THEORY AND PRACTICE OF LARGE-SCALE LOGISTICS: OFFLINE CONTEXTUAL BANDITS AND DECOMPOSITION METHODS

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Samuel Tan

December 2025

THEORY AND PRACTICE OF LARGE-SCALE LOGISTICS: OFFLINE CONTEXTUAL BANDITS AND DECOMPOSITION METHODS

Samuel Tan, Ph.D.

Cornell University 2025

This dissertation addresses two important problems in large-scale logistics, motivated by first-hand experience in industry and military settings. The first part focuses on offline contextual bandits, drawing from work at Uber on driver incentive programs. I introduce Empirical Soft Regret (ESR), a novel loss function for value-based learning that addresses limitations of accuracy-based approaches in misspecified settings. Unlike standard methods that fail when reward models are poorly specified, ESR provably yields policies that asymptotically achieve optimal performance while remaining compatible with gradient-based optimization. The value of this approach is demonstrated through applications in health datasets, news recommendation, and computational materials science.

The second part addresses large-scale logistics involving simultaneous routing and scheduling of commodity deliveries across intermodal networks. In collaboration with the United States Marine Corps and Navy, I develop a mixed-integer programming formulation for expeditionary warfare logistics that captures the various physical constraints placed on the network. To address computational limitations, I propose an efficient solution method based on dual decomposition that leverages Lagrangian duality to split the problem into smaller, computationally tractable subproblems.

This work bridges the gap between operations research theory and practice, demonstrating how theoretical foundations can be successfully translated into prac-

tical solutions for complex real-world logistics challenges.

BIOGRAPHICAL SKETCH

Samuel Tan was born and raised in Seattle, Washington. He received a Bachelor of Science degree in 2020 from Harvey Mudd College, where he jointly majored in Computer Science and Mathematics. His first experience with operations research was in undergraduate class, where he learned how useful math can be in making real-world decisions. It was that realization that started his journey to become a PhD student in the field. In his time at Cornell, he has been involved in several projects that have direct real-world impact, including working on Cornell's COVID modeling team, multiple internships at Uber, and a collaboration with the Marine Corps and Navy on optimizing their logistics.

After graduating, Sam will move to San Francisco where he will return to work at Uber full-time in the Driver Trip Pricing team.

This thesis is dedicated to my parents, Aiguo and Yanping.

ACKNOWLEDGEMENTS

I would first like to thank my advisor Peter Frazier for all of the guidance and advice over the past five years. I can think of no other human being on the planet who could have helped me grow so much as a researcher, mathematician, statistician, student, teacher, software engineer, data scientist, and communicator (among other things). His insightful questions, willingness to both teach and learn, and ability to draw connections are all beacons towards which I strive as I progress along my career.

I would also like to thank Nathan and Shane for serving on my committee and providing valuable feedback.

For the past three years, I have been fortunate to work with a wonderful group of collaborators as part of the Office of Naval Research grant. I would especially like to thank Haden Boone, Mathieu Dahan, Matthew Ford, Jeff Linderoth, Yongjia Song, and Huseyin Topaloglu for all of the experience and skills I have gained from working with you.

Even before coming here, I heard about how friendly the ORIE department is at Cornell and, in that respect, it did not disappoint. From faculty to students to staff, I have always had wonderful interactions and will miss the friendly atmosphere we share here. I want to especially thank my PhD cohort for making Ithaca feel like a home, particularly during the first few semesters in the peak of Covid. In particular, I want to thank Tao Jiang, Zhi Liu, Laurel Newman, Ricky Shapley, and Tyler Sam for all of the conversations and good food we have shared over the years; I will cherish our memories together.

To Jonah Botvinick-Greenhouse, Ashira Mawji, and Robin Armstrong, thank you for being wonderful roommates who have provided so much moral support the past two years.

To Jess Wu, thank you for being my emotional rock. You were there to celebrate my highs and to pick me up from my lows. Words cannot describe how much I appreciate all you have done.

Finally, I reserve my deepest gratitude for my family, without whom none of this would be possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Large-scale logistics are an integral part of modern society. Commonly used technologies such as online marketplaces, ridesharing and food delivery apps, and commercial airlines all rely on coordinating the movement of things or people across space and time. Although simple in concept, deploying these technologies in the real-world faces many technical challenges. Questions like how trips on ridesharing apps should be priced, or which optimization algorithm to use for real-time routing decisions require the expertise of operations research experts to answer. For such settings, it is important to develop theoretically grounded algorithms, as they provide performance guarantees and help decision-makers deploy solutions with confidence at scale.

To that end, this dissertation addresses two problems that I have faced in applied work over the past five years, both in large-scale logistics. The first came from a series of internships I did at Uber, where I worked on a team that allocated money to different incentive programs for drivers. The research question here was how to leverage machine learning to intelligently solve a budget allocation problem, given that each driver is associated with historically collected features. This is the basis for Part I of this dissertation. The second came from an applied project with the United States Marine Corps and Navy, where the goal was to develop a tool that assists logisticians in making decisions about scheduling and routing vehicles and commodities in areas of contestation. The key to addressing this challenge was understanding what the right model is and which algorithm to use, given the immense size of the naval logistics network. Part II addresses this problem.

In Part I, I study algorithms and applications for offline contextual bandits

and loss functions design. I first discuss my work on loss functions for offline contextual bandits in Chapter 2. Contextual bandits model sequential decision-making problems in which at each time step, an agent observes a context, takes a decision, and receives a reward. The goal of the agent is to maximize this reward over time by deriving a good *policy*, a function which maps from context to action. The offline variant of contextual bandits considers the case that the agent has a static dataset of historical context-action-reward tuples, and must still optimize the reward obtained in expectation over the distribution of contexts.

Offline contextual bandits model many important real-world problems, including personalized medicine [Ameko et al., 2020], news recommendation [Li et al., 2010], and online advertising [Chaudhuri et al., 2017]. I study value-based learning, a kind of learning algorithm which relies on learning a reward model as a function of the context and the reward. Standard value-based learning relies on accuracy-based loss functions for training the reward model, which only work well when the reward model is well-specified. In contrast, I study the mis-specified setting, in which case training on accuracy-based loss function fail to yield good policies. Instead, I propose a novel loss function, Empirical Soft Regret (ESR), which can provably yield a policy which asymptotically achieves the best performance in a model class. Importantly, this loss function can still be used with any machine learning model which relies on gradient-based optimization, such as neural networks. Beyond theory, I also demonstrate its effectiveness on a health dataset and a dataset on news recommendation.

In Chapter 3, I explore better loss functions for neural interatomic potentials in computational materials science. I first introduce a novel approach for improving minimum energy pathway (MEP) prediction by combining neural interatomic

potentials with ideas from offline contextual bandits. I proposed adapting the ESR loss to learning an energy model which optimizes for the downstream task of identifying low-energy reaction pathways. Experiments on the oxygen reduction reaction demonstrate that ESR enables accurate MEP predictions using significantly fewer DFT evaluations than traditional methods, outperforming standard baselines and showing promise for scalable materials discovery. I also introduce a new loss function to use when predicting physical quantities that depend on the distribution of energies of a material. Through the lens of statistical mechanics, I design a loss function, Canonical Ensemble Loss, which targets the Boltzmann distribution of energies, a property of the canonical ensemble of the material. Using a synthetic dataset of magnesium-iron oxide rock salts, preliminary results show promising performance on the task of predicting the critical temperature.

In Part II, I study large-scale logistics problems which involve the simultaneous routing and scheduling of commodity deliveries across an intermodal delivery network. Chapter 4 introduces a mixed-integer programming (MIP) formulation of this problem for expeditionary warfare involving the United States Marine Corps (USMC). In contested areas, the USMC may need to quickly deploy people, equipment, and sustenance to various locations across a wide network. The USMC has at its disposal many modes of transportation, such as air, sea, and ground, and many different vehicles for each mode with their own speeds, ranges, and capacities.

Currently, Marine logisticians plan the delivery of the commodities manually, using a combination of spreadsheets and presentation slides. We instead model this problem using a MIP, where the objective is to minimize the late and unmet demand, and the constraints capture the physical constraints of the logistics problem, such as vehicle speeds, ranges, capacities, etc. We demonstrate the efficacy of

our approach on a fictitious warfare scenario set in Southern California designed to reflect USMC training scenarios. Our approach is also currently in active use by the USMC in training exercises.

In Chapter 5, I discuss an efficient way of solving the same kinds of large-scale logistics problems as in the previous chapter. In practice, the MIP formulation of Chapter 4 can be solved by state-of-the-art MIP solvers as long as the size of the problem, as measured by the number of variables and constraints, can fit on a reasonable computer. However, once the problem becomes too large, memory and/or time requirements become practically infeasible, and an alternative solution method becomes necessary. I propose a method based on dual decomposition [Everett III, 1963] which leverages Lagrangian duality to split the MIP into many smaller problems, each of which require much less memory and time to solve. By solving these smaller problems in a loop, one can derive a solution for the overall MIP. I demonstrate this method on the same scenario in the previous chapter, showing that this method can yield solutions on large problems where the MIP cannot due to the compute required.

Part I

Offline Contextual Bandits and Loss Function Design

CHAPTER 2

VALUE-BASED LEARNING IN OFFLINE CONTEXTUAL BANDITS

This chapter is based on joint work with Peter Frazier.

2.1 Introduction

We study offline contextual bandits. Here, an agent takes an action a when presented with a context x drawn from a distribution $P_{\mathcal{X}}$ and receives a reward $r(x, a)$. The agent's goal is to learn a policy that maximizes the expected reward over the distribution of contexts, using only a fixed dataset of past interactions, i.e. without active exploration. Offline contextual bandits have many real-world applications such as personalized medicine [Ameko et al., 2020], news recommenders [Li et al., 2010], and online advertising [Chaudhuri et al., 2017].

Value-based learning first trains a reward model that minimizes an accuracy-based loss function on the observed rewards, then outputs the greedy policy with respect to this reward model [Beygelzimer and Langford, 2009]. Value-based learning is important to study because it is widely used [Rafieian and Yoganarasimhan, 2021, Huang et al., 2022, Sakhi, 2023] due to its ability to accommodate a wide range of supervised regression models such as random forests, convolutional neural networks, and transformers. Other advantages of value-based learning reported by the literature include its superior generalization behavior when using overparameterized models [Brandfonbrener et al., 2021] and its low variance vs. methods based on inverse propensity weights [Dudík et al., 2014].

A major challenge in deploying value-based learning is model misspecification,

that is, when the true reward model that generated the data is outside the class of models being used to fit the data. When the model is misspecified, the induced policy will not typically achieve zero regret. We provide a simple example showing that value-based learning fails to achieve asymptotically zero excess regret. More flexible models, such as neural networks with massive architectures, can reduce model misspecification but at the cost of higher inference times, hardware cost, memory use, and less interpretability.

While we should not expect to achieve zero regret when the model is misspecified, we can hope to achieve zero *excess* regret, defined as the additional regret incurred by a policy relative to the minimum regret achievable in a model class.

Hu et al. [2024] study this setting and show that a general version of the direct method, i.e. value-based learning, achieves zero excess regret asymptotically in the data size n. In particular, under the assumption that the MSE of the reward model goes to 0 as n grows, they provide a loss function, the IERM objective, which they theoretically shows yields asymptotically optimal regret over the policy class.

Under this framework, we provide a specific nearest-neighbor/matching estimator for the reward which satisfies the reward model assumption, as well as a softmax policy class which is suitable for gradient-based optimization. Moreover, by applying results from Hu et al. [2024] as well as independently derived theory, we show that the minimizer of this loss function, Empirical Soft Regret (ESR), enjoys asymptotically optimal regret in the policy class as the number of datapoints n grows to infinity.

The rest of the chapter is structured as follows. Section 2.2 overviews the literature. Section 2.3 presents our mathematical setting. Section 2.4 analyzes the

shortcomings of value-based learning. Section 2.5 presents our proposed loss function and Section 2.6 connects our work to Hu et al. [2024]. Section 2.7 goes over our independently derived theoretical results. Section 2.8 demonstrates the effectiveness of our method on experiments in both conditional average treatment effect estimation and contextual bandits. Finally, Section 2.9 concludes and outlines future work.

2.2 Related Work

Here we overview related work in offline contextual bandits. We also review approaches from other settings that are closely related to value-based learning.

2.2.1 Policy Learning in Offline Contextual Bandits

Algorithms for policy learning in offline contextual bandits are generally either value-based or policy-based. In value-based learning, the contextual bandit algorithm learns a reward model $\hat{r}(x, a)$ that predicts $r(x, a)$ from x and a, and then executes the greedy policy with respect to this reward model. In context x, this policy takes action $\pi(x) = \text{argmax}_a \hat{r}(x, a)$ [Beygelzimer and Langford, 2009]. Within the space of value-based methods, the "direct method" learns the reward model by regressing the observed r_i on (x_i, a_i) using the MSE loss function, i.e., by minimizing $\frac{1}{n}\sum_{i=1}^{n}(r(x_i, a_i) - \hat{r}(x_i, a_i))^2$. Other value-based learning methods incorporate pessimism [Nguyen-Tang et al., 2021] and distributional robustness [Yang et al., 2023].

In contrast, policy-based algorithms optimize over the space of stochastic poli-

cies $\pi(x)$, which map from x to the d-dimensional simplex (for d actions). The standard policy-based algorithm is the inverse-propensity weight (IPW) method [Rosenbaum and Rubin, 1983], which uses $\frac{1}{n}\sum_{i=1}^{n} r_i \frac{\pi(a_i|x_i)}{\pi_0(a_i|x_i)}$ as the loss function, where $\pi(a|x)$ is the probability of taking action a given context x under policy π, and π_0 is the policy used when collecting the data $\{x_i, a_i, r_i\}$. Many offline policy learning methods build on the IPW method, e.g. to account for propensity overfitting [Joachims et al., 2018, Swaminathan and Joachims, 2015a,c], pessimism [Sakhi et al., 2024, Wang et al., 2024], PAC-Bayesian bounds [Sakhi et al., 2023], propensity score variance [Behnamnia et al., Strehl et al., 2010, Kallus, 2018], limited overlap in context distributions between different actions [Kallus, 2021], tree- and forest-based learning [Kallus, 2017], reduction to weighted classification [Zhao et al., 2012, Zhang et al., 2012, Zhou et al., 2017], or distributional robustness [Si et al., 2023]. A closely related policy-based method is Swaminathan and Joachims [2015b], in which a softmax policy parameterization is also used, enabling gradient-based optimization. One important difference in policy-based learning is that they generally require knowledge of the propensities and/or logging policy as part of the training data $\pi_0(x_i)$.

The doubly robust method [Dudík et al., 2014] combines a reward model $\hat{r}$ and inverse-propensity weights $1/\pi_0$ to output $\hat{\pi} = \operatorname{argmin}_{\pi\in\Pi} \frac{1}{n}\sum_{i=1}^{n} \left(\sum_{a\in\mathcal{A}} \hat{r}(x_i, a)\pi(a|x_i) + \frac{\hat{\pi}(a_i|x_i)}{\pi_0(a_i,x_i)}(r_i - \hat{r}(x_i, a_i)) \right)$. Variants thereof address the high variance of inverse propensities [Su et al., 2020], distributional robustness [Kallus et al., 2022], or use k-fold cross-fitting to yield minimax optimal regret rates [Zhou et al., 2023].

When using overparameterized models, Brandfonbrener et al. [2021] compare value-based and policy-based methods, finding that value-based methods exhibit

the same good generalization performance of overparameterized supervised learning, while policy-based methods do not. They show this is because policy-based learning suffers from an "action-unstable" objective, i.e. one for which the minimizer for each individual datapoint is different from the minimizer over the entire dataset. They establish upper bounds on regret for value-based methods and lower bounds on regret for policy-based methods based on this property.

2.2.2 Predict-then-optimize

Like this work, the literature of predict-then-optimize, or decision-focused learning, explores alternative loss functions when training models for downstream decision-making [Elmachtoub and Grigas, 2022, Wilder et al., 2019, Mandi et al., 2022]. These works, however, assume full feedback, in sharp contrast with offline contextual bandits. In particular, they assume the objective function to be maximized has a known form $f(c, a)$, where a is the decision variable which must lie in some set $\mathcal{A}$ and c is an uncertain parameter. A parametric model $\hat{c}(x)$ is fit to training data $\{(x_i, c_i)\}_{i=1}^n$ by regressing $c \sim x$. Then, given context x, the approach outputs decision $\text{argmax}_{a \in \mathcal{A}} f(\hat{c}(x), a)$. Regret results when prediction error, $\hat{c}(x) \neq c(x)$, causes the value of the best action, $\max_{a \in \mathcal{A}} f(c(x), a)$, to strictly exceed the value of the chosen action, $f(c(x), \text{argmax}_{a \in \mathcal{A}} f(x, \hat{c}(x)))$.

In this model, $f(c, a)$ is known, so that for each datapoint (x_i, c_i), the counterfactual outcome $f(c_i, a)$ is known for each $a \in \mathcal{A}$. This makes maximizing the objective within the training data trivial; simply take $\text{argmax}_{a \in \mathcal{A}} f(c_i, a)$ for each i. In contrast, we only observe the output of the response function for a single action and context, necessitating a regression model $r \sim x, a$ that can predict counterfactual outcomes. Maximizing the outcome in the training data is not immediate.

