirements, we proceed as follows:

First, we introduce the set of auxiliary variables $\{X_e^\nu\}$, which denote the total amount of fluid conveyed through the edges $e \in E$ during the network sojourn in the considered control mode. Clearly, $\{X_e^\nu,\ e \in E\}$ must satisfy the following balance constraints:

$$\forall v \in V^c \backslash (V^\bullet \cup^\bullet V),\ \sum_{e \in E^\bullet(v)} X_e^\nu = \sum_{e' \in ^\bullet E(v)} X_{e'}^\nu \tag{412}$$

$$\forall v \in V^u \backslash V^\bullet,\ \forall e \in E^\bullet,\ X_e^\nu = d_v(e) \cdot \sum_{e' \in ^\bullet E(v)} X_{e'}^\nu \tag{413}$$

$$\forall e \in E, \quad X_e^\nu \geq 0 \tag{414}$$

Second, we introduce the variables $\{\Delta F_v^\nu\}$ that denote the total amount of fluid accumulated at some activated node v during the network sojourn in the considered control mode. Then, $\{\Delta F_v^\nu,\ v \in V;\ \nu \in \mathcal{V}\ \}$ should satisfy the following constraints:

$$\forall \text{ non-activated or completed node } v \text{ in } \nu, \Delta F_v^\nu = 0$$

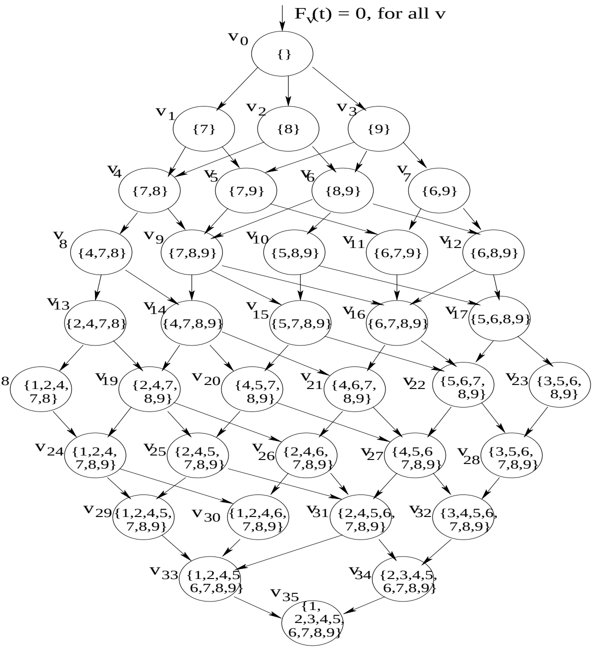

Figure 26: The control modes and interconnecting transitions of the graph $\mathcal{G}$ corresponding to the example problem instance of Figure 25

and

$$\forall \text{ activated but uncompleted node } v \text{ in } \nu,$$

$$\Delta F_v^{\nu} = \begin{cases} \sum_{e' \in {}^{\bullet}E(v)} X_{e'}^{\nu}, & \text{if } v \neq v_0 \\ \sum_{e' \in E(v)^{\bullet}} X_{e'}^{\nu}, & \text{otherwise} \end{cases} \tag{415}$$

At this point we provide with some remarks that concern the computation of the mode set implied by the underlying problem instance. As indicated by the example graph of Figure 26, the enumeration of the viable control modes and the interconnecting transitions can be performed systematically through a search process that starts from the initializing control mode ν_0, where all nodes have their fluid requirements uncovered. Subsequently, the search process reaches out to the subsequent control modes by flagging one node as completed, at a time, while observing the precedence constraints that are imposed by the structure of graph G. The resulting graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ has an acyclic structure, in which the different modes are layered according to their number of completed nodes. For further reference, we shall characterize these layers of $\mathcal{G}$ by $L_0,\ L_1, \ldots, L_{|V|-1}$, where the layer index i corresponds to the metric defining the layer, as discussed above. It is also interesting to notice that the last layer $L_{|V|-1}$ is a singleton, i.e., graph $\mathcal{G}$ has a unique sink node, corresponding to the control mode that activates node v_0.

