Q-learning for Automated Personnel Scheduling

submitted in partial fulfillment for the degree of master of science

Ben Platten
13408410

master information studies
Information Studies: Data Science
faculty of science
university of Amsterdam

fill in your date of defence in format yyyy-mm-dd

	1st Examiner	**2nd Examiner**
Title, Name	title and name of 1st Examiner	title and name of 2nd Examiner
Affiliation	affiliations of 1st Examiner	affiliations of 2nd Examiner
Email	email of 1st Examiner	email of 2nd Examiner

Q-learning for Automated Personnel Scheduling

Ben Platten
University of Amsterdam
Amsterdam, The Netherlands
ben.platten@student.uva.nl

ABSTRACT

This research will investigate the use of Q-learning for automated personnel scheduling. Automated personnel scheduling is computationally complex and ethically sensitive. Exact, heuristic, and hybrid methods are complex and / or don't generalise well when problem formulations change. It is difficult to make a qualitative comparison between techniques because of high variance in constraints, objectives, and weights used in different studies. The data set and problem formulation from The Second International Nurse Rostering Competition are ideal for bench-marking a Q-learning scheduler against other techniques. The scheduler will also be tested in a Randstad problem setting characterised by high numbers of employees per schedule.

KEYWORDS

Q-learning, automated personnel scheduling

1 INTRODUCTION

Matching personnel with shifts in a way that ensures coverage, legality and employee well-being is a task that becomes exponentially more complicated as scale increases [6].

Randstad Groep is a Dutch human resource consulting firm responsible for scheduling over 63,000 employees into over 170,000 shifts every week. Employees are matched with work from several different industries such as retail, healthcare, and catering. Staff shortages and increased demand have made this process even more challenging in recent years.

Solving personnel scheduling with computers is a combinatorial optimisation (CO) problem generally known as the Nurse Rostering Problem (NRP). CO is the process of searching for an optimal solution amongst a finite set of possibilities [10]. CO problems generally carry hard and soft constraints; hard constraints must be met and soft constraint violations should be avoided. A single violation of a hard constraint renders the solution infeasible, whereas the degree of soft constraint violations reflects the quality of a solution. CO is mostly concerned with cases where an exhaustive search (trying every possible combination) for a solution is not possible, and instead algorithms are used to quickly rule out large parts of the search space. Classic problems in CO are the Travelling Salesman - where the objective is to find the shortest route through a list of cities - and the knapsack, where the objective is to maximise the value of objects in a knapsack without violating a weight constraint.

NRP originally focused on the assignment of nurses to shifts in a 24 hour cycle and has been studied since the 1960s by researchers from computer science, mathematics, operations research and medicine [4]. A typical hard constraint in NRPs is that there is sufficient staff coverage at all times. A soft constraint could be an employee's preference not to work on weekends.

For Randstad, personnel scheduling is characterised by high volumes - as many as 600 employees to schedule in one week for a single client - and high variance between clients in terms of problem requirements such as planning horizon, shift types, and requisite skills. Due to staff shortages, coverage is considered a soft constraint; a schedule is acceptable if it includes more or less employees than requested.

Exhaustive search - trying every possible combination of nurses and shifts - is not a viable solution for NRPs because of computational complexity [6]. Instead, algorithms are used and the most common approaches seen in the literature can be categorised as exact methods, heuristic methods, and hybrid methods [4]. Exact methods are mathematical, linear programming based approaches that dominated the early history of NRP. The modelling process can be very challenging and they become unfeasible as the search space increases [4]. Heuristic methods are algorithms that are not guaranteed to terminate with an optimal solution [4]. Heuristic methods rely on handcrafted heuristics that guide their search procedures for specific problems but don't generalise well to new ones.

Reinforcement learning (RL) algorithms - in which learning agents improve decision making through interactions with an environment - have been shown to learn the rules of complex systems without human input and to then generalise well to new problems [13]. RL techniques are not widely used for CO problems but there are cases that have had success.

[8] outperformed manual nurse schedules outperformed manual schedules with a Q-learning based scheduler.

Whether RL methods generalise to new optimisation problems better than exact, heuristic or hybrid methods is yet to be tested. The numerous differences in the problem formulation (constraints, objectives, and weights) seen in the NRP literature has led to a gap in the research for a qualitative comparison of solution methods [14].

A standardised data set and problem formulation is provided by Second International Nurse Rostering Competition. A simulator framework generates test problems from combinations of nurses (35,70, or 110) and weeks (4 or 8) with a small but representative range of constraints. An evaluator determines the validity and cost (in terms of constraint violations) of a schedule. The results of the best performing teams at INRC-II are public so can be used for qualitative comparison.