2.2.3 Contextual Optimization with Bandit Feedback

Hu et al. [2024] study contextual linear optimization with bandit feedback. In this setting, an offline dataset of n data points $\mathcal{D} = \{(X_1, Z_1, C_1), \ldots, (X_n, Z_n, C_n)\}$ is drawn from a distribution on (X, Z, C). Here X_i is the context, Z_i is the action lying in a polytope $\mathcal{Z}$, and C_i is the observed cost which is linear in Z, i.e. $C_i = Z_i^\top Y$ for some Y. Given that $\mathbb{E}[Y|X = x] = f_0(x)$, the goal is to derive a policy $\hat{\pi}(x)$ which incurs low regret, defined as $\mathrm{Reg}(\pi) = \mathbb{E}_X[f_0(X)^\top \pi(X) - \min_{z \in Z} f_0(X)^\top z]$. This setting captures offline contextual bandits when Z is the simplex, i.e. when the vertices of the polytope $\mathcal{Z}$ comprise the canonical basis.

Hu et al. [2024] study three algorithms: the direct method, inverse spectral weighting, and the doubly robust method, which all rely on optimizing an unbiased estimate for the value of a policy $\mathbb{E}_X[f_0(X)^\top \pi(X)]$ over a policy class $\Pi_{\mathcal{F}}$ induced by a model class $\mathcal{F}$ (which may or may not contain f_0). They derive a general bound on the regret achieved by the resulting policy. They show that the regret bound enjoys a fast rate when the induced policy class is well-specified, i.e. $\min_{\pi \in \Pi_{\mathcal{F}}} \mathrm{Reg}(\pi) = 0$, and a slower rate when $\min_{\pi \in \Pi_{\mathcal{F}}} \mathrm{Reg}(\pi) > 0$.

§2.6 analyzes their result for the direct method for offline contextual bandits in more detail.

2.2.4 Causal Machine Learning

The literature on causal machine learning has studied settings with data of the form (x_i, t_i, y_i) where x_i are contextual features, t_i is the binary treatment indicator, and y_i is the outcome, where it is generally assumed only the outcome from one

treatment is observed for each x_i.

Most of this literature considers the problem of treatment effect estimation and quantifying uncertainties in these estimates, not in designing policies with low regret. In particular, many methods in the literature construct estimators for the conditional average treatment effect (CATE), defined as $\mathbb{E}[y(1) - y(0)|x]$, with theoretical guarantees on the quality of the estimator. In particular, there has been much recent interest in causal metalearners, a class of methods which leverage black-box machine learning models in workflows which include custom loss functions, constructing additional training signals, and data splitting [Künzel et al., 2019, Kennedy, 2023, Nie and Wager, 2021].

CATE estimation and policy learning are related, but not the same. When one wishes to achieve low regret, all that matters is that the treatment with the best outcome is also the one with the highest estimated CATE. Beyond this, it does not matter how accurate the estimate is. Moreover, there is a tradeoff between low regret and high CATE accuracy, as with a limited model class, a model which is optimized for CATE would not generally align with that which minimizes regret, and vice versa.

As such, one would expect a method designed to achieve the maximum CATE accuracy possible in the model class to not achieve minimum regret. Comparing our proposed method against four such CATE estimators (S-, T-, R-, and DR-learners), we demonstrate this to be the case empirically in our numerical experiments in §2.8.1.

Analogous to policy-based methods for offline contextual bandits, methods for learning policies directly have been proposed. These include [Kitagawa and

Tetenov, 2018, Athey and Wager, 2021], which are based on similar ideas as the IPW and doubly robust methods from offline contextual bandits. We compare with policy-based benchmarks in our experiments and find that our method achieves lower regret.

2.3 Model

We now define our mathematical model. An agent interacts with a K-armed contextual bandit where in each round t, the agent is presented with a context $x_t \in \mathcal{X} \subset \mathbb{R}^d$ drawn independently from a fixed unknown distribution $P_{\mathcal{X}}$, takes action $a_t \in \mathcal{A}$ where $|\mathcal{A}| = A$, and receives reward $r_t := r(x_t, a_t)$ where $r(\cdot, \cdot)$ is fixed but unknown. The agent would like to choose a_t based on x_t to maximize the expected value of the reward r_t, where the expectation is taken over contexts x_t. The agent is presented with historical data of the form (x_i, a_i, r_i) where $x_i \sim P_{\mathcal{X}}$ and $a_i \sim \pi_0(x_i)$ for some unknown logging policy π_0. They use this data to train a reward model $\hat{r}_\theta(x, a)$, where $\theta \in \Theta$ parameterizes the reward model; this captures, for example, the weights in ordinary least squares, or the weights and biases in a deep neural network. After training, the agent fixes the model and uses it to derive a policy $\pi : X \mapsto \Delta(A)$, a function which takes as input a context x and outputs a probability distribution over actions. In the case of value-based learning, this distribution is defined to be a point mass on the action with the maximum predicted reward, and the notation in this case is $\pi(x) = \text{argmax}_a \, \hat{r}_\theta(x, a)$.

The objective is to minimize expected regret. For model parameters θ and context x, the regret, in the case of value-based learning, is

$$\mathrm{R}_\theta(x) := \max_{a \in \mathcal{A}} r(x, a) - r(x, \operatorname*{argmax}_{a \in \mathcal{A}} \hat{r}_\theta(x, a)),$$

and the expected regret is $\mathbb{E}_{x\sim P_{\mathcal{X}}} R_\theta(x)$. We also define excess regret for a model θ and its model class Θ as $\mathbb{E}_{x\sim P_{\mathcal{X}}} R_\theta(x) - \min_{\theta\in\Theta} \mathbb{E}_{x\sim P_{\mathcal{X}}} R_\theta(x)$.

Note that the expected reward, $\mathbb{E}_{x\sim P_{\mathcal{X}}} r(x, \operatorname{argmax}_{a\in\mathcal{A}} \hat{r}_\theta(x,a))$ differs from (negative) reward by only a constant: $\mathbb{E}_{x\sim P_{\mathcal{X}}}[\max_a r(x,a)]$ and so any maximizer θ of the expected reward would simultaneously minimize regret.

2.4 Analysis of Value-based Learning

In value-based learning (also referred to interchangeably as the direct method), one usually trains a reward model that minimizes the empirical MSE loss: $L_{MSE}(r, \hat{r}_\theta) := \frac{1}{n}\sum_{i=1}^{n} \left(r(x_i,a_i) - \hat{r}_\theta(x_i,a_i)\right)^2$. The policy which derived from this reward model is then $\pi(a|x) = \mathbb{1}\{a = \operatorname{argmax}_{a\in\mathcal{A}} \hat{r}(x,a)\}$, i.e. greedy with respect to the predicted reward model.

MSE is focused on predictive accuracy and does not consider the downstream impact of reward model errors on regret. For example, MSE treats the decision variables a_i and contextual features x_i in the same way, despite their distinct roles in influencing regret.

In contrast, we study the setting that the reward model is misspecified, i.e. $\mathbb{E}_{x\sim P_{\mathcal{X}}} \sum_{a\in\mathcal{A}} (r(x,a) - \hat{r}_{\hat{\theta}_{MSE}}(x,a))^2$ does not vanish even as $n \to \infty$. This is an important setting, as in practice one can rarely ensure the model class used in training contains the true model. In particular, the following counterexample shows that minimizing MSE can yield nonzero excess regret, even as $n \to \infty$, when the reward model is misspecified.

Proposition 2.4.1. *Let $d > c > 0$ be real numbers. Let there be one context $x_0 \in$*

$\mathcal{X}$ and two actions $c, d \in \mathcal{A} \subset \mathbb{R}$. Let $P_{\mathcal{X}}$ be a point mass at x_0, $\pi_0(a|x_0) > 0$ for $a = c, d$, and $r(x, a) = -c - d + a$ so that the data-generating distribution only produces two tuples (in (x, a, r) format) with positive probability: $(x_0, c, -d), (x_0, d, -c)$. Let the model class contain two functions: $\hat{r}_\theta(x, a) = \theta \cdot a$ for $\theta = \pm 1$. Then the MSE-minimizing model does not minimize regret in the model class.

Proof. Consider the MSE for $\theta = +1$. We have $\hat{r}_\theta(0, c) = c$ and $\hat{r}_\theta(0, d) = d$. For both possible datapoints, the squared error is $(c + d)^2$. It immediately follows that the MSE is also $(c + d)^2$. Likewise, for $\theta = -1$, we have $\hat{r}_\theta(0, c) = -c$ and $\hat{r}_\theta(0, d) = -d$. For both datapoints, the squared error equals $(c - d)^2$, and therefore so is the MSE. This is smaller than the MSE for $\theta = +1$. Therefore, the MSE minimizer is $\theta = -1$. However, choosing $\theta = -1$ yields positive regret for all n; the policy which is greedy with respect to $\hat{r}_{-1}$ always chooses action d, incurring (expected) regret $d - c$. On the other hand $\theta = +1$ trivially yields zero regret. □

Next we show how value-based learning can be adapted to address this gap and lower excess regret.

2.5 Empirical Soft Regret

We construct an alternative loss function for training the reward model that directly targets the expected reward attained by the induced policy. In particular, we leverage the insight that the ability of the reward model to accurately predict the true reward is not directly important; we only care about its ability to generate good decisions $\operatorname{argmax}_x \hat{r}_\theta(x, w)$ and achieve low regret.

Ideally, a training method would directly maximize the empirical reward over

θ

$$\frac{1}{n}\sum_{i=1}^{n} r(x_i, \operatorname*{argmax}_{a\in\mathcal{A}} \hat{r}_\theta(x_i, a)). \tag{2.1}$$

However, this is difficult because the only dependence of the loss on the parameter θ is through an argmax, making any gradient-based optimization nontrivial to apply.

To resolve this nondifferentiability, we adapt the softmax policy parameterization from Swaminathan and Joachims [2015b]. In particular, consider the "soft" version which replaces the argmax with a softmax function:

$$\frac{1}{n}\sum_{i=1}^{n}\sum_{a\in\mathcal{A}} r(x_i, a)\frac{e^{k\hat{r}_\theta(x_i,a)}}{\sum_{a'\in\mathcal{A}} e^{k\hat{r}_\theta(x_i,a')}}. \tag{2.2}$$

This loss is smooth with respect to θ, where the level of smoothness is controlled by k, enabling the use of autodifferentation and gradient-based optimization such as in neural networks.

Moreover, we can interpret this as the regret incurred by a stochastic policy $\pi_\theta(\cdot|x)$, where we enforce a softmax parameterization:

$$\pi_\theta(a|x) = \frac{e^{k\hat{r}_\theta(x_i,a)}}{\sum_{a'\in\mathcal{A}} e^{k\hat{r}_\theta(x_i,a')}}.$$

In other words, instead of interpreting $r_\theta(x, a)$ as a prediction for the reward, we view it as the logit in a randomized policy. We can now define the expected regret under a softmax policy:

$$\mathrm{R}^k_\theta(x) := \max_{a\in\mathcal{A}} r(x, a) - \sum_{a\in\mathcal{A}} r(x, a)\frac{e^{k\hat{r}_\theta(x,a)}}{\sum_{a'\in\mathcal{A}} e^{k\hat{r}_\theta(x,a')}}.$$

However, in resolving the nondifferentiability, we encounter a different problem: the data we receive does not typically contain samples of all actions $a \in \mathcal{A}$ for the same context x. Therefore, (2.1) is typically not computable from the data. We now address the challenge that the dataset does not contain all $a \in \mathcal{A}$ for each x_i in the data. First, denote $n_a(i)$ as the index of the nearest neighbor for point i in the set of $\{x_j\}_{j \neq i}$ with action a. Specifically, we define $n_a(i)$ for datapoint i as $n_a(i) \in \text{argmin}_{j \neq i, a_j = a} \|x_i - x_j\|_2$.

Then we define $N(i)$ as the set of $n_a(i)$ for all $a \in \mathcal{A}$ i.e. $N(i) = \{n_a(i) : a \in \mathcal{A}\}$. Then our loss function, called the Empirical Soft Regret (ESR) and written $L_{ESR,k}$, is

$$L_{ESR,k}(r, \hat{r}_\theta) = -\frac{1}{n}\left[\sum_{i=1}^{n} \sum_{j \in N(i)} r(x_j, a_j) \frac{e^{k\hat{r}_\theta(x_i, a_j)}}{\sum_{j' \in N(i)} e^{k\hat{r}_\theta(x_i, a_{j'})}}\right], \tag{2.3}$$

where the negative sign is present since losses are generally to be minimized. The use of the nearest neighbor approximates the loss function that could be computed if we observed rewards for all actions $a \in \mathcal{A}$ on each context x. In training, we minimize this loss function over θ, where again k is a parameter controlling the smoothness of the loss function.

We view our method as a value-based method for offline contextual bandits, as we still learn a reward model $r_\theta(x, a)$. However, unlike the direct method (which uses MSE), our method directly optimizes for the reward attained by the resulting policy, which simultaneously minimizes the incurred regret. As such, we expect our method to have lower regret, even though they may not produce accurate predictions $\hat{r}_\theta(x, a)$ for r. We explore this theoretically below and observe this empirically in our experiments in §2.8.

We will see in §2.8.2 that ESR also outperforms various policy-based methods for offline contextual bandits. We hypothesize that ESR has a particular advan-

tage when the average reward across actions $\frac{1}{|\mathcal{A}|}\sum_{a\in\mathcal{A}} r(x,a)$ is highly variable. Such variability degrades the performance of policy-based method based on inverse propensity weights but ESR's loss function removes most of this variability. We give insight into this phenomenon with a simple example in A.1.3.

2.6 Connection to Hu et al. [2024]

Hu et al. [2024] show that a generalized version of the direct method can achieve asymptotically zero excess regret. In particular, they consider a twofold variation of the direct method that partitions the dataset into two subsets D_1, D_2, each of size $n/2$. D_1 is used to fit the "nuisance" function r. One such method they mention is to minimize MSE within some function class Θ_1 $\hat{\theta}_1 \in \text{argmin}_{\theta\in\Theta}\sum_{i\in D_1}(r_i - \hat{r}_\theta(x_i,a_i))^2$, which would correspond to the traditional value-based learning approach with MSE. They derive an estimate for the reward earned by a policy π as $\frac{2}{n}\sum_{i\in D_2}\sum_{a\in\mathcal{A}}\hat{r}_{\hat{\theta}_1}(x_i,a)\pi(a|x_i)$. Then they switch the roles to analogously construct a second estimate $\frac{2}{n}\sum_{i\in D_1}\sum_{a\in\mathcal{A}}\hat{r}_{\hat{\theta}_2}(x_i,a)\pi(a|x_i)$, and compute

$$\pi_{\hat{\theta}_{DM}} \in \underset{\pi\in\Pi_{\Theta_2}}{\text{argmax}}\,\frac{1}{n}\sum_{j=1}^{2}\sum_{i\in D_j}\sum_{a\in\mathcal{A}}\hat{r}_{\hat{\theta}_{3-j}}(x_i,a)\pi(a|x_i),$$

where Π_{Θ_2} is some policy class parameterized by $\theta \in \Theta_2$.

Remark 2.6.1. Note that in the case that no cross-fitting is used, then the objective is $\text{argmax}_{\pi\in\Pi}\frac{1}{n}\sum_{i=n}\sum_{a\in\mathcal{A}}\hat{r}(x_i,a)\pi(a|x_i)$, where $\hat{r}$ is fit on the entire dataset. In this case, when Π is the A-dimensional simplex, then it is clear the optimal π^* would be the one which is greedy with respect to $\hat{r}$, i.e. $\pi^*(a|x) = \mathbb{1}\{a = \text{argmax}_a\,\hat{r}(x,a)\}$.

This corresponds to what we previously defined as value-based learning. In particular, when $\hat{r}$ is fit by minimizing MSE on the entire dataset, this is commonly referred to as value-based learning [Brandfonbrener et al., 2021] or the direct method (with MSE) [Nguyen-Tang et al., 2021, Sakhi, 2023, Jeunen and Goethals, 2021] in the literature.

Under the assumption that the nuisance estimation error vanishes with n, i.e. $\mathbb{E}_{x\sim P_{\mathcal{X}}} \sum_{a\in\mathcal{A}}(r(x,a) - \hat{r}_{\hat{\theta}_i}(x,a))^2 \overset{n\to\infty}{\to} 0$ for $i = 1, 2$, then

$$\mathbb{E}_{x\sim P_{\mathcal{X}}} R_{\hat{\theta}_{DM}}(x) \overset{n\to\infty}{\to} \min_{\theta\in\Theta} \mathbb{E}_{x\sim P_{\mathcal{X}}} R_\theta(x),$$

i.e. $\hat{\theta}_{DM}$ achieves asymptotically zero excess regret. Hu et al. [2024] mention that one can use a well-specified function class Θ_1 for $\hat{\theta}_1, \hat{\theta}_2$ to guarantee that the nuisance estimation error vanishes while maintaining a simpler function class Θ_2 to compute $\hat{\theta}_{DM}$ (over Π_{Θ_2}).

We now clarify the relationship between ESR and the direct method as described in Hu et al. [2024].

In the absence of cross-fitting, the approach of Hu et al. [2024] is to solve

$$\operatorname*{argmin}_{\theta\in\Theta} \frac{1}{n} \sum_{i=1}^{n} \hat{r}(x,a)\pi_\theta(a|x), \tag{2.4}$$

where $\hat{r}(\cdot)$ is some estimate for the reward $r(\cdot)$ and Θ parameterizes the policy class. We observe that ESR is identical to this IERM objective in the specific case that (1) a matching/1-nearest-neighbor estimate $\hat{r}_{1NN}$ is used for estimating r and (2) a softmax policy class Π_Θ is used, i.e. $\pi_\theta(a|x) \propto \exp(k\hat{r}_\theta(x,a))$.