Any solution of the MIP formulation belonging to the restricted space of control laws characterized in the previous section, can be effectively represented by the following two elements: (i) a directed path from the source to the sink node of the aforementioned graph $\mathcal{G}$, and, (ii) the nodal fluid accumulations that take place at each visited mode. In order to characterize the aforementioned paths of $\mathcal{G}$, we: introduce the binary variables $\delta_\nu, \nu \in \mathcal{V}$, and we stipulate that $\delta_\nu = 1$ *iff* mode ν belongs on the path followed by the considered solution. Obviously, the pricing of the variables $\delta_\nu,\ \nu \in \mathcal{V}$, must be restricted by an additional set of constraints, which will ensure that they express meaningful paths from the source to the sink node of graph $\mathcal{G}$. Letting ${}^\bullet\nu$ denote the immediate predecessors of any mode $\nu \in \mathcal{V}$, such a constraint set can be structured as follows:

$$\forall i \in \{0, 1, \ldots, |V|-1\},\quad \sum_{\nu \in L_i} \delta_\nu = 1 \tag{416}$$

$$\forall \nu \in \mathcal{V} \backslash \{\nu_0\},\quad \delta_\nu \le \sum_{\nu' \in {}^\bullet\nu} \delta_{\nu'} \tag{417}$$

$$\forall \nu \in \mathcal{V},\quad \delta_\nu \in \{0, 1\} \tag{418}$$

Indeed, the combination of Constraints 416 and 418 expresses the fact that any path from the source to the sink mode of $\mathcal{G}$ has exactly one node belonging to each of the layers of $\mathcal{G}$. On the other hand, Constraint 417 enforces the path feasibility with respect to the connectivity of $\mathcal{G}$.

In order to complete the characterization of the space of the control laws considered by the proposed formulation, we must also link the pricing of the variables

$\delta_\nu, \nu \in \mathcal{V}$, to the pricing of the variables $X_e^\nu, e \in E, \nu \in \mathcal{V}$, that define the fluid accumulations at the different control modes. For this, consider a pricing of the variables $\delta_\nu,\ \nu \in \mathcal{V}$, according to any pattern that satisfies Constraints 416–418. Then, it should be clear from the above discussion, that any control law which is in agreement with this pricing, will engage only control modes ν with $\delta_\nu = 1$. Control modes ν with $\delta_\nu = 0$ will not contribute anything to the required fluid accumulations. In the light of Equations 412–414, this effect is communicated in the proposed formulation by setting

$$\forall \nu \in \mathcal{V}, \quad \sum_{e \in E^\bullet(v_0)} X_e^\nu \leq \delta_\nu \cdot M^\nu \tag{419}$$

The parameter M^ν appearing in the above equation is of the, so called, "big-M" type, and it must be adequately large to avoid any unintentional / artificial constraining of the left hand side of Equation 419, in the case that $\delta_\nu = 1$. In the considered problem context, a pertinent value for M^ν is provided by the combined fluid requirement of all the nodes that are activated but not completed in mode ν.

Equations 416–418 combined with Equations 412–415 and Equation 419 provide a complete characterization of the entire set of flows presenting the structure that was identified by Proposition 8. It remains to express the constraints arising by the nodal fluid requirements and the objective function that measures the performance of any such satisficing flow. The constraints imposing the nodal fluid requirements can be succinctly expressed as follows:[2]

$$\forall v \in V, \quad \sum_{\nu \in \mathcal{V}} \Delta F_v^\nu = \bar{F}(v) \tag{420}$$

Similarly, the stated objective of minimizing the overall fluid losses can be expressed

[2] Constraint 420 can be relaxed to a "$\geq$"-type, with some potential gains in computational time. The resulting values for the X_e^ν variables will still define an optimal flow for the original continuous-time optimal control problem of Eqs 397–402, but the order in which the nodal fluid requirements will be completed under this solution, might differ from that suggested by the pricing of the variables δ_ν.

by

$$\min \sum_{\nu \in \mathcal{V}} \sum_{e \in E^{\bullet}(v_0)} X_e^{\nu} \tag{421}$$

The following theorem recapitulates all the above discussion and provides a formal expression to the presented developments. The notation ${}^{\bullet}e$ used in its statement implies the starting node for edge e.