This paper will answer the following research question:

To what extent can Q-learning optimise the cost function of Nurse Rostering problems generated from the INRC-II data?

sub-questions:

(1) How does performance of a Q-learning scheduler compare to the to the top performing methods used at INRC-II?
(2) How is performance of a Q-learning scheduler affected by variance in the planning horizon of test problems?

(3) How is performance of a Q-learning scheduler affected by variance in the number of nurses in test problems?
(4) How long does a Q-learning scheduler take to train?
(5) How well does a Q-learning scheduler generalise to a Randstad data set?
(6) How does performance of a Q-learning scheduler on Randstad data compared to a manual schedule?

2 LITERATURE REVIEW

In computer science, Computational Complexity Theory groups problems together based on their difficulty [9]. If the number of steps required to solve a problem grows by the size of the input raised to a fixed power, it is in class P. P stands for polynomial and refers to the problem being solvable in polynomial time. In practical terms, this means that the problem can be solved by a regular computer (although it may take a while) [1]. If a problem takes an exponential amount of time to solve (in other words, if no solution has yet been found for solving these problems in polynomial time), but a given solution can be easily verified (in polynomial time), then the problem is in class NP [1]. It is important to note that there are more difficult classes of problems than even the hardest NP problems, for example those where it is hard to verify whether a proposed solution is indeed feasible [10]. A good example of an NP problem is Sudoku; trying every possible combination of numbers will take an exponential amount of time, but verifying that the rows and columns contain values 1-9 is simple. There are two additional sub-categories of NP; NP-hard and NP-complete. The definitions of these sub-categories are both highly technical and difficult to explain succinctly. NP-complete represents the hardest problems in NP. If we could find efficient solutions for an NP-complete problem, we could also find the efficient solutions of every other problem in NP [9]. NP-hard problems are at least as hard as the hardest problems in NP but do not have to be in NP; NP problems are decision problems with yes/no outputs, whereas NP-hard problems do not have this requirement [9]. This is complicated by the fact that optimisation problems, such as NRPs, can be reformulated as decision problems.

The complexity of a NRP is determined by the combination of constraints and parameters [6]. Usually, problems with a large number of possible shift types, a large number of nurses and/or a long planning period, are expected to require more computational effort. Some simplified test problems, or isolated sub-problems can be solved in polynomial time. But most real-world examples and test problems are considered NP-hard or NP-complete. For our purposes, it is sufficient to say that NRP is a complex problem and we will refer to it as NP-hard, as it is most often described in the literature.

Some example problems include more than 20 different shift types, more than 20 different skills and a planning period of up to 13 weeks [3] The objective function of a NRP often includes one, or a combination, of the following objectives: minimising the number of constraint violations, minimising the number of nurses, minimising overtime, maximising the coverage, maximising satisfaction of personal preferences [14]. A significant proportion of the complexity of a NRP can be attributed to the personalised aspect of the schedules: the same schedule will have very different costs if given to one nurse or another, if they have different contracts or preferences on days off [6].

The most common methods - exact, heuristic, and hybrid - have different ways of dealing with the complexity of NRPs.

In their comprehensive review, Burke et al [4], declared that exact methods "cannot cope with the enormous search spaces that are represented by real problems (at least on their own)". However, clever modelling techniques have been used to formulate integer linear programs (ILPs) with huge numbers of variables. [11] used branch-and-price, a technique that augments linear relaxation - a method for solving hard ILPs by temporarily relaxing the integer constraints - by dropping a proportion of the constraints and then checking if the subsequent solution is feasible. Modelling complex ILPs is very challenging and powerful solvers are needed to run them [10].

The most common form of heuristic optimisation is meta-heuristics, which use strategies inspired by other systems to guide the search process. For example, Ant colony optimisation meta-heuristic algorithms build solutions by mimicking the foraging behaviour of ants [7]. Although effective and simpler to implement than exact methods, if the problem statement changes slightly, heuristics need to be revised [4].

Methods that hybridise exact methods with heuristic approaches are known as hybrid approaches and [4] went so far as to say that "some kind of (hybrid) heuristic method offers the only realistic way of tackling such difficult and challenging problems in the foreseeable future."

In general it seems that models seen in the literature are complex, narrowly applicable to a certain environment, and lack generality, but it is difficult draw conclusions when there is so much variance between studies.