We now consider applying the results of Hu et al. [2024], where the matching-

based estimator $\hat{r}_{1NN}(\cdot)$ is used for r in the first stage, and we define the policy class as the softmax policy with parameter k given by $\pi_\theta(a|x) \propto \exp(k\hat{r}_\theta(x,a))$.

Denote $\hat{\theta}_n^{IERM}$ the minimizer of the cross-fitted version of 2.4. Then the regret bound achieved in Remark 3 (based on Theorem 1, when $2(1+\alpha)/\alpha$ is integer from Assumption 3) is

$$\begin{aligned}\mathrm{Reg}(\pi_{\hat{\theta}_n^{IERM}}) - \mathrm{Reg}(\pi^*) \leq B(12\Theta\sqrt{\tilde{C}(\alpha,\gamma)}\sqrt{\frac{\eta\log(n+1)+\log(8/\delta)}{n}}^{\frac{2\alpha+2}{\alpha+2}} \\ + \frac{24(\alpha+1)}{\alpha+2}B\Theta\Big(\sqrt{\tilde{C}(\alpha,\gamma)}\left(\frac{\mathrm{Reg}(\pi^*)}{B}\right)^{\frac{\alpha}{2(1+\alpha)}}\sqrt{\frac{\eta\log(n+1)+\log(8/\delta)}{n}} \\ + \frac{\eta\log(n+1)+\log(8/\delta)}{n}\Big) + \frac{2\alpha+2}{\alpha+2}Rate^N(n/2,\delta/4),\end{aligned}$$

where $Rate^N(n,\delta)$ denotes the rate at which $\mathbb{E}_{x\sim P_\mathcal{X}}\sum_a(\hat{r}(x,a)-r(x,a))^2$ converges to 0 (with probability at least $1-\delta$), and $\alpha,\gamma,\eta,\Theta,\tilde{C}$ are various terms required by the assumptions.

In our setting, we consider $\hat{r}$ as the nonparametric 1-nearest neighbor matching-based estimator [Abadie and Imbens, 2006]. When r has zero variance, which is what we consider in this section, then the mean-squared error rate is $O(n^{-1/d})$ with high probability [Abadie and Imbens, 2006]. Therefore the dominant term, as a function of n, in Remark 3 from Hu et al. [2024] is $O(n^{-1/d})$. We will compare our independently derived rate to this in the next section.

2.7 Theory

We now present independently derived theoretical results on the asymptotic excess regret of policies trained on the ESR loss function.

We state the following theorem on the performance of using the ESR loss function in regression settings. We assume a finite action space, i.e. $|\mathcal{A}| = K$, $\mathcal{X} \subset \mathbb{R}^d$, and require the following assumptions.

Assumption 2.7.1 (Regularity conditions on r)**.** There exists finite constants $\Delta, L > 0$ such that ground truth reward r satisfies:

(a) $\|r\|_\infty \leq \Delta$, i.e. the rewards are bounded.

(b) $r(x, a)$ is $L-$Lipschitz in x for each a.

Assumption 2.7.2 (Model class Θ)**.** The class of models r_θ parameterized by $\theta \in \Theta$, denoted r_Θ, satisfies the following for some finite constants L', B, $C > 0$, and $D > 0$:

(a) $r_\theta(x, a)$ is L'-Lipschitz in x for each a and $\|r_\theta\|_\infty \leq B$ for all θ.

(b) The uniform covering number [Anthony and Bartlett, 2009] for r_Θ satisfies $N_\infty(\alpha, r_\Theta, n) \leq \left(\frac{Cn}{\alpha}\right)^D$ for $\alpha \leq 2B$.

From Theorem 14.5 of Anthony and Bartlett [2009], Assumption 2.7.2 (a) and (b) are satisfied for multi-layer, fully connected neural networks with weights bounded in $\|\cdot\|_1$, and Lipschitz, bounded, and nondecreasing activation functions, where D is the number of weights. See A.1.1 for details.

We require the following assumptions on the distribution $P_\mathcal{X}$ from which contexts are drawn.

Assumption 2.7.3 (Distributional assumptions on $P_\mathcal{X}$)**.** The distribution $P_\mathcal{X}$ satisfies:

(a) The support Ω of $P_{\mathcal{X}}$ is bounded. On the support, the density is bounded below by $\underline{p} > 0$.

(b) Let $B_\delta(x)$ be the ball of radius δ around $x \in \mathcal{X}$. There exists $v \in (0, 1]$ such that, for all $x \in \mathcal{X}$ and $\delta > 0$, $\text{Vol}(B_\delta(x) \cap \mathcal{X}) \geq v\text{Vol}(B_\delta(x))$

Both assumptions 2.7.10(a) and 2.7.10(b) are standard in the analysis of nearest-neighbor type estimators [Zhao and Lai, 2020, 2022, Lin et al., 2023]. Assumption 2.7.10(a) is satisfied for a wide class of truncated continuous distributions, e.g. the Gaussian or Laplace distributions truncated at some threshold value. Assumption 2.7.10(b) lower bounds the angle at corner points in the support, avoiding supports which have extremely sharp corners. For example, the boundary of a hyperrectangle on $\mathbb{R}^d$ has $v = 2^{-d}$.

Finally, we assume the following on the logging policy:

Assumption 2.7.4 (Positivity of Logging Policy)**.** There exists a $p > 0$ such that the logging policy π_0 satisfies $\pi_0(a|x) > p$ for all $a \in \mathcal{A}, x \in \mathcal{X}$.

This is a standard assumption to ensure all actions are observed as the number of points goes to infinity; for instance, this is required for the IPW estimator to be unbiased.

Under these assumptions, we have the following theorem on the asymptotic behavior of the regret incurred by the minimizer of the ESR. Define $\mathbb{P}$ as the distribution of the n datapoints $\{x_i\}_{i=1}^n$.

Theorem 2.7.5. *Define θ^k as the minimizer of the expected soft regret within Θ, and define $\hat{\theta}_{ESR,n}$ as the minimizer of the ESR loss function over n datapoints. Suppose that Assumptions 2.7.1 through 2.7.4 hold. Then there exists constants $F, G, H, K, M > 0$ which depend on $\Delta, L, L', A, B, C, \underline{p}, v, p$ such that,*

$$\mathbb{P}\left(|\mathbb{E}_{x\sim P_{\mathcal{X}}} R^k_{\hat{\theta}^{ESR}_{n,k}}(x) - \mathbb{E}_{x\sim P_{\mathcal{X}}} R^k_{\theta_k}(x)| \geq t\right) \leq \max\left(4, F\left(\frac{nk}{t}\right)^D\right) \exp\left(-Gnt^2\right) + Hn \exp\left(\frac{-(n-1)Kt^d}{M^d}\right).$$

Observe that by reversing the bound to be in terms of the probability δ, we get a bound that is $\tilde{O}(n^{-1/d})$ in high probability (where $\tilde{O}$ ignores log terms), which therefore matches the bound from Hu et al. [2024] up to log terms.

Proof. We define the following terms.

$$s_k(\theta) = \mathbb{E}_{x\sim P_{\mathcal{X}}} \sum_{a=1}^{A} r(x,a) \frac{\exp(k\hat{r}_\theta(x,a))}{\sum_{a'=1}^{A} \exp(k\hat{r}_\theta(x,a))},$$

$$\hat{s}_{n,k}(\theta) = \frac{1}{n}\sum_{i=1}^{n}\sum_{a=1}^{A} r(x_i,a) \frac{\exp(k\hat{r}_\theta(x_i,a))}{\sum_{a'=1}^{A} \exp(k\hat{r}_\theta(x_i,a))},$$

and

$$\hat{s}^{ESR}_{n,k}(\theta) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j\in n(i)} r(x_j,a_j) \frac{\exp(k\hat{r}_\theta(x_i,a_j))}{\sum_{a'=1}^{A} \exp(k\hat{r}_\theta(x_i,a))}.$$

We also define

- $\theta_k \in \operatorname{argmin}_\theta s_k(\theta)$
- $\hat{\theta}^k_n \in \operatorname{argmin}_\theta \hat{s}_{n,k}(\theta)$
- $\hat{\theta}^{ESR}_{n,k} \in \operatorname{argmin}_\theta \hat{s}^{ESR}_{n,k}(\theta)$.

Let us prove a high probability bound on $\|s_k - \hat{s}_{n,k}\|_\infty$.

To do so, we first derive the Lipschitz constant for the loss function for an individual context x. We have that

$$\begin{aligned}
&|\sum_{a=1}^{A} r(x,a)\frac{\exp(k\hat{r}_\theta(x,a))}{\sum_{a'=1}^{A}\exp(k\hat{r}_\theta(x,a))} - \sum_{a=1}^{A} r(x,a)\frac{\exp(k\hat{r}_{\theta'}(x,a))}{\sum_{a'=1}^{A}\exp(k\hat{r}_{\theta'}(x,a))}| \\
&\qquad \leq |\sum_{a=1}^{A} r(x,a)\left(\frac{\exp(k\hat{r}_\theta(x,a))}{\sum_{a'=1}^{A}\exp(k\hat{r}_\theta(x,a))} - \frac{\exp(k\hat{r}_{\theta'}(x,a))}{\sum_{a'=1}^{A}\exp(k\hat{r}_{\theta'}(x,a))}\right)| \\
&\qquad \leq \|r(x,\cdot)\|_2 \|\frac{\exp(k\hat{r}_\theta(x,\cdot))}{\sum_{a'=1}^{A}\exp(k\hat{r}_\theta(x,a))} - \frac{\exp(k\hat{r}_{\theta'}(x,\cdot))}{\sum_{a'=1}^{A}\exp(k\hat{r}_{\theta'}(x,a))}\|_2 \\
&\qquad \leq A\Delta L' k\|\theta - \theta'\|_2,
\end{aligned}$$

where the second-to-last inequality is from Cauchy-Schwarz, and the last inequality is a consequence of Assumption 2.7.2 and Proposition 4 of Gao and Pavel [2017].

We now adapt Theorem 17.7 from Anthony and Bartlett [2009]. (The original theorem assumed a codomain of F being $[0,1]$; we expand the codomain of F to $[-b,b]$, as the arguments do not depend on this codomain as long as the Lipschitzness extends to the larger set).

Theorem 2.7.6. *Suppose that F is a set of functions defined on a domain X and mapping into the real interval $[-b,b]$, $Y \subset \mathbb{R}$, and $B \geq 1$. Let P be any probability distribution on $Z = X \times Y$, ϵ be any real number between $0, 1$, and n any positive integer. Then for a loss function $\ell : [-b,b] \times Y \mapsto [0,B]$ satisfying*

$$|\ell(y_1, y_0) - \ell(y_2, y_0)| \leq L|y_1 - y_2|$$

for all y_0, y_1, y_2, then

$$\mathbb{P}\left(\sup_{f\in F}|\mathbb{E}_{x,y\sim P}\ell(f(x),y)-\frac{1}{n}\sum_{i=1}^{n}\ell(f(x_i,y_i)|\geq\epsilon\right)\leq 4N_1\left(\frac{\epsilon}{8L},F,2n\right)\exp\left(\frac{-n\epsilon^2}{32B^2}\right).$$

To use this theorem, we observe from the above arguments that our loss function satisfies the condition of the theorem for $L = A\Delta kL'$.

Then, observing that the N_1 covering number is always smaller than the N_∞ covering number, $|R^k_\theta(x)| \leq \Delta$ for all x, θ, and applying Assumption 2.7.2(b), we plug in r_θ for f and $8A\Delta kL'\alpha = \epsilon$ to yield

$$\|s_k - \hat{s}_{n,k}\|_\infty$$

is larger than $8A\Delta kL'\alpha$ with probability at most $4\left(\frac{Cn}{\alpha}\right)^D \exp(-2A^2L'^2nk^2\alpha^2)$ for $\alpha \leq 2B$.

For $\alpha > 2B$, notice that because $|r_\theta|_\infty \leq B$ by Assumption 2.7.2(a), then the covering number $N_\infty(\alpha, F, 2n) = 1$, since any θ will satisfy $|r_\theta(x) - r_{\theta'}(x)| \leq 2B < \alpha$ for any other θ', for all x. Therefore for $\alpha > 2B$, the bound is simply $4\exp(-2A^2L'^2nk^2\alpha^2)$ for $\alpha \leq 2B$. We can collapse these two cases in an overall bound: $\|s_k - \hat{s}_{n,k}\|_\infty$ is larger than $8A\Delta kL'\alpha$ with probability at most

$$4\max\left(1,\left(\frac{Cn}{\alpha}\right)^D\right)\exp\left(-2A^2L'^2nk^2\alpha^2\right).$$

We then move on to bounding $\|\hat{s}_{n,k} - \hat{s}^{ESR}_{n,k}\|_\infty$.

We proceed directly. We can write $\sup_\theta |\hat{s}_{n,k}(\theta) - \hat{s}^{ESR}_{n,k}(\theta)|$ as

$$
\begin{aligned}
&\sup_\theta |\frac{1}{n}\sum_{i=1}^{n}\left(\sum_{a=1}^{A} r(x_i,a)\frac{\exp(k\hat{r}_\theta(x_i,a))}{\sum_{a'=1}^{A}\exp(k\hat{r}_\theta(x_i,a))} - \sum_{j\in n(i)} r(x_j,a_j)\frac{\exp(k\hat{r}_\theta(x_i,a_j))}{\sum_{a'=1}^{A}\exp(k\hat{r}_\theta(x_i,a))}\right)| \\
&= \sup_\theta |\frac{1}{n}\sum_{i=1}^{n}\sum_{a=1}^{A}(r(x_i,a) - r(x_{j_i(a)},a))\frac{\exp(k\hat{r}_\theta(x_i,a))}{\sum_{a'=1}^{A}\exp(k\hat{r}_\theta(x_i,a))}| \\
&\leq \sup_\theta \frac{1}{n}\sum_{i=1}^{n}\sum_{a=1}^{A}|(r(x_i,a) - r(x_{j_i(a)},a))\frac{\exp(k\hat{r}_\theta(x_i,a))}{\sum_{a'=1}^{A}\exp(k\hat{r}_\theta(x_i,a))}| \\
&\leq \frac{1}{n}\sum_{i=1}^{n}\sum_{a=1}^{A}|(r(x_i,a) - r(x_{j_i(a)},a))| \\
&\leq AL\frac{1}{n}\sum_{i=1}^{n}|x_{j_i(a)} - x_i|
\end{aligned}
$$

where $j_i(a)$ denotes the index of the matched neighbor for datapoint i with action a, the second-to-last inequality follows since each softmax term is positive and bounded above by 1, for any θ, and the last inequality comes from Assumption 2.7.2.

Now let us derive a bound for $\frac{1}{n}\sum_{i=1}^{n}|x_{j_i(a)} - x_i|$. First consider the term for one i, i.e. $|x_{j_i(a)} - x_i|$.

We have the following

$$
\mathbb{P}(|x_i - x_{j_i(a)}| > \epsilon) = \sum_{a_i=1}^{A}\mathbb{P}(|x_i - x_{j_i(a)}| > \epsilon | a_i = a)\mathbb{P}(a_i = a) \tag{2.5}
$$

For notational simplicity, let $\mathbb{P}(a_i = a) = p_a$. Also, denote $A_n(a)$ as the number of datapoints with action a in the dataset of n points.

WLOG, we consider the case that $a_i = 1$. We have that

$$
\begin{aligned}
&\mathbb{P}(|x_i - x_{j_i(a)}| > \epsilon | a_i = 1) \\
&\quad = \sum_{m=0}^{n-1} \mathbb{P}(|x_i - x_{j_i(a)}| > \epsilon | a_i = 1, A_n(a) = m) \mathbb{P}(A_n(a) = m | a_i = 1) \\
&\quad = \sum_{k=0}^{n-1} \mathbb{P}(|x_i - x_{j_i(a)}| > \epsilon | a_i = 1, A_n(a) = m) \binom{n-1}{m} p_a^m (1 - p_a)^{n-1-m},
\end{aligned} \tag{2.6}
$$

where the second inequality follows since, given that the ith datapoint has action 1, the probability of observing m datapoints with action a in total is just the probability of observing m a's among the $n-1$ remaining datapoints.

Now let us bound

$$
\mathbb{P}(|x_i - x_{j_i(a)}| > \epsilon | a_i = 1, A_n(a) = m).
$$

We observe that this probability is equal to the probability that all m points with $a_j = a$ are outside the radius-ϵ ball centered at x_i. Under Assumption 2.7.3, this probability is thus equal to

$$
\begin{aligned}
&\mathbb{P}(|x_i - x_{j_i(a)}| > \epsilon | a_i = 1, A_n(a) = m) \\
&\qquad = \int_{x \in \Omega} \mathbb{P}(|X_i - x_{j_i(a)}| > \epsilon | a_i = 1, A_n(a) = m, X_i = x_i) p_{\mathcal{X}}(x_i) dx \\
&\qquad = \int_{x \in \Omega} \left(1 - \frac{8\pi^2 v \epsilon^d \underline{p}}{15}\right)^m p_{\mathcal{X}}(x_i) dx \\
&\qquad = \left(1 - \frac{8\pi^2 v \epsilon^d \underline{p}}{15}\right)^m,
\end{aligned}
$$

where the second inequality uses that the volume of a $d-$dimensional ball of radius r is bounded by $\frac{8\pi^2 r^d}{15}$.