Theorem 21 *Consider the MIP formulation defined by Equations 412–421 and let $X_e^{\nu^*}$, $e \in E$, $\nu \in \mathcal{V}$, denote the modal flows established by its optimal solution. Furthermore, define the flow functionals $f_e(t)$, $e \in E$, $t \in [0, \infty)$, by setting $\forall e \in E$,*

$$f_e(t) = \frac{\sum_{\nu \in L_i} X_e^{\nu^*}}{\sum_{e' \in E^{\bullet}({}^{\bullet}e)} \sum_{\nu \in L_i} X_{e'}^{\nu^*}} \tag{422}$$

if there exists an $i \in \{0, 1, \ldots, |V| - 1\}$ such that $\sum_{j=0}^{i-1} \sum_{\nu \in L_j} \sum_{e' \in E^{\bullet}(v_0)} X_{e'}^{\nu^*} \leq t < \sum_{j=0}^{i} \sum_{\nu \in L_j} \sum_{e' \in E^{\bullet}(v_0)} X_{e'}^{\nu^*}$, and

$$f_e(t) = 0 \tag{423}$$

otherwise. Then, $< f_e(t),\ e \in E,\ t \in [0, \infty) >$ is an optimal flow for the original formulation of Equations 397–402, and

$$\int_0^{\infty} I_{\{F_{v_0}(t) \leq \bar{F}(v_0)\}} dt = \sum_{\nu \in \mathcal{V}} \sum_{e \in E^{\bullet}(v_0)} X_e^{\nu^*} \tag{424}$$

□

We must notice that MIP formulations can be computationally expensive, and, in the considered case, things are further complicated by the fact that the derived MIP formulation involves a number of variables and constrains that is exponentially large with respect to the size of elements that are involved in the original problem statement. As a result, in Chapter 5 we constrain the ONV-II problem over a class of randomized policies which are simpler and accept a fluid relaxation that can be easily addressed through standard optimization techniques.

REFERENCES

[1] Bechhofer, R. E., "A single-sample multiple decision procedure for ranking means of normal populations with known variances," *Annals of Mathematical Statistics*, vol. 25, pp. 16–39, 1954.

[2] Bertsekas, D. P., *Dynamic Programming and Optimal Control.* Belmont, MA: Athena Scientific, 1995.

[3] Bertsekas, D. P., *Nonlinear Programming, 2nd ed.* Belmont, MA: Athena Scientific, 1999.

[4] Bertsekas, D. P., "Dynamic programming and suboptimal control: A survey from ADP to MPC," *European Journal of Control*, vol. 11, pp. 310–334, 2005.

[5] Bertsekas, D. P. and Tsitsiklis, J. N., *Neuro-Dynamic Programming.* Belmont, MA: Athena Scientific, 1996.

[6] Bertsimas, D. and Gamarnik, D., "Asymptotically optimal algorithms for job shop scheduling and packet switching," *Journal of Algorithms*, vol. 33, pp. 296–318, 1999.

[7] Bertsimas, D. and Sethuraman, J., "From fluid relaxations to practical algorithms for job shop scheduling: The makespan objective," *Mathematical Programming*, vol. 92, pp. 61–102, 2002.

[8] Bertsimas, D. and Tsitsiklis, J. N., *Introduction to Linear Optimization.* Belmont, MA: Athena Scientific, 1997.

[9] Billingsley, P., *Convergence of probability measures.* NY: Wiley and Sons, 1968.

[10] Bountourelis, T. and Reveliotis, S., "Optimal node visitation in stochastic digraphs," *IEEE Trans. on Automatic Control, forthcoming.*

[11] Brafman, R. I. and Tennenholtz, M., "R-max - a general polynomial time algorithm for near-optimal reinforcement learning," *J. Mach. Learn. Res.*, vol. 3, pp. 213–231, 2003.

[12] Chung, F. R. K. and Lu, L., "Survey: Concentration inequalities and martingale inequalities: A survey," *Internet Mathematics*, vol. 3, no. 1, 2006.

[13] Chvátal, V., *Linear Programming.* N.Y., N.Y.: W. H. Freeman & Co., 1983.

[14] Cormen, T. H., Leiserson, C. E., and Rivest, R. L., *Introduction to Algorithms, 2nd ed.* Boston, MA: MIT Press, 2001.

[15] Even-Dar, E., Mannor, S., and Mansour, Y., "PAC bounds for multi-armed bandit and Markov decision processes," in *Proceedings of COLT'02*, pp. 255–270, 2002.

[16] Even-Dar, E., Mannor, S., and Mansour, Y., "Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems," *J. Mach. Learn. Res.*, vol. 7, pp. 1079–1105, 2006.

[17] Even-Dar, E. and Mansour, Y., "Learning Rates for Q-learning," *J. Mach. Learn. Res.*, vol. 5, pp. 1–25, 2004.

[18] Feller, W., *An Introduction to Probability Theory and its Applications, Vol. II (2nd. ed.).* N.Y.: Wiley, 1971.