There have been attempts to provide a representative and standardised problem formulation and data set for NRP researchers; the International Nurse Rostering Competition (INRC) and the Second INRC (INRC-II) invited researchers to compete to build the best performing scheduler using any method of their choosing [5]. In both competitions the problem formulation was based on the academic literature, with the aim of balancing simplicity with representation of real cases. INRC-II replaced the static planning horizon of previous competition with a multi-stage approach, to better reflect real-world setting. This means that solvers are required to schedule up to 8 consecutive weeks, passing information from one week to the other in order to compute constraint violations properly. The 1st and 2nd placed teams at INRC-II used exact methods, 3rd place used heuristics.

2.1 Related work: Reinforcement Learning

The most widely used ML methods are a class of algorithms known as supervised learning. These algorithms are not applicable to most combinatorial optimization problems because they require access to well-curated, labeled training data [2]. One of the key strengths of RL, another form of Machine Learning (ML), is that agents can learn the rules of an environment without the need for explicit programming or examples of optimal endpoints.

RL can be neatly summarised as *the science of decision making* and refers to a computational approach to learning from interaction [10].

A combination of the state of the environment and an available action, a state-action pair (s, a), has an expected return known as a state-action value function, or a Q-value. A policy is a conditional distribution that the agent uses to select actions. The main aim of RL is to find an optimal policy such that the state-action value function is optimized

Two important categories of RL are those which learn by optimising the value function and those which directly optimise the policy [12]. The Q-learning algorithm finds the optimal policy by iteratively learning the optimal Q-values for state action pairs and is considered one of the most important breakthroughs in reinforcement learning. In anything other than small toy problems or simple games, it is impractical to compute optimal Q-values for all possible state-action pairs, so, for problems with a larger state space the Q-values are approximated. A well-known approach of value function approximation is Deep Q-Learning (DQN) which using a neural network for function approximation [15].

[8] outperformed manual nurse schedules on a custom Fairness Indicator Score by applying Q-learning to NRP. Their fairness score, which combines equality, preference and Korean employment law, is an innovative way to make ethical considerations part of the primary objective. Q-learning is a RL algorithm that seeks to find the best action to take given the current state [].

Instead of learning the values associated with the state-action pairs, Policy gradient methods learn a parameterised policy that maximizes reward [12].

[2] achieved close to optimal results on another NP-hard problem, the knapsack problem, using a form of policy gradient RL to optimise the parameters of a neural network. Their solution is computationally expensive, but requires little engineering or heuristic design.

3 METHODOLOGY

This section describes the methodology used to answer the proposed research questions. The first section describes the INRC-II problem formulation and data set in detail. In the second section we present the detailed framework and algorithm of the proposed Q-learning scheduler and explain why it is the chosen technique. The final section describes the Randstad planning data and how it will be used to create representative test problems.

3.1 INRC-II problem formulation and data set

INRC-II problem formulation uses a multi-stage planning horizon meaning that a schedule has to consider data from the preivous week. There are 3 separate input sources:

- scenario: information global to all weeks (nurse contracts, shift types, skills)
- week-data: specific data of a single week (daily coverage requirements, nurse preferences)
- history: information that must be passed from one week to another to compute constraint violations

  border data: for keeping track of consecutive days on/off, last day of previous week

  counters: cumulative values over the weeks; total shifts, total worked weekends

A command-line simulation/validation software is used to simulate the solution process and to evaluate the quality of the solver A validator is also provided which checks for the validity of a solution of an instance, and calculates the corresponding objective function value, according to the constraints' weights.

3.2 Q-learning

Q-learning has been chosen as the RL algorithm for this research beacuse it is conceptually simpler and has been shown to converge faster than policy gradient methods [15]/

We will use the Q-learning algorithm to create a scheduler that takes uses input data to derive a schedule in which the sum of Reward converges to the maximum.

3.3 High volume planning problems

Using real world data from Randstad, some test problems will be created involve large numbers of employees. We will test a Q-learning schedulers performance on these problems.

4 RISK ASSESSMENT

- Offline Q-learning is difficult; there is no guarantee that a model will converge on acceptable schedules in the time available. Furthermore, only 1 paper that applies Q-learning to NRP has been found.
- It will be challenge to train a model on two different datasets. Particularly as the INRC-II data set and problem formulation is complicated and the Randstad data set needs to be created. The Randstad data requires significant domain knowledge and pre-processing to be used.
- The INRC II is now 7 years old, the simulator and evaluator may be depreciated.