We plug this into (2.6) to yield

$$\begin{aligned}\mathbb{P}(|x_i - x_{n(i)}| > \epsilon | a_i = 1) &\leq \sum_{m=0}^{n-1} \left(1 - \frac{8\pi^2 v \epsilon^d \underline{p}}{15}\right)^m \binom{n-1}{m} p_0^m (1-p_0)^{n-1-m} \\ &= \left(1 - \frac{8\pi^2 v \epsilon^d \underline{p} p_0}{15}\right)^{n-1} \\ &\leq \left(1 - \frac{8\pi^2 v \epsilon^d \underline{p} p}{15}\right)^{n-1},\end{aligned}$$

where the first equality comes from the binomial expansion, and the second inequality comes from Assumption 2.7.10. By symmetry, we also have that for any other $a' \neq 1$

$$\mathbb{P}(|x_i - x_{j_i(a)}| > \epsilon | a_i = a') \leq \left(1 - \frac{8\pi^2 v \epsilon^d \underline{p} p}{15}\right)^{n-1}$$

Plugging these both into (2.5), we have

$$\begin{aligned}\mathbb{P}(|x_i - x_{j_i(a)}| > \epsilon) &\leq \sum_{a=1}^{A} (1 - p_a) \left(1 - \frac{8\pi^2 v \epsilon^d \underline{p} p}{15}\right)^{n-1} \\ &\leq A(1 - \underline{p}) \left(1 - \frac{8\pi^2 v \epsilon^d \underline{p} p}{15}\right)^{n-1}\end{aligned}$$

again using Assumption 2.7.4.

We then conclude from the union bound that

$$\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n}|x_{j_i(a)}-x_i|>\epsilon\right)\leq An(1-\underline{p})\left(1-\frac{8\pi^2 v\epsilon^d \underline{p}\underline{p}}{15}\right)^{n-1}$$
$$\leq An(1-\underline{p})\exp\left(\frac{-(n-1)8\pi^2 v\epsilon^d \underline{p}\underline{p}}{15}\right)$$

which follows from the fact that $(1-\frac{1}{x})^x$ converges to e^{-1} from below as $x\to\infty$.

Therefore $\|\hat{s}_{n,k}-\hat{s}^{ESR}_{n,k}\|_\infty \geq AL\epsilon$ with probability at most $An(1-\underline{p})\exp\left(\frac{-(n-1)8\pi^2 v\epsilon^d \underline{p}\underline{p}}{15}\right)$.

Now observe the following triangle inequality:

$$|s_k(\hat{\theta}^{ESR}_{n,k})-s_k(\theta_k)|\leq |s_k(\theta_k)-\hat{s}_{n,k}(\hat{\theta}_{n,k})|+|\hat{s}_{n,k}(\hat{\theta}_{n,k})-\hat{s}^{ESR}_{n,k}(\hat{\theta}^{ESR}_{n,k})|$$
$$+|\hat{s}^{ESR}_{n,k}(\hat{\theta}^{ESR}_{n,k})-\hat{s}_{n,k}(\hat{\theta}^{ESR}_{n,k})|+|\hat{s}_{n,k}(\hat{\theta}^{ESR}_{n,k})-s_k(\hat{\theta}^{ESR}_{n,k})|$$

We will use the following lemma:

Lemma 2.7.7. *Let f,g be functions which map from $\mathcal{S}\subset\mathbb{R}^m$ to $\mathbb{R}$. Suppose f and g attain their minimums, and let $s^*_f\in\operatorname{argmin}_{s\in\mathcal{S}} f(s)$ and $s^*_g\in\operatorname{argmin}_{s\in\mathcal{S}} g(s)$. Then if $\|f-g\|_\infty\leq\delta$, then $|f(s^*_f)-g(s^*_g)|\leq\delta$.*

Proof. We have that $f(s^*_f)\leq f(s^*_g)\leq g(s^*_g)+\delta$, where the first inequality comes from s^*_f being a minimizer and the second comes from the infinity-norm bound. Symmetrically, we also have that $g(s^*_g)\leq g(s^*_f)\leq f(s^*_f)+\delta$. Combining these two inequalities immediately yields the result. □

Using Lemma 2.7.7 and the union bound, we immediately get that $|s_k(\hat{\theta}^{ESR}_{n,k})-s_k(\theta_k)|\geq 16A\Delta kL'\alpha+2AL\epsilon$ with probability at most $4\max\left(1,\left(\frac{Cn}{\alpha}\right)^D\right)\exp(-2A^2L'^2nk^2\alpha^2)+An(1-\underline{p})\exp\left(\frac{-(n-1)8\pi^2 v\epsilon^d \underline{p}\underline{p}}{15}\right)$.

To show the desired result, we set $16A\Delta kL'\alpha = 2AL\epsilon = \frac{t}{2}$, which yields $\alpha = \frac{t}{32A\Delta kL'}$ and $\epsilon = \frac{t}{4AL}$. Thus we have the final result, after subtracting off the maximum reward per context and taking the negation:

$$\begin{aligned}
&\mathbb{P}\big(|\mathbb{E}_{x\sim P_{\mathcal{X}}} R^k_{\hat{\theta}^{ESR}_{n,k}}(x) - \mathbb{E}_{x\sim P_{\mathcal{X}}} R^k_{\theta_k}(x)| \geq t\big) \\
&\qquad \leq 4\max\left(1, \left(\frac{32\Delta CAL'nk}{t}\right)^D\right)\exp\left(\frac{-nt^2}{512\Delta^2}\right) \\
&\qquad\quad + An(1-p)\exp\left(\frac{-(n-1)8\pi^2 v\underline{p}pt^d}{15(4AL)^d}\right).
\end{aligned}$$

In other words, the excess regret converges to 0 in probability.

□

2.7.1 Noisy Reward

In the above sections, we considered $r(x, a)$ to be noise-free, in which case our bounds agree with those from Hu et al. [2024]. In this section, we explore the effect of adding noise to r, so that the 1-NN/matching estimator $\hat{r}$ is no longer consistent [Györfi et al., 2002]. In this case, the rate from Hu et al. [2024] will no longer vanish with n since their regret bound depends on the rate at which the estimator $\hat{r}$ converges to r (in mean square, see Proposition 2). In contrast, we show a novel result that using our same estimator for $\hat{r}$ still yields a policy whose excess regret asymptotically converges to 0.

Concretely, we now suppose that $r(x, a) = \overline{r}(x, a) + \epsilon_{x,a}$ where $\epsilon_{x,a}$ are i.i.d. with mean 0 and finite variance $\sigma^2_{x,a} \leq \sigma^2$. In this case, we have the following result. We define $\overline{R}^k_\theta(x) = \max_{a\in\mathcal{X}} \overline{r}(x, a) - \sum_{a=1}^A \overline{r}(x, a)\frac{\exp(k\hat{r}(x,a))}{\sum_a \exp(k\hat{r}(x,a))}$.

Theorem 2.7.8. *Define θ^k as the minimizer of the expected soft regret within Θ, and define $\hat{\theta}_{ESR,n}$ as the minimizer of the ESR loss function over n datapoints. Suppose that Assumptions 2.7.1 through 2.7.4 hold, and furthermore assume Ω is convex. Then there exists constants $F', G', H', K', M', Q > 0$ which depend on $\Delta, L, L', A, B, C, \underline{p}, v, p, \mathrm{diam}(\Omega)$ such that,*

$$\mathbb{P}\left(|\mathbb{E}_{x\sim P_{\mathcal{X}}}\overline{R}^k_{\hat{\theta}^{ESR}_{n,k}}(x) - \mathbb{E}_{x\sim P_{\mathcal{X}}}\overline{R}^k_{\theta_k}(x)| \geq t\right) \leq \max\left(4, F'\left(\frac{nk}{t}\right)^D\right)\exp\left(-G'nt^2\right) + H'n\exp\left(\frac{-(n-1)K't^d}{M'^d}\right) + \frac{Q}{nt^2}.$$

Proof. We now introduce additional s terms, similar to the previous proof. We have

$$\overline{s}_k(\theta) = \mathbb{E}_{x\sim P_{\mathcal{X}}}\sum_{a=1}^{A}\overline{r}(x,a)\frac{\exp(k\hat{r}_\theta(x,a))}{\sum_{a'=1}^{A}\exp(k\hat{r}_\theta(x,a))},$$

$$\hat{\overline{s}}_{n,k}(\theta) = \frac{1}{n}\sum_{i=1}^{n}\sum_{a=1}^{A}\overline{r}(x_i,a)\frac{\exp(k\hat{r}_\theta(x_i,a))}{\sum_{a'=1}^{A}\exp(k\hat{r}_\theta(x_i,a))},$$

$$\hat{\overline{s}}^{ESR}_{n,k}(\theta) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j\in n(i)}\overline{r}(x_j,a_j)\frac{\exp(k\hat{r}_\theta(x_i,a_j))}{\sum_{a'=1}^{A}\exp(k\hat{r}_\theta(x_i,a))},$$

and finally

$$\hat{s}^{ESR}_{n,k}(\theta) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j\in n(i)}r(x_j,a_j)\frac{\exp(k\hat{r}_\theta(x_i,a_j))}{\sum_{a'=1}^{A}\exp(k\hat{r}_\theta(x_i,a))}.$$

Call $\overline{\theta}$, $\overline{\theta}^k_n$, $\overline{\theta}^{ESR}_{n,k}$, and $\hat{\theta}^{ESR}_{n,k}$ the respective minimizers.

We know from the previous theorem that the first three terms are close in $\|\cdot\|_\infty$ norm with high probability. We now show $\mathbb{E}_x\left[\sup_\theta |\hat{\bar{s}}^{ESR}_{n,k}(\theta) - \hat{s}^{ESR}_{n,k}(\theta)|^2\right]$ converges to 0 with n.

For notational simplicity, below we use the notation that $\epsilon_i = \epsilon_{x_i,a_i}$. Therefore we are aiming to bound $\mathbb{E}_x\left[\sup_\theta |\frac{1}{n}\sum_{i=1}^n \sum_{j\in n(i)} \epsilon_j \frac{\exp(k\hat{r}_\theta(x_i,a_j))}{\sum_{a'=1}^A \exp(k\hat{r}_\theta(x_i,a))}|^2\right]$

We use the bias-variance decomposition. It is clear that the mean is 0, since each ϵ_k has mean zero by assumption. We then turn our attention to the variance.

Observe that $\mathrm{Var}(\sup_\theta \frac{1}{n}\sum_{i=1}^n \sum_{j\in n(i)} \epsilon_j \frac{\exp(k\hat{r}_\theta(x_i,a_j))}{\sum_{a'=1}^A \exp(k\hat{r}_\theta(x_i,a))}) \leq \mathrm{Var}\sum_{i=1}^n \sum_{j\in n(i)} \epsilon_j$, since each softmax term is at most 1.

So let's consider $\mathrm{Var}\sum_{i=1}^n \sum_{j\in n(i)} \epsilon_j$. Observe that we can then rewrite the difference $\frac{1}{n}\sum_{i=1}^n \sum_{j\in n(i)} \epsilon_j$ to instead index over the number of times each datapoints is used as a neighbor, i.e. $\frac{1}{n}\sum_{k=1}^n N_k\epsilon_k$ where N_k is defined as the number of times datapoint k appears as a neighbor j for any $i \in 1,\ldots,n$. By the multiclass version of Lemma 3 from Abadie and Imbens [2006] (which uses the convex support assumption), we have that $N_k = O_p(1)$ and that $\mathbb{E}[N_k^q]$ is bounded uniformly in n for any $q > 0$. In particular, suppose the uniform bound for $q = 2$ is $W(A, p, v, \underline{p}, \mathrm{diam}(\Omega), d)$, which depends on other parameters but not n. For simplicity, below we just use W.

Then we have that

$$\mathbb{E}\left[(\frac{1}{n}\sum_{k=1}^n N_k\epsilon_k)^2\right] = \frac{1}{n^2}\sum_{k=1}^n \mathbb{E}[N_k]^2\mathbb{E}[\epsilon_k^2] \leq \frac{W\sigma^2}{n},$$

where the first equality uses that the ϵ_k are i.i.d and that the N_k are independent of the ϵ_k, and the second inequality uses the bounds on $\mathbb{E}[N_k^2]$ and $\sigma^2_{x,a}$.

Therefore we have $\mathbb{E}\left[\sup_\theta |\hat{\overline{s}}^{ESR}_{n,k}(\theta) - \hat{s}^{ESR}_{n,k}(\theta)|^2\right] \leq \frac{W\sigma^2}{n}$, and applying Markov's inequality yields

$$\mathbb{P}\left(\sup_\theta |\hat{\overline{s}}^{ESR}_{n,k}(\theta) - \hat{s}^{ESR}_{n,k}(\theta)| \geq t'\right) \leq \frac{W\sigma^2}{nt'^2}.$$

By the same arguments as in Theorem 2.7.5 and applying Lemma 2.7.7 and the union bound to the statement of Theorem 2.7.5, we set $16A\Delta kL'\alpha = 2AL\epsilon = 2t' = \frac{t}{3}$ to get the final bound .

$$\begin{aligned}
&\mathbb{P}\big(|\mathbb{E}_{x\sim P_\mathcal{X}}\overline{R}^k_{\hat{\theta}^{ESR}_{n,k}}(x) - \mathbb{E}_{x\sim P_\mathcal{X}}\overline{R}^k_{\theta_k}(x)| \geq t\big) \\
&\qquad \leq 4\max\left(1, \left(\frac{48\Delta CAL'nk}{t}\right)^D\right)\exp\left(\frac{-nt^2}{1152\Delta^2}\right) \\
&\qquad\quad + An(1-p)\exp\left(\frac{-(n-1)8\pi^2 v\underline{p}pt^d}{15(6AL)^d}\right) + \frac{36W\sigma^2}{nt^2}.
\end{aligned}$$

□

This is related to undersmoothing phenomenon [Newey et al., 1998], since traditionally nearest-neighbor type estimators require the number of neighbors to grow with the number of datapoints n to yield consistent estimates [Shalizi, 2019]. In this case, we are using a variant of the 1-nearest neighbor estimate for r, which has variance which does not go to 0 with n, but which nevertheless yields consistency for the reward estimate since it is part of a weighted average across n samples.

2.7.2 Deterministic Policies

In the previous section, we compared the performance of the policy trained on ESR loss against policies which can be represented in the softmax parameterization, for fixed k.

In this section, we compare the performance of the policy trained on ESR against the best deterministic policy in the model class. Our theoretical result here requires a slightly different set of assumptions. In particular, we assume for simplicity that $|\mathcal{A}| = \{0, 1\}$, and the following additional assumptions.

Assumption 2.7.9 (Model class Θ)**.** The class of models r_θ parameterized by $\theta \in \Theta$, denoted r_Θ, satisfies the following additional constraints.

(a) Θ is compact.

(b) For all $\theta \in \Theta$, we have that $\forall \lambda$ such that $|\lambda| \leq 1$, $\exists \theta'$ such that $\hat{\delta}_\theta(x) = \lambda \hat{\delta}_{\theta'}(x)$ for all $x \in \mathcal{X}$.

Assumption 2.7.9 (b) states that the predicted differences $\hat{\delta}_\theta(x)$ can be scaled down arbitrarily small across all contexts x. A simple example satisfying this is a neural network which can scale down the output $\hat{r}_\theta(a, x)$ arbitrarily small. This is simply achieved for neural networks with a linear output layer (as is standard in regression) by fixing all weights before the output layer, and scaling down the weights of the output layer by λ.

In this case, we restrict our attention to the finite $|\mathcal{X}|$ assumption and instead, replace Assumption 2.7.3 with the following:

Assumption 2.7.10 (Distributional assumptions on $P_\mathcal{X}$)**.** $P_\mathcal{X}$ has full support on $|\mathcal{X}|$, so that there exists a lower bound $\underline{p}$ of the probability mass of any $x \in \mathcal{X}$.

Finally, we have the following new assumption.

Assumption 2.7.11 (Optimality of Deterministic Policies)**.** Define A and $\overline{A}$ as follows:

$$A := \left\{ \{ \frac{1}{1 + \exp(\lambda \operatorname{sgn}\{\delta(x)\} \hat{\delta}_\theta(x))} : x \in \mathcal{X} \} : \lambda \in \mathbb{R}, \theta \in \Theta \right\}$$

and

$$\overline{A} := \left\{ \{ \frac{1}{2} | \operatorname{sgn}\{\delta(x)\} - \operatorname{sgn}\{\hat{\delta}_\theta(x)\} | : x \in \mathcal{X} \} : \theta \in \Theta \right\}.$$

Assume that

$$\min_{a \in \mathrm{cl} A} \mathbb{E}_{x \sim P_\mathcal{X}} |\delta(x)| a = \min_{a \in \overline{A}} \mathbb{E}_{x \sim P_\mathcal{X}} |\delta(x)| a \tag{2.7}$$

The interpretation of this assumption is that there exists a deterministic policy which is optimal in the model class, for all choices of k; the RHS of (2.7) is the optimal regret achieved by a deterministic policy, and the corresponding LHS is the optimal regret for any softmax (randomized) policy, for any choice of k.