[19] Fiechter, C. N., "Efficient reinforcement learning," in *Proceedings of COLT'94*, pp. 88–97, ACM, 1994.

[20] Glasserman, P. and Yao, D., *Monotone Structure in Discrete-Event Systems.* NY,NY: John Wiley & Sons, Inc., 1994.

[21] Gut, A., "On the moments and limit distibutions of some first passage times," *The Annals of Probability*, vol. 2, No. 2, pp. 277–308, 1974.

[22] Hoeffding, W., "Probability inequalities for sum of bounded random variables," *Journal of the American Statistical Association*, vol. 58, pp. 13–30, 1963.

[23] Janson, S., "Moments for first-passage and last-exit times, the minimum, and related quantities for random walks with positive drift.," *Advances in Applied Probability*, vol. 18, No. 4, pp. 865–879, 1986.

[24] Joag-Dev, K. and Prochan, F., "Negative association of random variables with applications," *The Annals of Statistics*, vol. 11, No. 1, pp. 286–295, 1983.

[25] Kearns, M. and Singh, S., "Finite-sample convergence rates for q-learning and indirect algorithms," *Neural Information Processing Systems*, vol. 11, pp. 996–1002, 1999.

[26] Kearns, M. and Singh, S., "Near-optimal reinforcement learning in polynomial time," *Machine Learning*, vol. 49, pp. 209–232, 2002.

[27] Kearns, M. J. and Vazirani, U. V., *An Introduction to Computational Learning Theory.* Cambridge, MA: The MIT Press, 1994.

[28] Kim, S.-H. and Nelson, B. L., "Selecting the best system," tech. rep., School of Industrial & Systems Eng., Georgia Tech, 2004.

[29] Mitchell, T. M., *Machine Learning.* McGraw Hill, 1997.

[30] Niño–Mora, J., "Stochastic scheduling," in *Encyclopedia of Optimization* (Floudas, C. A. and Pardalos, P. M., eds.), pp. 367–372, Kluwer, 2001.

[31] Papadimitriou, C. H., "Games against nature," *Journal of Computer and System Sciences*, vol. 31, pp. 288–301, 1985.

[32] Papadimitriou, C. H., *Computational complexity.* Addison-Wesley, 1994.

[33] Pinedo, M., *Scheduling: Theory, Algorithms and Systems (2nd ed.).* Upper Saddle River, NJ: Prentice Hall, 2002.

[34] Pitman, J., *Probability.* N.Y.: Springer-Verlag, 1993.

[35] Reveliotis, S. A., "Uncertainty management in optimal disassembly planning through learning-based strategies," in *Proceedings of the NSF–IEEE–ORSI Intl. Workshop on IT-enabled Manufacturing, Logistics and Supply Chain Management*, pp. –, NSF/IEEE/ORSI, 2003.

[36] Reveliotis, S. A., "Modelling and controlling uncertainty in optimal disassembly planning through reinforcement learning," in *IEEE Intl. Conf. on Robotics & Automation*, pp. –, IEEE, 2004.

[37] Reveliotis, S. A., "Uncertainty management in optimal disassembly planning through learning-based strategies," *IIE Trans.*, vol. 39, pp. 645–658, 2007.

[38] Ross, S. M., *Stochastic Processes.* NY: Wiley and Sons, 1996.

[39] Shao, Q., "A comparison theorem on moment inequalities between negatively associated and independent random variables.," *J. Theor. Probab.*, vol. 13, pp. 343–356, 2000.

[40] Sutton, R. S. and Barto, A. G., *Reinforcement Learning.* Cambridge, MA: MIT Press, 2000.

[41] Thrun, S., Burgard, W., and Fox, D., *Probabilistic Robotics.* Cambridge, MA: The MIT Press, 2005.

[42] Watkins, C., *Learning from Delayed Rewards.* PhD thesis, Cambridge University, Cambridge, UK, 1989.

[43] Winston, W. L., *Introduction To Mathematical Programming: Applications and Algorithms, 2nd ed.* Belmont, CA: Duxbury Press, 1995.

VITA

Theologos Bountourelis was born in Volos, Greece, on September 10, 1979. He received a B.S. in Mathematics from the Aristotle University of Thessaloniki, Greece, in 2001, and a M.S. in Operations Research from the Georgia Institute of Technology, in 2004. His research interests, are in the area of Stochastic control theory, Machine Learning theory and its applications in various technological contexts.