5 PROJECT PLAN

The project starts on the 4th of January and the deadline is the 30th of June. The project will take 20-hours per week. Moreover, I will have weekly meetings with my supervisors.

Week	date	task
1-3	01.03 - 01.23	literature review and problem understanding
4-7	01.24 - 02.18	interview with domain experts. finish thesis design
8-9	02.21 - 03.06	EDA
10-11	03.07 - 03.20	begin experimenting with q-learning
12-13	03.21 - 04.03	develop model on on INRC data
14	04.04 - 04.10	mid-point check-in with supervisors
15-17	04.11 - 05.01	develop model on on Randstad data
18	05.02 - 05.08	evaluate results
19-21	05.09 - 05.29	Continue overall evaluation and start writing thesis
22-24	05.30 - 06.12	Send first thesis design version to supervisors. Integrate feedback.
19-21	06.13 - 06.30	final submission.

REFERENCES

[1] [n.d.]. The odds that P=NP is 3% | Scott Aaronson and Lex Fridman. https://www.youtube.com/watch?v=8h0_yaSRwDM

[2] Irwan Bello, Hieu Pham, Quoc V. Le, Mohammad Norouzi, and Samy Bengio. [n.d.]. Neural Combinatorial Optimization with Reinforcement Learning. ([n. d.]). arXiv:1611.09940 http://arxiv.org/abs/1611.09940

[3] Burak Bilgin, Patrick De Causmaecker, Benoît Rossie, and Greet Vanden Berghe. [n.d.]. Local search neighbourhoods for dealing with a novel nurse rostering model. 194, 1 ([n. d.]), 33–57. https://doi.org/10.1007/s10479-010-0804-0

[4] Edmund K. Burke, Patrick De Causmaecker, Greet Vanden Berghe, and Hendrik Van Landeghem. [n.d.]. The State of the Art of Nurse Rostering. 7, 6 ([n. d.]), 441–499. https://doi.org/10.1023/B:JOSH.0000046076.75950.0b

[5] Sara Ceschia, Nguyen Dang, Patrick De Causmaecker, Stefaan Haspeslagh, and Andrea Schaerf. [n.d.]. The Second International Nurse Rostering Competition. 274, 1 ([n. d.]), 171–186. https://doi.org/10.1007/s10479-018-2816-0

[6] S. J. M. den Hartog. [n.d.]. *On the Complexity of Nurse Scheduling Problems.* http://dspace.library.uu.nl/handle/1874/330858 Accepted: 2016-05-09T17:00:27Z.

[7] Ghaith M. Jaradat, Anas Al-Badareen, Masri Ayob, Mutasem Al-Smadi, Ibrahim Al-Marashdeh, Mahmoud Ash-Shuqran, and Eyas Al-Odat. [n.d.]. Hybrid Elitist-Ant System for Nurse-Rostering Problem. 31, 3 ([n. d.]), 378–384. https://doi.org/10.1016/j.jksuci.2018.02.009

[8] In-Chul JUNG, Yeun-Su KIM, Sae-Ran IM, and Chun-Hwa IHM. [n.d.]. A Development of Nurse Scheduling Model Based on Q-Learning Algorithm. 9, 1 ([n. d.]), 1–7. https://doi.org/10.24225/KJAI.2021.9.1.1

[9] Antonina Kolokolova. [n.d.]. CS 6901 (Applied Algorithms) – Lecture 3. ([n. d.]), 5.

[10] Ger M. Koole. [n.d.]. *An introduction to business analytics.* MG books.

[11] Antoine Legrain, Jérémy Omer, and Samuel Rosat. [n.d.]. A rotation-based branch-and-price approach for the nurse scheduling problem. 12, 3 ([n. d.]), 417–450. https://doi.org/10.1007/s12532-019-00172-4 Publisher: Springer.

[12] Yuxi Li. [n.d.]. Deep Reinforcement Learning: An Overview. ([n. d.]). arXiv:1701.07274 http://arxiv.org/abs/1701.07274

[13] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. [n.d.]. Human-level control through deep reinforcement learning. 518, 7540 ([n. d.]), 529–533. https://doi.org/10.1038/nature14236

[14] Sanja Petrovic and Greet Vanden Berghe. [n.d.]. A comparison of two approaches to nurse rostering problems. 194, 1 ([n. d.]), 365–384. https://doi.org/10.1007/s10479-010-0808-9

[15] Richard S Sutton and Andrew G Barto. [n.d.]. Reinforcement Learning: An Introduction. ([n. d.]), 352.

Appendix A TITLE OF YOUR APPENDIX

Put your appendix here.