Note that this assumption is not always satisfied. In particular, consider the following example. Let $\mathcal{X} = \{-10, 1\}$, and $P_\mathcal{X}(X = -10) = P_\mathcal{X}(x = 1) = \frac{1}{2}$. Let $\Theta = [-M, M]$ for some large positive constant M. Suppose that $\delta(x) = 1$ for all x, and $\hat{\delta}_\theta(x) = \theta x$ (e.g. $\hat{r}_\theta(x, a) = a\theta x$), so that for $a \in A$

$$\mathbb{E}_{x \sim P_\mathcal{X}} |\delta(x)| a = \frac{1/2}{1 + \exp(-10\lambda\theta)} + \frac{1/2}{1 + \exp(\lambda\theta)}.$$

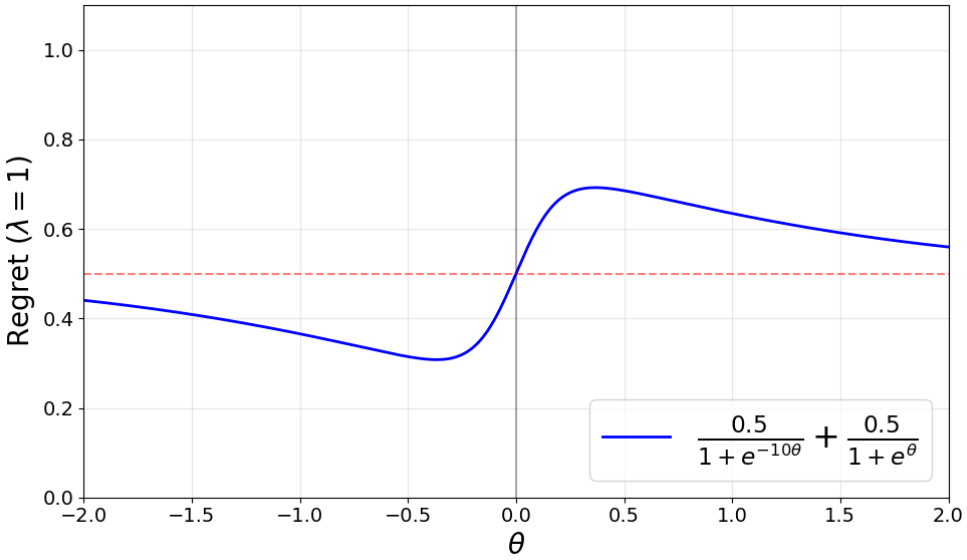


Figure 2.1: Example violating Assumption 2.7.11.

Observe that for $a \in \overline{A}$, $\mathbb{E}_{x\sim P_{\mathcal{X}}}|\delta(x)|a = \frac{1}{2}$ for any choice of θ.

However, take $\lambda = 1$, and $\theta = -0.36722$. Then $\mathbb{E}_{x\sim P_{\mathcal{X}}}|\delta(x)|a \approx 0.30779 < 0.5$, and so the LHS of (2.7) does not equal the RHS. See Figure 2.1 for a graph of $\mathbb{E}_{x\sim P_{\mathcal{X}}}|\delta(x)|a$.

Under the assumptions from and these additional assumptions, we have the following theorem.

Theorem 2.7.12. *Define θ^* as the minimizer of the expected regret within Θ, and define $\hat{\theta}_{ESR,n}$ as the minimizer of the ESR loss function over n datapoints. Suppose Assumptions 2.7.1, 2.7.2, 2.7.4, 2.7.9, and 2.7.11 hold. Then*

$$\mathbb{E}_{x\sim P_{\mathcal{X}}} R^k_{\hat{\theta}_{ESR,n}}(x) \xrightarrow{P} \mathbb{E}_{x\sim \mathcal{P}_{\mathcal{X}}} R_{\theta^*}$$

as long as $k \in \Theta(n^\beta)$ for some constant $\beta > 0$.

Proof. As the action space is binary, we may rewrite $R_\theta(x)$ as:

$$\begin{aligned} R_\theta(x) &= \max_{a\in\mathcal{A}} r(x_i, a) - r(x, \operatorname*{argmax}_{a\in\mathcal{A}} \hat{r}_\theta(x, a)) \\ &= \frac{1}{2}|\operatorname{sgn}\{r(x,1) - r(x,0)\} - \operatorname{sgn}\{\hat{r}_\theta(x,1) - \hat{r}_\theta(x,0)\}| \cdot |r(x,1) - r(x,0)|. \end{aligned}$$

We define $s(\theta)$ as $s(\theta) := \mathbb{E}_{x\sim\mathcal{X}}\left[R_\theta(x)\right]$, and $\theta^s \in \operatorname{argmin}_\theta s(\theta)$.

Here we define the behavior when there is a tie in $\hat{r}_\theta(x, a)$ (i.e. when the argmax of $\hat{r}_\theta(x, a)$ is both actions) as incurring half the regret if the true minimizer is either 0 or 1 (but not both).

Let us first show the following lemma, where we adopt the same notation as in the proof of Theorem 2.7.5.

Lemma 2.7.13. $\lim_{k\to\infty} s_k(\theta_k) = s(\theta^s)$.

Proof. We first define the following quantity.

$$s_k(\lambda, \theta) := \langle |\delta(x)| p_{\mathcal{X}}(x), \frac{1}{1 + \exp(k\lambda \operatorname{sgn}\{\delta(x)\}\hat{\delta}_\theta(x)} \rangle,$$

where this notation implicitly assumes $|\mathcal{X}|$ is finite and so $|\delta(x)|p_{\mathcal{X}}(x)$ is the vector of $|\delta(x_i)|p_{\mathcal{X}}(x_i)$ and analogously for the second term.

Now define A_k as

$$A_k := \left\{ \{\frac{1}{1 + \exp(\lambda \operatorname{sgn}\{\delta(x)\}\hat{\delta}_\theta(x))} : x \in \mathcal{X}\} : |\lambda| \le k, \theta \in \Theta \right\}$$

Observe that A_k is compact: $|\lambda| \le k$ is compact, and so is $\{\operatorname{sgn}\{\delta(x)\}\hat{\delta}_\theta(x) : \theta \in \Theta\}$ (since Θ is compact by assumption 2.7.2. The function $f(x, y) = xy$ is

continuous, so $\{\lambda \operatorname{sgn}\{\delta(x)\}\hat{\delta}_\theta(x) : |\lambda| \leq k, \theta \in \Theta\}$ is compact. Finally, $g(z) = \frac{1}{1+\exp(z)}$ is continuous, so A_k is compact.

Therefore the minimizer $b_k := \min_{|\lambda|\leq k, \theta\in\Theta} s_1(\lambda, \theta) = \min_{\vec{a_k}\in A_k} \langle |\delta(x)| p_{\mathcal{X}}(x), \vec{a}_k \rangle$ exists.

But also note that $b_k = \min_{|\lambda|\leq k, \theta\in\Theta} s_1(\lambda, \theta) = \min_{|\lambda|\leq 1, \theta\in\Theta} s_k(\lambda, \theta) = \min_{\theta\in\Theta} s_k(\theta)$.

Now we show that $\lim_{k\to\infty} b_k = \min_{a\in \mathrm{cl}A} \langle |\delta(x) p_{\mathcal{X}}(x), a\rangle$.

Let $a^* := \operatorname{argmin}_{a\in\mathrm{cl}A} \langle |\delta(x)| p_{\mathcal{X}}(x), a\rangle$, which exists since $\langle |\delta(x)| p_{\mathcal{X}}(x), \cdot \rangle$ is continuous in the second argument.

Consider a sequence of $\{a_m\} \subset A$ such that $\lim_{m\to\infty} a_m = a^*$. Note each $a_m \in A_{k(m)}$ for some $k(m)$, by definition of A (in particular, since A has $\lambda \in \mathbb{R}$). Note $b_{k(m)} \leq \langle |\delta(x)| p_{\mathcal{X}}(x), a_m \rangle$ since $b_{k(m)}$ is the minimizer.

Now fix $\epsilon > 0$. By continuity of $\langle |\delta(x)| p_{\mathcal{X}}(x), \cdot \rangle$, there exists some M such that for all $m \geq M$, $\langle |\delta(x)| p_{\mathcal{X}}(x), a_m \rangle \leq \langle |\delta(x)| p_{\mathcal{X}}(x), a^* \rangle + \epsilon$.

Therefore $k(m)$ satisfies $b_{k(m)} \leq \langle |\delta(x)| p_{\mathcal{X}}(x), a^* \rangle + \epsilon$.

Also $b_{k(m)} \geq \langle |\delta(x)| p_{\mathcal{X}}(x), a^* \rangle$, since $A_{k(m)} \subset \mathrm{cl}A$.

Also, for any $k > k(m)$, $b_k \leq b_{k(m)}$ since the minimum is taken over a bigger set, and $b_k \geq \langle |\delta(x)| p_{\mathcal{X}}(x), a^* \rangle$ remains true for the same reason.

Therefore for all $\epsilon > 0$, there exists a K such that for all $k \geq K$, $|b_k - \min_{a\in\mathrm{cl}A} \langle |\delta(x)| p_{\mathcal{X}}(x), a\rangle| \leq \epsilon$. This implies $\lim_{k\to\infty} b_k = \min_{a\in\mathrm{cl}A} \langle |\delta(x)| p_{\mathcal{X}}(x), a\rangle|$, and the last term equals $\min_\theta s(\theta)$ by Assumption 2.7.11.

□

We first observe that the same argument for bounding $\|s_k - \hat{s}_{n,k}\|_\infty$ holds as for Theorem 2.7.5

For $\|\hat{s}_{n,k} - \hat{s}^{ESR}_{n,k}\|_\infty$, we need to slightly adjust the proof for bounding $\mathbb{P}(|x_i - x_{n(i)}| > \epsilon | a_i = 1, A_n(0) = m)$.

Given that $\mathcal{X}$ is finite, we can write this as

$$
\begin{aligned}
&\mathbb{P}(|x_i - x_{n(i)}| > \epsilon | a_i = 1, A_n(0) = m) \\
&\quad = \sum_{x \in \mathcal{X}} \mathbb{P}(|x_i - x_{n(i)}| > \epsilon | a_i = 1, A_n(0) = m, X_i = x) \mathbb{P}(x_i = x | a_i = 1, A_n(0) = m) \\
&\quad \leq \sum_{x \in \mathcal{X}} \mathbb{P}(|x_i - x_{n(i)}| > 0 | a_i = 1, A_n(0) = m, X_i = x) \mathbb{P}(x_i = x | a_i = 1, A_n(0) = m) \\
&\quad = \sum_{x \in \mathcal{X}} (1 - \mathbb{P}(x_i = x))^m \mathbb{P}(x_i = x | a_i = 1, A_n(0) = m) \\
&\quad \leq \sum_{x \in \mathcal{X}} (1 - \underline{p})^m \mathbb{P}(x_i = x | a_i = 1, A_n(0) = m) \\
&\quad = (1 - \underline{p})^m,
\end{aligned}
$$

where the second inequality is a consequence of the definition of $x_{n(i)}$, since $|x_i - x_{n(i)}| > 0$ is equivalent to $x_{n(i)} \neq x_i$, which in turns is equivalent to all k points comprising $A_n(0)$ not having context x_i, and last inequality comes from Assumption 2.7.4.

We plug this into (2.6) to yield

$$\begin{aligned}\mathbb{P}(|x_i - x_{n(i)}| > \epsilon | a_i = 1) &\leq \sum_{m=0}^{n-1} (1-\underline{p})^m \binom{n-1}{m} p_0^m (1-p_0)^{n-1-m} \\ &= (1 - \underline{p}p_0)^{n-1} \\ &\leq (1-\underline{p}p)^{n-1},\end{aligned}$$

where the first equality comes from the binomial expansion, and the second inequality comes from assumption 2.7.10. By symmetry, we also have that

$$\mathbb{P}(|x_i - x_{n(i)}| > \epsilon | a_i = 0) \leq (1-\underline{p}p)^{n-1}$$

Plugging these both into (2.5), we have

$$\begin{aligned}\mathbb{P}(|x_i - x_{n(i)}| > \epsilon) &\leq (1-p_0)(1-\underline{p}p)^{n-1} + (1-p_1)(1-\underline{p}p)^{n-1} \\ &\leq (2-2\underline{p})(1-\underline{p}p)^{n-1},\end{aligned}$$

agains using Assumption 2.7.4.

We then conclude from the union bound that

$$\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n} |x_{n(i)} - x_i| > \epsilon\right) \leq 2n(1-\underline{p})(1-\underline{p}p)^{n-1}.$$

It follows that $\|\hat{s}_{n,k} - \hat{s}^{ESR}_{n,k}\|_\infty \geq \epsilon \left(L + \frac{\Delta L' k}{4}\right)$ with probability at most $2n(1-\underline{p})(1-\underline{p}p)^{n-1}$.

Then, as in the proof of Theorem 2.7.5, we use Lemma 2.7.7 and the union bound to conclude that $|s_k(\hat{\theta}^{ESR}_{n,k}) - s_k(\theta_k)| \geq 8\Delta k L' \alpha + \epsilon(2L + \frac{\Delta L' k}{2})$ with probability at most $4\max\left(1, \left(\frac{Cn}{\alpha}\right)^D\right)\exp(-\frac{L'^2 n k^2 \alpha^2}{2}) + 2n(1-\underline{p})\exp\left(\frac{-(n-1)8\pi^2 v \epsilon^d \underline{p}\underline{p}}{15}\right)$.

Again, we set $8\Delta kL'\alpha = \epsilon(2L + \frac{\Delta L'k}{2}) = \frac{t}{2}$, to get that $|s_k(\hat{\theta}^{ESR}_{n,k}) - s_k(\theta_k)| \geq t$ with probability at most $4\max\left(1, \left(\frac{4\Delta CL'n}{\alpha}\right)^D\right)\exp(-\frac{nt^2}{512\Delta^2}) + 2n(1-\underline{p})(1-\underline{p}p)^{n-1}$.

We set $k(n)$ as a function of n, so that $\lim_{n\to\infty} k_n = \infty$. In particular, we set $k \in \Theta(n^\beta)$ for some $\beta < 0$, so that $\alpha \in \Theta(n^{-\beta})$.

Therefore we have that for $k \in \Theta(n^\beta)$, we have $|s_k(\hat{\theta}^{ESR}_{n,k}) - s_k(\theta_k)| \geq t$ with probability at most $4\max\left(1, \left(4\Delta CL'n^{1+\beta}\right)^D\right)\exp(-\frac{nt^2}{512\Delta^2}) + 2n(1-\underline{p})(1-\underline{p}p)^{n-1}$.

Now consider $\mathbb{P}\left(|s_k(\hat{\theta}^{ESR}_{n,k}) - s(\theta_s)| \geq s\right)$. We can use the union bound to get the following upper bound

$$\begin{aligned}\mathbb{P}(|s_k(\hat{\theta}^{ESR}_{n,k}) - s(\theta_s)| \geq s) \leq & \mathbb{P}\left(|s_k(\hat{\theta}^{ESR}_{n,k}) - s_k(\theta_k) \geq \frac{s}{2}|\right) \\ & + \mathbb{P}\left(|s_k(\theta_k) - s(\theta) \geq \frac{s}{2}|\right). \end{aligned} \tag{2.8}$$

From above, when $k \in \Theta(n^\beta)$, the first term of (2.8) is bounded by $4\max\left(1, \left(4\Delta CL'n^{1+\beta}\right)^D\right)\exp(-\frac{ns^2}{2048\Delta^2}) + 2n(1-\underline{p})(1-\underline{p}p)^{n-1}$ (which goes to 0 as $n \to \infty$), and the second term goes to 0 by Lemma 2.7.13.

Therefore when $k \in \Theta(n^\beta)$, we conclude that $s_k(\hat{\theta}^{ESR}_{n,k})$ converges in probability to $s(\theta_s)$, as desired.

□

2.8 Experiments

This section contains results from experiments in a semi-synthetic benchmark dataset based on the Infant Health and Development Program (IHDP) and a real

news recommender dataset from Yahoo, comparing the performance of state-of-the-art models with that of ESR-trained models. A link to all code is in the appendix.

2.8.1 IHDP Dataset

We evaluate our method on a standard benchmark for CATE estimators. The dataset is based on the Infant Health and Development Program (IHDP), which measured the impact of specialist home visits on future cognitive test scores for $n = 747$ (608 control/139 treatment) children. Each individual child is represented by 25 covariates. We follow the procedure used in Hill [2011], Johansson et al. [2016], Shalit et al. [2017], Johansson et al. [2022], which uses the log-linear simulated outcome implemented as setting "A" in the NPCI package [Dorie, 2016]. We use 1000 replications with a 70:30 train/test split.

We compare a model trained on the ESR loss against several state-of-the-art CATE estimators in the literature: the S-, T- [Künzel et al., 2019], R- [Nie and Wager, 2021], and DR-learners [Kennedy, 2023]. The CATE estimate is used to produce a decision by taking the sign of the estimated CATE, i.e. $\hat{a} = \operatorname{sgn} \widehat{CATE}(x)$. Each CATE estimator uses a two-layer neural network. We also include a policy learner based on the IPW method [Kitagawa and Tetenov, 2018], where $\hat{\pi} = \operatorname{argmax}_{\pi \in \Pi} \frac{1}{n} \sum_{i=1}^{n} \frac{y_i \pi(a_i|x_i)}{\pi_0(a_i|x_i)}$. Note that $\pi_0(a_i|x_i)$ is the (assumed) known probability that treatment a_i is chosen given x_i; here they are unknown so we use the corresponding empirical probabilities from the data.

As this dataset is semi-synthetic, the counterfactual outcome is given and so regret can be calculated in a straightforward manner. Table 2.1 shows the perfor-

Table 2.1: (Left) CTR (95% CI) for test regret over 1000 replications. The ESRloss was computed using $k = 10$. (Right) Comparison of regret vs. k for ESR, 95% CI over 1000 replications.

Learner	Regret
S	[1.02, 1.36]
T	[0.70, 0.97]
R	[2.81, 3.19]
DR	[0.77, 1.21]
IPW	[0.97, 1.29]
ESR	[0.37, 0.46]

k	Regret
1	[0.70, 0.83]
5	[0.43, 0.53]
10	[0.37, 0.46]
25	[0.38, 0.46]
50	[0.39, 0.48]
100	[0.40, 0.49]

mance of the learners compared against the model trained with ESR. The ESR loss-trained model handily outperforms the other metalearners in regret. Table 2.1 also shows the regret incurred under the ESRloss when varying k (which controls the smoothness of the loss). The performance varies but is generally stable in the $[5, 100]$ range.

2.8.2 News Recommender

To evaluate the performance of our method on real-world data, we use a news recommender dataset from Yahoo! [Chu et al., 2009]. This dataset is often used for testing contextual bandit algorithms [Li et al., 2010, Semenov et al., 2022]. This dataset is comprised of over 45 million user visits to the Yahoo Today Module over a ten day period in May 2009. Each user was presented with an article chosen uniformly at random from a pool of 20 articles, and Yahoo recorded whether the user clicked. Each user is associated with a set of 5 features which were derived from embedding the actual user-level data to obscure personally identifiable information. The reward is given by whether the user clicks (a reward of 1) or not (a reward of

0). The task is to predict whether a user will click, given the action space of the twenty articles $\{a_1, \ldots, a_{20}\}$ and given the user-level features as context.

As calculating exact nearest neighbors is computationally expensive when the number of datapoints is large (as is the case here), we rely on the Ball-Tree method [Omohundro, 1989] to calculate approximate nearest neighbors (we compare against the KD-tree [Bentley, 1975], an alternative algorithm, in the appendix; both perform similarly). Nearest neighbors are computed only once, at the beginning of model training.

As our reward model class for all methods evaluated, we use a fully-connected neural network with two hidden layers. For our method and the direct method, we use a single output node. For the ESR-trained model, we mirror the direct method by choosing the action based on the maximum predicted reward as opposed to using the randomized softmax policy; we expect the results to be robust to this choice. In training the ESR, we use a schedule for the parameter k: $k_i = k_0 + i$ for training epoch i (we set $k_0 = 5$). This lets the ESR loss function approximate the true regret better as the training loss converges; we found this to improve performance. For the policy-based methods, the final layer is a softmax output to twenty nodes, as the output is necessarily a probability distribution over the twenty articles.

For comparison, we compare our method with the direct method [Beygelzimer and Langford, 2009] that trains with MSE. Among policy-based methods, we compare against the IPW method [Rosenbaum and Rubin, 1983], the BanditNet [Joachims et al., 2018], and the IPW method with the Pseudo-Loss regularizer [Wang et al., 2024]. These methods were chosen as benchmarks as they, like our method, are amenable to the use of black-box machine learning models that accept custom loss functions. These methods are all built on the IPW method, which is

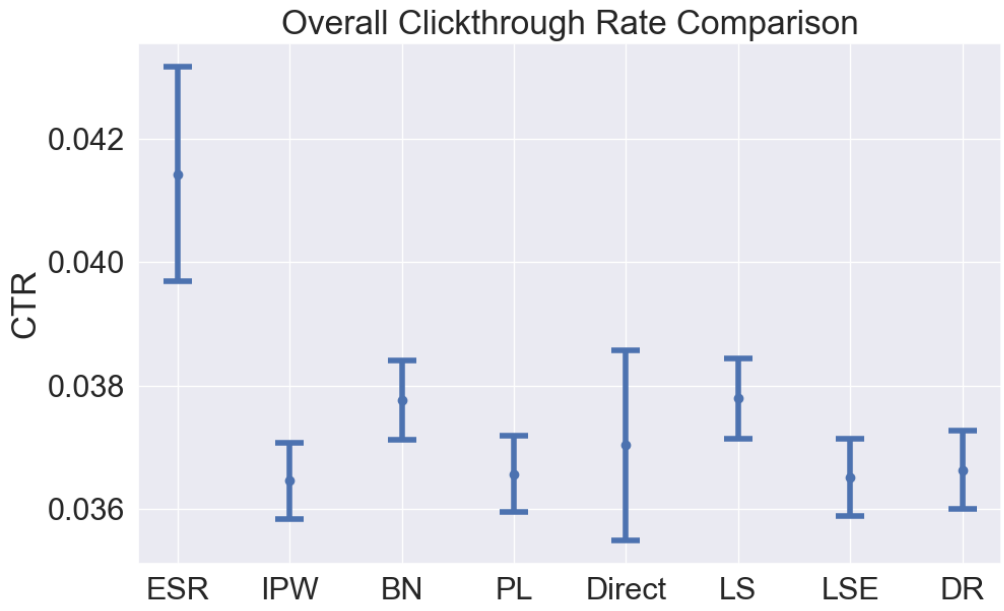


Figure 2.2: 95% confidence intervals for clickthrough rates across all days. The ESR-trained model outperforms all other models.

to minimize $\hat{\pi} = \text{argmax}_{\pi \in \Pi} \frac{1}{n} \sum_{i=1}^{n} \frac{r_i \pi(a_i|x_i)}{\pi_0(a_i|x_i)}$. The BanditNet (BN) shifts the reward r_i by a Lagrangian multiplier λ, while the Pseudo-Loss (PL) regularizer adds $\frac{\beta}{n} \sum_{i=1}^{n} \sum_{a \in \mathcal{A}} \frac{\pi(a|x_i)}{\pi_0(a|x_i)}$ to the objective. Here λ and β are treated as hyperparameters; we show the best performing choices in our results. We also compare against smoothed IPW methods based on logarithmic smoothing (LS) Sakhi et al. [2024] and using log-sum-exp (LSE) [Behnamnia et al.]. Finally, we compare against the doubly robust (DR) method [Dudík et al., 2014]. In the appendix, we also compare against a variant of DR with shrinkage [Su et al., 2020]; we only include standard DR here in the main body because it performed the best.

Evaluation

For each article pool, we select 70% of the users as the training set and the remaining 30% as the test set. Because we do not have click data for all articles for each user, we must use another metric to gauge performance. Given a policy π that deterministically maps from x to action a, we use the expected click-through rate of the policy π, where the expectation is over features x. To evaluate this, we use *off-policy* evaluation, a technique common in contextual bandits [Agarwal and Kakade, 2019], leveraging the fact that the dataset was collected under a known random policy (uniform random decisions). The estimand is the value of policy π, i.e. $V(\pi) := \mathbb{E}_{x\sim P_{\mathcal{X}}, a\sim\pi(x)}\left[r(x,a)\right]$. To estimate this quantity, we only look at the datapoints (x_i, a_i, r_i) such that our model's outputted decision $\pi(x_i)$ matches the observed action a_i, and compute the average performance out of these points. Specifically, we define our estimator $\hat{V}(\pi)$ as $\hat{V}(\pi) := \frac{\sum_{i=1}^{m} \mathbb{1}\{\pi(x_i)=a_i\} r_i}{\sum_{i=1}^{m} \mathbb{1}\{\pi(x_i)=a_i\}}$.

We have the following result (proof in the appendix) on the consistency of this estimator.

Proposition 2.8.1. *The estimator $\hat{V}(\pi)$ is consistent for $V(\pi)$ as $m \to \infty$.*

Figure 2.2 and Figure 2.3 summarize the results. The model trained on ESR outperforms all other methods on most of the days. Moreover, its estimated overall CTR is higher and the 95% confidence interval for its overall CTR does not overlap with the confidence intervals for any of the other baseline methods. This suggests that the model trained on the ESR loss is indeed doing a better job of capturing the dependence of the click-through rate on the article choice.

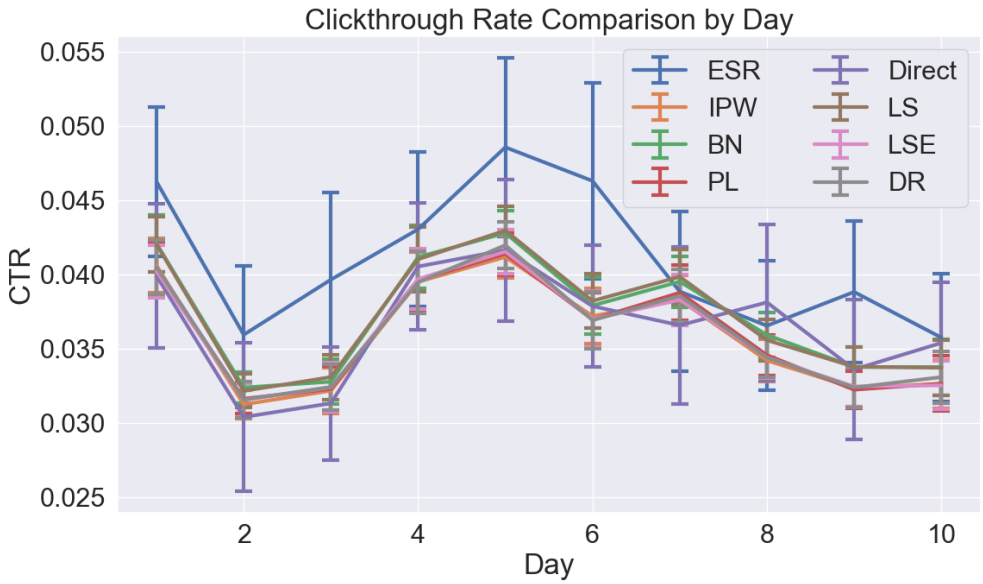


Figure 2.3: 95% CI for CTR of ESR loss-trained model vs. other methods over ten days.

2.9 Conclusion and Future Work

This chapter tackled the problem of model misspecification in value-based learning. Instead of mean squared error, we proposed an alternative ESR loss function that targets the learning of the model towards minimizing the regret incurred by the policy derived from the reward model. We justified our approach theoretically in that optimizing the ESR loss function yields asymptotically optimal regret within a model class, even when the reward model is misspecified. We also provided numerical results on a standard semi-synthetic benchmark and real-world data that our approach outperforms standard methods within contextual bandit and causal inference problems.

Several future directions stem from this work. One is to develop theory for more general actions spaces beyond discrete $\mathcal{A}$. Another is applying similar methods to

other machine learning models, especially reinforcement learning. Likewise, we see potential to extend our methodology to learning from preferences in the fine-tuning of large language models as in Rafailov et al. [2023].

CHAPTER 3

LOSS FUNCTION DESIGN FOR NEURAL INTERATOMIC POTENTIALS

This chapter is based on joint work with Colin Bundschu and Peter Frazier.

3.1 Introduction

In computational materials science, predicting the energy of a configuration of a material is a ubiquitous problem. The energy of a configuration determines almost all physical properties of interest, such as thermodynamic stability, phase diagrams, and chemical reactivity.

The function describing the potential energy of a configuration given the positions of its atoms is known as the interatomic potential. An important challenge in calculating the interatomic potential is that accurate calculations of energy rely on solving complex quantum mechanics equations, which become computationally expensive as the number of atoms increase. In particular, to evaluate the energy of a newly proposed material, the standard approach is to use density functional theory (DFT), a computationally expensive procedure which outputs a materials' ground-state energy given its atomic configuration. Calculating DFT involves solving a system of coupled, nonlinear partial differential equations over the electron density. This process involves iteratively updating the electron density until self-consistency is reached. This process can take anywhere from hours to weeks on supercomputers for proposed materials [Giustino, 2014].

The computation required by DFT has driven development of machine learning approaches which make energy prediction faster and more accessible for large-

scale materials discovery. While DFT-calculated energies are often used as ground truths, recent work in computational materials science explores ways to apply machine learning to predict energies, in particular with carefully designed neural network architectures [Smith et al., 2017, Schütt et al., 2018, Batatia et al., 2022]; such models are known as neural potentials.

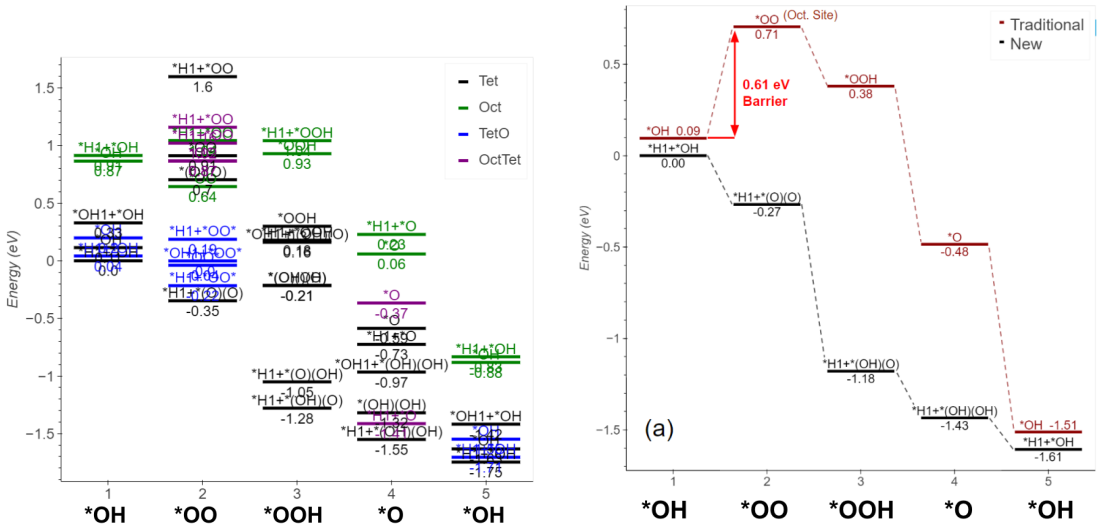
In this chapter, we explore loss function design for the task of neural interatomic potentials. Whereas previous methods focus on designing neural architectures and/or feature engineering, we focus on changing the loss function used in training, independent of the model choice. Section 3.2 introduces such a loss function in the framework of offline contextual bandits for the problem of minimum energy pathway prediction. Section 3.5 proposes loss functions to use when the physical quantity to predict depends on the probability distribution of energies of a material.

3.2 Minimum Energy Path Prediction

In this section, we examine the important task of predicting the minimum energy pathway (MEP) of catalyzed reactions. In general, a catalyzed reaction can be broken down into multiple intermediate states, each with its associated energy. Key to understanding the theoretical efficiency of the reaction is the minimum energy path, the most energetically favorable route through these states. Since the MEP includes the rate-limiting step(s) of the reaction, the MEP dictates the overall kinetics.

An illustration of the minimum energy pathway for a cobalt spinel oxide on the oxygen reduction reaction is shown in Figure 3.1. Here there are four steps of the reaction ,*OH, *OO, *OOH, *O (which repeats), and plotted in Figure 3.1a are

the energies of the different intermediates for each step. Figure 3.1b shows two proposed minimum energy pathways for this reaction, where the "new" pathway was only discovered after using DFT to calculate the comprehensive set of energies for all listed intermediates.



(a) Energies of states/intermediates. (b) Two proposed minimum energy pathways.

Figure 3.1: Example MEP for Co_3O_4

We now introduce mathematical notation to describe the minimum energy pathway. In general, we restrict our attention to a single chemical reaction, catalyzed with different materials. Let $\mathcal{M}$ denote the space of catalyst materials. Suppose that the chemical reaction has S steps. Then for each step $s \in S$, let the set of intermediates be I_s. For each intermediate i and material $m \in \mathcal{M}$, there is an associated energy $E(m, i)$. For a material m, the minimum energy pathway is given by the set of minimum-energy intermediates for each step. Specifically, the steps along the minimum energy pathway are $\{\text{argmin}_{i \in I_s} E(m, i) : s = 1, \ldots, S\}$.

The traditional method for calculating the MEP involves evaluating $E(m, i)$ for all intermediates $i \in I_s$ and for all steps S, each requiring a DFT calculation. As

previously mentioned, this is an enormous computational challenge and is generally very slow in practice. Below, we explore an alternate method which reduces the compute required through machine learning.

3.3 Partial MEP Evaluation

Instead of requiring DFT-calculated energies for the entire set of intermediates I_s for each step $s \in \{1, \ldots, S\}$, we propose a method for leveraging machine learning (ML) to select only a subset of intermediates for which to run DFT, resulting in a significant reduction in computation required to calculate the MEP without sacrificing prediction quality. We call this partial MEP evaluation.

In particular, we use an ML model to predict energies, given a material m and intermediate i. We leverage methods from offline contextual bandits to train the ML model so that it produces intermediates which are more likely to fall on the intermediate. As ML models are much faster to evaluate than DFT, this dramatically reduces the compute required to accurately predict the MEP.

3.3.1 Methodology

We propose the following method: we train a ML regression model $f_\theta(m, i)$ which predicts the energies $E(m, i)$ for all materials $m \in \mathcal{M}$, intermediates $i \in I_s$, and steps $s \in \{1, \ldots, S\}$. We call the trained model f_{θ^*}, where θ^* are the parameters of the machine learning model (e.g. neural network weights and biases). Then for each step s, we use f_{θ^*} to predict the N intermediates with the lower predicted energies for that step, and use DFT to evaluate only those intermediates. We then

calculate the minimum energy pathway from only those intermediates for which we have DFT-calculated energies.

Algorithm 1 summarizes this approach. Here $\texttt{TopN}(\cdot)$ refers to the function which returns the N largest elements of the input set, and $\texttt{ComputeMEP}(\cdot)$ simply computes the intermediates which minimize the energy, by step, among the input set.

Algorithm 1: Neural Predictor for MEP Prediction

Input: Material encoding $\mathbf{x}$, evaluation budget N, trained neural network $f_{\theta^\star}$

Output: Estimated minimum energy pathway $\{\hat{i}_{min,s}\}_{s=1}^{S}$

for *step* $s = 1$ **to** S **do**
  for *intermediate* $i = 1$ **to** I_s **do**
    $\hat{E}_{s,i} \leftarrow f_{\theta^\star}(\mathbf{x}, i)$;
  $\Omega_s \leftarrow \texttt{TopN}(\{-\hat{E}_{s,i}\}_{i=1}^{I_s}, N)$;
  Evaluate DFT energies $\{\tilde{E}_{s,i}\}$ for $(s,i) \in \Omega_s$;
$\{\hat{i}_{min,s}\}_{s=1}^{S} \leftarrow \texttt{ComputeMEP}(\{\tilde{E}_{s,i}\}_{s=1}^{S})$;
return $\{\hat{i}_{min,s}\}_{s=1}^{S}$

3.3.2 Loss Functions for Training f_θ

The approach mentioned in the previous section relied on a trained machine learning model f_{θ^*} which generates energy predictions, given a material m and intermediate i. In this section, we describe how this model is trained.

A naive approach for training f_θ is to treat it as a standard ML regression problem, where the features are the material m and intermediate i, and the out-

put is a prediction for the energy $E(m, i)$ (calculated from DFT). In such cases, one generally chooses an accuracy-based loss function such as mean-squared error (MSE), and solve

$$\theta^* \in \underset{\theta \in \Theta}{\operatorname{argmin}} [f(m, i) - E(m, i)]^2. \tag{3.1}$$

However, this choice of loss function neglects the connection between f_θ and the downstream task of predicting the MEP. Intuitively, the model f_{θ^*} only needs to do well at predicting the intermediate with the minimum energy for each step s; it does not need to waste effort in predicting the energies for all intermediates.

We draw on methods from offline contextual bandits to find better loss functions to use for training f_θ. We now cast the problem of training f_θ as an offline contextual bandit. Offline contextual bandits model decision-making problems where an agent chooses an action $a \in A$ when presented with a context $x \in X$, and receives a reward $r(x, a) \in \mathbb{R}$. The goal of the agent is to learn a policy $\pi : X \to A$ that obtains a high expected reward, where the expectation is taken over contexts x. Algorithms for this problem train entirely from static datasets, generally with partial feedback, i.e., only the reward for the action taken is observed.

In the setting of partial MEP evaluation, we view the context x as the material m, the action a as the intermediate i, and the reward r as the negative energy $-E(m, i)$. The training data consists of logged context-action-reward data, for which we only observe rewards (energies) for each chosen action (intermediate) out of many possible. In our application, we make a simplification that all states are observed for each material in the dataset, though this is not generally the case.

Observe that the goal of maximizing the "reward" of $-E(m, i)$ across inter-

mediates $i \in I_s$ exactly aligns with finding the intermediate i_s which lies on the minimum energy pathway for step s. We can then use the predictions from f_θ for each step s to estimate an approximate minimum energy pathway. Therefore we may apply methods from offline contextual bandits for learning a policy $\pi : X \to A$ (or $\pi : X \to \Delta(A)$ in the case of a randomized policy), which predicts the minimum energy intermediate for a given material m and step s.

Under this framework, we consider the class of value-based learning, a common class of algorithms for offline contextual bandits which models the reward as a function of context and action. Instead of using the usual accuracy-based loss function such as MSE, our method, Empirical Soft Regret (ESR), uses a loss function which directly targets the performance of the policy derived from the predicted reward model.

Specifically, let M_{log} denote the set of materials in the logged dataset. The standard value-based approach for offline contextual bandits is to train the regression model f_θ on MSE (just as in Equation (3.1)), i.e.

$$\min_\theta \sum_{m \in \mathcal{M}_{log}} \sum_{s=1}^{S} \sum_{i=1}^{I_s} (f_\theta(m,i) - E(m,i))^2,$$

which aims the neural network at learning the energies accurately. If the set of neural networks $\{f_\theta : \theta \in \Theta\}$ is "well-specified," in the sense that there exists a θ^* which perfectly predicts the energy, then we can expect this loss function to work well when used for MEP prediction. However, this is generally not expected to be true, and when the model is misspecified, then one cannot guarantee its performance on the MEP prediction task.

Instead, we propose the following, which targets the loss function to learning

the minimum energy intermediate:

$$\min_{\theta} \sum_{m \in \mathcal{M}_{log}} \sum_{s=1}^{S} \sum_{i=1}^{I_s} E(m,i) \frac{\exp(f_\theta(m,i))}{\sum_{i'=1}^{I_s} \exp(f_\theta(m,i'))}.$$

Observe that this is the same as the ESR loss from Chapter 2 up to a constant, and with $k = 1$. Therefore we know from Chapter 2 that this benefits from theoretical guarantees on its performance on predicting the minimum energy intermediate. We also demonstrate the effectiveness empirically in the next section.

3.4 Numerical Experiments

We apply our methodology to the Oxygen Reduction Reaction (ORR):

$$\mathrm{O_2 + H_2O + 4\,e^- \longrightarrow 4\,OH^-}$$

The ORR is central in many electrochemical processes, including fuel cells. We apply our method on a dataset of DFT-calculated energies for the ORR, which consists of 245 spinel oxides, where there are 27 intermediates for each of four steps in the reaction.

To evaluate performance, we consider the error in predicting the minimum energy barrier, which is the largest positive difference in energy between two consecutive steps. In practice, the minimum energy barrier captures most of the physical characteristics of the chemical reaction since it identifies the rate-limiting step, i.e. the most energetically difficult part of the transition between two steps.

Figure 3.2 shows the behavior of the error on the minimum energy barrier vs. the number of intermediates N to evaluate per step. We find that ESR outperforms

standard benchmarks, requiring 40% fewer DFT evaluations to achieve threshold accuracy of 0.1 eV (the accepted accuracy of DFT calculations [Hinuma et al., 2017]) , as opposed to 20% fewer in the competing stochastic approach.

Here the baselines are:

- Mean prediction: predict the average MEP barrier within the training dataset
- Intermediate energy model: train an ML model on individual intermediate energies and directly use predictions to compute the MEP barrier.
- Composition only model : train a regression model that directly maps the elemental composition to the MEP barrier.
- Stochastic partial evaluation: randomly sample N intermediates per step.
- Full evaluation: use DFT to calculate all 27 intermediate energies for each step.

Note the first three baselines do not collect additional DFT evaluations and therefore are constant with N.

We also ran an experiment with the same neural network architecture but trained with MSE, as in traditional value-based learning. The performance of this method was statistically identical to that of the stochastic partial evaluation, i.e. the energy predictions provided no additional value relative to random sampling.

3.5 Distributional Loss Functions for Energy Prediction

In statistical mechanics, one assumes that for a fixed temperature, the probability distribution of a material's configurations follows the Boltzmann distribution, with

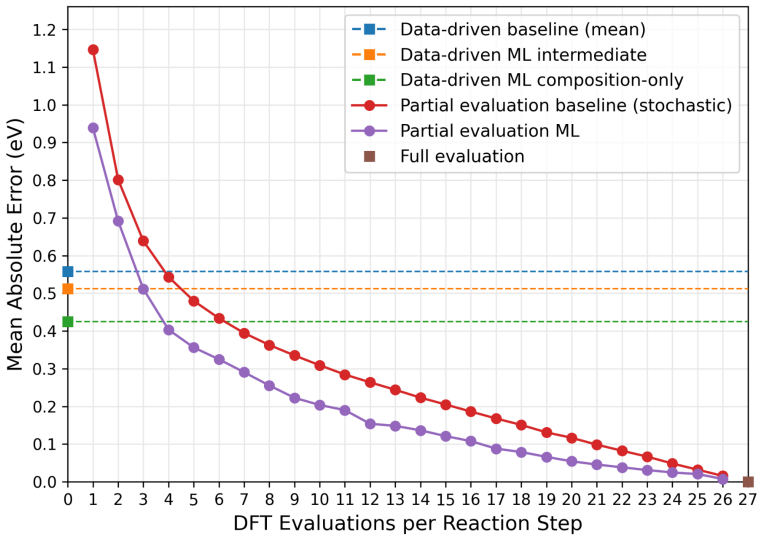


Figure 3.2: Mean absolute error of barrier prediction vs. number of evaluations.

probabilities proportional to the energy. From this distribution, one can derive many physically meaningful properties, such as the heat capacity, magnetization, and susceptibility.

Given a proposed material, the current method for predicting these quantities is to use a model to predict the energy based on the configuration, and use these predicted energies to derive a probability distribution, and finally to derive the quantity of interest from this distribution. Although this method works well if the model class used to predict the energy is well-specified, in general the function describing the energy as a function of the configuration is assumed to be complex, making well-specification difficult to achieve.

Instead of using a model which is trained on predicting the energy accurately, we propose a model which trains on a different loss function which aims to learn the downstream Boltzmann distribution well, as this distribution ultimately determines the various quantities of interest. In particular, we propose a methodology which minimizes a distance between a predicted energy distribution and

the observed energy distribution, where the distance can be any distance between probability distributions.

3.5.1 Statistical Mechanics Refresher

Statistical mechanics posits that, for each value of energy E, a material can be in one of many configurations, each having that energy level. The density of states $D(E)$ is a function which represents the number of configurations for a given energy E.

One additionally assumes that at a fixed temperature T, the probability that a material is in a particular configuration i with associated energy value E_i is proportional to $\exp(\frac{-E_i}{k_B T})$, where $k_B = 8.617333262 \cdot 10^{-5}$ eV K^{-1} is a universal constant known as the Boltzmann constant. The normalizing constant $\sum_i \exp(\frac{-E_i}{k_B T})$ is known as the partition function Z.

From this distribution, we can derive the following physical quantities

- Internal energy is the expected value of the energy:

$$U(T) = \frac{1}{Z} \sum_i E_i \exp\left(\frac{-E_i}{k_B T}\right)$$

- Heat capacity is proportional to the variance of the energy:

$$C_v(T) = \frac{1}{k_B T^2} \left(\frac{1}{Z} \sum_i E_i^2 \exp\left(\frac{-E_i}{k_B T}\right) - \left(\frac{1}{Z} \sum_i E_i \exp\left(\frac{-E_i}{k_B T}\right) \right)^2 \right)$$

- The critical temperature $T_c := \mathrm{argmax}_T \, C_v(T)$,

among many others.

3.5.2 Methodology

The current approach for predicting the distribution of energies consists of three steps.

1. First, we train an energy model $\hat{E}_\theta(x)$ which predicts a material's energy E given its configuration x. For example, one can train a cluster expansion (CE) model, which fits a linear regression on a set of features derived by considering p-body interactions between the atoms in the material [Rosenbrock et al.]. In general (as is the case in linear regression), the loss function used is based on predicting the energy accurately, e.g. mean-squared error:
$$\sum_{i=1}^{n} \left(\hat{E}_\theta(x) - E_i \right)^2 .$$
2. After a trained energy model $\hat{E}_\theta$ is obtained, $\hat{E}_\theta$ is then used in an algorithm which samples from the density of states based on the predicted energies $\hat{D}_\theta(E)$; examples include Metropolis-Hastings or Wang-Landau [Wang and Landau, 2001]. In the case of Wang-Landau specifically, the output of this step is (1) an approximate distribution of the energies for this material across all configurations, where this histogram approximates $\hat{D}_\theta(E)$, and (2) a set of visited configurations and their energies, where the distribution of visited energies is roughly uniform. See Chapter A for details.
3. Finally, after this set of configurations is generated, then the distribution of energies for any temperature T is fully characterized, and the physical quantities mentioned in the previous subsection can be estimated based on this distribution.

We replace the loss function used to train the energy prediction model in step 1

with a different loss function, which accounts for the downstream task of generating a distribution. For the following, we first suppose that the dataset is comprised of (x_i, E_i) for $i = 1, \ldots, n$, where x_i are the configurations and E_i the energies, and all possible configurations of the material are given.

First, we choose a temperature T at which to compute the distribution. In practice, this should be chosen according to domain knowledge about what temperature range would be important in determining the physical quantity of interest. Then, we define a probability distribution P given by the energies in the training dataset:

$$\mathbb{P}_P(E = E_i) = \frac{\exp(-\frac{E_i}{k_B T})}{\sum_{j=1}^{n} \exp(-\frac{E_j}{k_B T})}.$$

Then, given a model in our model class $\hat{E}_\theta$, we define a predicted probability distribution $\hat{P}_\theta$ over the dataset:

$$\mathbb{P}_{\hat{P}_\theta}(E = E_i) = \frac{\exp(-\frac{\hat{E}_\theta(x_i)}{k_B T})}{\sum_{j} \exp(-\frac{\hat{E}_\theta(x_j)}{k_B T})}.$$

Then our loss function is simply $D(P||\hat{P}_\theta)$ where $D(P||Q)$ is a measure of statistical distance between two probability distributions. We call this loss function the Canonical Ensemble Loss (CEL). Examples include Kullback-Leibler (KL) divergence, Jensen-Shannon divergence, and total variation distance. We found the KL divergence to work best; it is defined as

$$D_{KL}(P||Q) := \sum_{i=1}^{m} P(E_i) \log \frac{P(E_i)}{Q(E_i)}.$$

Therefore our loss function is

$$\sum_{i=1}^{n} \frac{\exp(-\frac{E_i}{k_B T})}{\sum_{j=1}^{n} \exp(-\frac{E_j}{k_B T})} \log \frac{\frac{\exp(-\frac{E_i}{k_B T})}{\sum_{j=1}^{n} \exp(-\frac{E_j}{k_B T})}}{\frac{\exp(-\frac{\hat{E}_\theta(x_i)}{k_B T})}{\sum_j \exp(-\frac{\hat{E}_\theta(x_j)}{k_B T})}}. \tag{3.2}$$

In practice, however, it is very rare to have all configurations of material available in a dataset. In particular, even simple structures like lattices have combinatorially many configurations, and enumerating them all is computationally expensive.

Instead, one can use Wang-Landau as a subroutine before step (1) to first generate samples of configurations uniformly across the range of energies, where for each energy level E we also have an approximate density of states $D^0(E)$. Of course, this step also requires having an approximate energy predictor $\hat{E}_{\theta^0}$, trained an initial batch of data.

In this way, we can compute a dataset $x_i, E_i^0, D_0(E_i^0)$ of configurations x_i, estimated energies E_i^0 energies and density of states $D_0(E_i^0)$. Then to approximate (3.2), we weigh the probabilities by these approximate density of states accordingly:

$$\mathbb{P}_P(E = E_i^0) = \frac{D^0(E_i^0) \exp(-\frac{E_i^0}{k_B T})}{\sum_{j=1}^{n} D^0(E_j^0) \exp(-\frac{E_j^0}{k_B T})},$$

and

$$\mathbb{P}_{\hat{P}_\theta}(E = E_i^0) = \frac{D^0(E_i^0) \exp(-\frac{\hat{E}_\theta(x_i)}{k_B T})}{\sum_j D(E_j^0) \exp(-\frac{\hat{E}_\theta(x_j)}{k_B T})}.$$

The loss function corresponding to (3.2) is then adjusted accordingly:

$$\sum_{i=1}^{n} \frac{D^0(E_i^0)\exp(-\frac{E_i^0}{k_BT})}{\sum_{j=1}^{n} D(E_j^0)\exp(-\frac{E_j^0}{k_BT})} \log \frac{\frac{D^0(E_i^0)\exp(-\frac{E_i^0}{k_BT})}{\sum_{j=1}^{n} D^0(E_j^0)\exp(-\frac{E_j^0}{k_BT})}}{\frac{D^0(E_i^0)\exp(-\frac{\hat{E}_\theta(x_i)}{k_BT})}{\sum_j D^0(E_j^0)\exp(-\frac{\hat{E}_\theta(x_j)}{k_BT})}}. \tag{3.3}$$

In summary, our proposed methodology is follows.

1. Given an initial sample of a material's configuration x_i and ground-truth energies E_i, use an initial class of models Θ^0 to fit an approximate energy model $E^0(x)$.
2. Use E^0 in Wang-Landau to generate (1) a set of configurations x_i and energies $E_i^0 := E^0(x_i)$ such that the distribution of the E_i^0 is uniform across the range of energies and (2) an approximate density of states $D^0(E)$.
3. On this dataset of x_i, E_i^0, D_i^0, minimize (3.3) over a class of models Θ, e.g. linear models or neural networks to yield an energy model $\hat{E}_{\theta^*}(x)$
4. Use $\hat{E}_{\theta^*}(x)$ to generate a refined estimate for the density of states $\hat{D}^{\theta^*}(E)$.
5. Calculate the physical quantity of interest, which involves constructing a distribution of energies with $\hat{D}^{\theta^*}$, varying temperature.

3.5.3 Preliminary Numerical Experiments on Rock Salts

We conduct numerical experiments on configurations of the MgO (magnesium oxide) rocksalt, where 50% of the Mg atoms are replaced with Fe (iron). To have a metric which is a single scalar, we focus on predicting the critical temperature T_c.

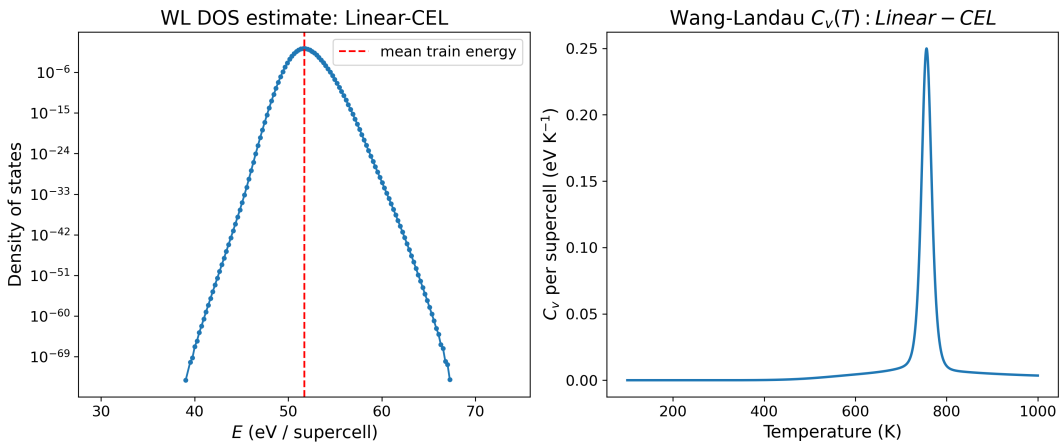


Figure 3.3: Density of states and heat capacity from CEL-trained linear model. Estimated $T_c = 756$ K

We focus on linear models based on CE; in particular, suppose that x represents the features that are calculated from the expansion. We consider energy models $\hat{E}_\theta(x) = \theta^\top x$, which we expect to be misspecified for the true energy function E. We also use MACE [Batatia et al., 2022] as a proxy for DFT-calculated ground truth energies for computational efficiency.

For both models, 500 samples were used to first train a least squares model to generate a uniformly distributed energy dataset, also of size 500, as described in step 1 of section 3.5.2. We then use this second batch to train linear models, minimizing the two loss functions. In particular, the CEL loss function also takes the density of states as input. Also, our CEL loss used $T = 700$, based on domain knowledge suggesting that $[600, 800]$ is an important temperature range for understanding the physical properties of this material.

Figures 3.3 and 3.4 show results from linear models trained on CEL and a standard linear model trained on minimizing least squares. We see both models are essentially in agreement, including on the critical temperature (756 K vs 763K), suggesting the CEL is doing reasonably well in comparison to the standard method.

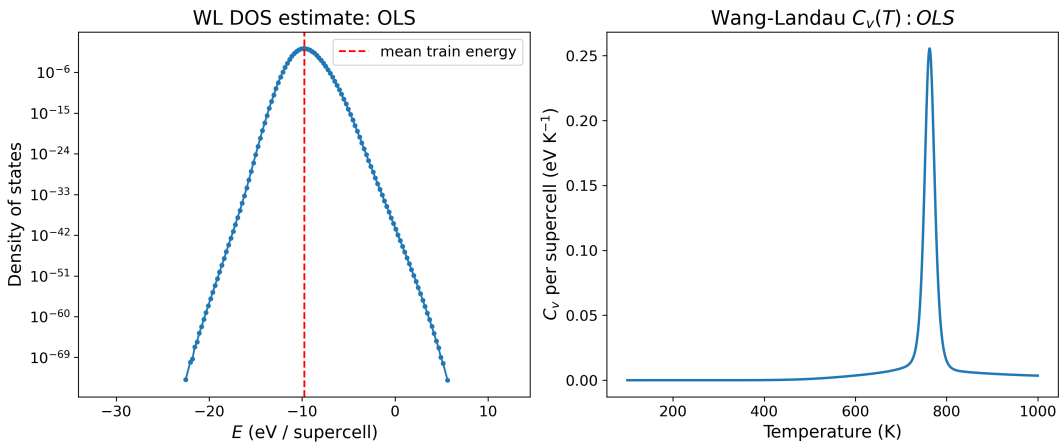


Figure 3.4: Density of states and heat capacity from MSE-trained linear model. Estimated $T_c = 763$ K

For reference, Wu et al. [1993] calculated the critical temperature for this particular material to be 612 K from experimental data. However, this was calculated using various quasichemical assumptions and should not be treated as ground truth.

We view these preliminary results are promising in that our method seems to be doing as well as the status quo, and we believe there are many avenues to improve upon our methodology.

3.6 Conclusion

This chapter demonstrates that moving beyond traditional accuracy-based loss functions like mean squared error can significantly improve the performance of neural interatomic potentials for downstream tasks in computational materials science. Through the application of Empirical Soft Regret loss to minimum energy pathway prediction, we showed that framing the problem as an offline contextual bandit and directly optimizing for the policy objective rather than energy prediction accuracy leads to substantial improvements in identifying rate-limiting

steps with fewer DFT evaluations. Similarly, our Canonical Ensemble Loss demonstrates the potential for training energy models that better capture the Boltzmann distributions underlying thermodynamic properties, even when working with misspecified model classes. While the preliminary results on rock salt systems show promise, with critical temperature predictions comparable to standard methods, future work should explore more complex materials and investigate the theoretical guarantees of CEL in the context of statistical mechanics. Together, these contributions highlight that task-specific loss function design represents a new way to leverage neural interatomic potentials to improve the efficiency and accuracy of computational materials discovery.

Part II

Optimizing Large-scale Logistics

CHAPTER 4

MILITARY LOGISTICS PLANNING FOR EXPEDITIONARY WARFARE

This chapter is based on joint work with J. Haden Boone, Matthew Ford, Mathieu Dahan, Peter Frazier, Yongjia Song, and Huseyin Topaloglu.

4.1 Introduction

In recent years, the United States Marine Corps (USMC) has undergone extensive force restructuring under the *Force Design* plan to enhance its capabilities and reshape its combat power for potential peer-level competition in the Indo-Pacific region, as directed by the 2018 National Defense Strategy [Mattis, 2018, Berger, 2020]. The plan's key goals are for the USMC's forces to become more agile, versatile, as well as amphibious by working more closely with the United States Navy. These changes are designed to ensure that the USMC remains an effective expeditionary force, ready to support naval and joint operations in a variety of global scenarios.

Expeditionary warfare involves military operations carried out by forces deployed to foreign territories, typically in hostile or contested environments. It is characterized by the rapid deployment of expeditionary forces that are specially trained and equipped to operate independently of established support bases. These forces are capable of executing a wide range of missions, including combat operations, humanitarian assistance, and peacekeeping [United States Marine Corps, 2023]. This form of warfare is crucial for maintaining global stability and addressing crises efficiently, as it enables military forces to respond swiftly to emerging

threats, secure strategic locations, and support allies or deter adversaries [McKnight, 2019].

However, rapidly deploying and sustaining expeditionary forces presents significant logistics challenges, primarily because such forces operate far from established supply networks and infrastructure. One of the key challenges is coordinated and timely transport of essential supplies, including food, fuel, and ammunition, to remote or contested areas. The complexity of logistics planning is heightened by limited ability to rely on local resources and the need to transport supplies over long distances, often requiring a combination of air, sea, and ground transportation [Bradford, 2006]. Additionally, the dynamic nature of expeditionary operations demands a high degree of flexibility and adaptability in logistics planning. Planners must account for rapid changes in the operational environment, such as shifting frontlines, evolving threats, and variable weather conditions [Apte, 2018]. These factors can disrupt supply lines and complicate the scheduling and routing of logistics convoys, requiring contingency planning and the ability to quickly reallocate resources.

Overall, the logistical challenges in sustaining expeditionary operations are multifaceted and require rapid and meticulous planning capable of responding to the uncertainties inherent in expeditionary warfare. However, the current analysis manually conducted by USMC logistics planners during training exercises remains qualitative and labor-intensive. While there have been efforts toward using simulation and predictive analytics tools to estimate logistics requirements during an expeditionary mission, there has been no prescriptive analytics work on routing and scheduling logistics assets, which is vital for guaranteeing mission success [Reith et al., 2016, de Castro et al., 2023]. On the other hand, the operations research

models on scheduled service network design for logistics planning are not applicable to expeditionary settings, as they do not account for critical features such as flexible sourcing and heterogeneity in connector characteristics [Crainic and Hewitt, 2021]. As a result, this research aims *to develop the first prescriptive analytics tool to assist the USMC with efficient expeditionary logistics planning.*

To design this decision-support tool, we formulate a mixed-integer program (MIP) to optimize the routing and scheduling of connectors and commodities within an intermodal expeditionary logistics network. The MIP aims to minimize penalties for delayed fulfilled demand, while accounting for critical features such as connector capacities, travel and maintenance times, location capacities, and commodity availability. MIP model that incorporates realistic logistical constraints and objectives. We then test our model on a fictitious expeditionary scenario in Southern California designed to reflect USMC training scenarios. Our results show the practical applicability and effectiveness of the proposed MIP in coordinating logistics assets to fulfill demands in a timely manner. Via key performance measures, our tool provides valuable insights into the optimal utilization of logistics assets. By conducting experiments with what-if scenarios, we also demonstrate its capability to rapidly adjust logistics plans, while identifying vulnerabilities in the logistics network. Importantly, our tool has been utilized and validated by the USMC during classified tabletop exercises.

4.2 Literature Review

4.2.1 Military Airlift

The MIP formulation used in this work closely resembles the formulations used in the military airlift literature [Morton et al., 1996, Rosenthal et al., 1997, Killingsworth and Melody, 1997, Baker et al., 2002]. Military airlift is the problem of routing cargo and troops through a logistic network using a fleet of aircraft, where the routing must obey certain constraints due to physics (weight capacities) and/or policies (crew rest/flight hour requirements). The basic formulation used in this line of work is the same; specific constraints were added over time depending on the application. In the basic formulation, the logistics problem is solved in discrete time, where aircraft and cargo depart from their respective origins and the aircraft must carry cargo to satisfy demand at designated demand destinations. The main decision variables index over the type of commodity, the vehicle (aircraft), route (one of a precomputed set of routes), and time step at which the vehicle begins the route. The objective is the sum of unmet/late demand as well as other secondary penalties/rewards for deadheading crews, minimizing plane use, etc. These models have been used to provide insights to the US Air Force [Killingsworth and Melody, 1997, Baker et al., 2002] as well as support operations in Iraq and Afghanistan [Brown et al., 2013].

The formulation used in military airlift has several notable differences compared to our formulation. One such difference is that their movement variables index over routes between an origin and a destination, where a route may visit multiple airfields in between. In contrast, our movement variables index over arcs defined by pairs of locations. The use of the route-based formulation is a consequence of the

small number of routes considered in realistic instances, e.g. Rosenthal et al. [1997] used an instance with 313 routes, while the largest number of routes mentioned in a test instance in Baker et al. [2002] was 289. In their case, the space of routes is reduced significantly due to the small number of unique origins and destinations and assumptions about what reasonable routes look like due to domain knowledge, e.g. given an instance of delivering from the West Coast to South Korea, domain knowledge dictates stops through Alaska (if going through the North Pacific) or Hawaii/Guam/Okinawa (if going through the Mid-Pacific) before a second stop in Japan [Baker et al., 2002]. In contrast, we consider scenarios in which there may be exponentially many paths to consider due to the number of different naval bases and demand locations, as well as the fact that back-and-forth trips are allowable and often expected. Therefore an arc-based formulation suited our application so that the number of variables did not explode.

Another important difference is the focus on aircraft and the associated constraints based on aircraft-specific policies. For example, one such constraint is that the flight crew must take 12 hours of rest after 16 hours of flight time [Baker et al., 2002]. Although we do not track this constraint exactly, we approximate this by including a delay time for each arc that a vehicle travels, which is meant to include the time that is required for crew rest and turnaround in the case of an aircraft. Specifically, in our formulation, we can enforce a more conservative version of this behavior by disallowing any arcs over 16 hours and including a 12 hour delay for any traveled arc. There are other minor policy constraints, e.g., involving aircraft utilization and aerial refueling that are modeled in military airlift; we do not model such constraints here, though we could approximate such behavior by adding simple constraints on cumulative arcs traveled and refueling locations, respectively.

Finally, this line of work [Morton et al., 1996, Rosenthal et al., 1997, Killingsworth and Melody, 1997, Baker et al., 2002] solves the formulation as an LP due to computational constraints (at the time) preventing the use of a MIP. Due to advancements in integer programming methods since the publication of those works, we are able to solve our formulation as a MIP, yielding solutions that are more practically useful for informing decisions. In particular, it is easy to come up with an instance in which solving an LP yields no unmet demand, whereas solving a MIP does (e.g., when a vehicle is allowed to split in half to satisfy demands at two different locations); see the proof of Theorem 5.5.2 for such an example.

Beyond airlift, there is also a similar line of work on Navy-specific integer programming formulations for ships [Borden, 2001, Powell, 2004, Brown and Carlyle, 2008, Brown et al., 2017, Brown and Kline, 2021], where the problem is to optimize the routing of the US Navy's combat logistics force (CLF), comprised of only about 30 ships of three distinct types, to supply cargo and fuel to combatant ships. The methods in these works differ in whether the movement index over routes (called "voyages") [Brown et al., 2017] as in the airlift literature, or arcs between locations [Borden, 2001, Powell, 2004, Brown and Carlyle, 2008, Brown and Kline, 2021] as in our approach. In any case, the problem of optimizing the CLF is significantly smaller in scope; we consider logistics for expeditionary warfare, which encompasses a much larger logistics network involving all modes of transportation utilized by the Marine Corps and the Navy (air, sea, and ground). Note that an important feature of the CLF logistics problem is that the destination nodes are ships that move over time; we also naturally handle this case since our travel times between locations are allowed to vary over time.

4.2.2 Expeditionary Logistics Planning

Traditional methods used for expeditionary logistics planning heavily rely on manual planning and heuristic approaches. These methods involve logistics planners using their expertise, intuition, and simple heuristic rules to develop logistics plans [United States Marine Corps, 2022]. Although such heuristics reduce decision-making complexity, they have been shown to often lead to biased and suboptimal solutions [Williams, 2010, Joint Chiefs of Staff, 2011].

In recent years, the military has invested in data-driven tools to make more informed logistics decisions. For instance, the Operational Logistics (OPLOG) Planner is a demand planning tool used to estimate logistics requirements for classes of supply needed to support a unit's mission. It calculates requirements based on unit data and operational variables such as terrain, weather, and mission phases [Chavers, 2009]. Similarly, the Logistics Estimation Workbook (LEW) is an Excel-based tool that uses data from combat profiles and usage rates to provide quick forecasts for supply, transportation, and maintenance requirements [Green, 2016]. However, these tools do not determine how to feasibly transport supplies using available connectors in order to meet those requirements on time, which is critical for assessing the feasibility of a maneuver course of action (COA) [Schwartz et al., 2019].

The closest work related to ours is the Military Logistics Network Planning System (MLNPS) developed by Rogers et al. [2018]. The MLNPS is a simulation-based system that represents the expeditionary logistics network as a Virtual Factory. In the simulation model, supply requisitions are associated with already determined sourcing locations and transportation modes and routes. Viewed as a queuing system, the model determines at each discrete event which supplies are loaded

on a vehicle based on their revised slacks (i.e., maximum time they can wait at their locations and still meet their due dates). This simulation model is used to estimate the performance of logistics plans and highlight choke points, allowing logistics planners to shift or add transportation assets to improve performance. While this proof-of-concept has shown valuable promise, the decision process remains restricted, as routes have already been determined and load sequencing is based on a heuristic rule. In contrast, we propose an optimization-based tool that jointly determines the routing and scheduling of connectors and commodities, including commodity sourcing. Such complex intermodal logistics plans achieve a higher level of efficiency, thereby enhancing the capabilities of maneuver COAs.

4.2.3 Service Network Design

The logistics planning problem faced by the USMC during expeditionary warfare exhibits similarities with the service network design problem for flow and load planning encountered by commercial consolidation trucking carriers [Crainic, 2000, Wieberneit, 2008, Bakir et al., 2021]. Initial research in trucking service network design focused on flat (static) network models to determine the optimal number of transportation vehicles dispatched between facilities weekly [Powell and Sheffi, 1983, Crainic et al., 1984, Powell, 1986, Crainic and Rousseau, 1986, Crainic and Roy, 1988, Powell and Koskosidis, 1992, Greening et al., 2023]. However, such formulations are not applicable to expeditionary logistics, as they do not explicitly model the scheduling of vehicle dispatches, which is essential for determining a feasible logistics plan to sustain expeditionary forces over a short time horizon.

Recent advancements in commercial flow and load planning have introduced more detailed time-expanded network models, known as scheduled service network

design problems, which explicitly account for the timing of vehicle dispatches. Variants of such problems, including those addressing empty resource management, stochastic shipment volumes, and travel times, have been explored [Lin, 2001, Pedersen et al., 2009, Andersen et al., 2009, Lium et al., 2009, Bai et al., 2014, Zhu et al., 2014, Crainic et al., 2016, Demir et al., 2016, Scherr et al., 2019, Wang and Qi, 2020]. However, high-quality logistics plans often necessitate fine time discretization to accurately capture consolidation and synchronization opportunities, resulting in very large and challenging MIPs to solve. Heuristic solution methods have been developed to tackle large-scale instances [Jarrah et al., 2009, Erera et al., 2013, Lindsey et al., 2016], while other work has explored dynamic approaches to determine exact dispatch times, eliminating the need for predetermined time discretization [Boland et al., 2017, Hewitt, 2019, Boland et al., 2019, Scherr et al., 2020, Marshall et al., 2021, Hewitt, 2022]. However, such formulations cannot be used to model expeditionary logistics operations, as they typically assume homogeneous vehicles, fixed origin-destination pairs, and aim to minimize logistics costs. In contrast, expeditionary logistics planning requires dispatching sea, air, and land connectors with heterogeneous characteristics (speed, range, capacity), and primarily aims to fulfill demand on time, possibly from multiple locations. *To the best of our knowledge, we propose the first scheduled service network design model for expeditionary logistics planning.*

4.3 USMC Expeditionary Logistics

Our approach is focused on decision support for expeditionary logistics, focused on supporting a commander's choice of a course of action (COA) and then making high-level logistics plans for closing the force and supporting that force during

execution of the chosen COA.

Current logistics decision support for COA selection has many manual aspects. Logistics officers do use tools like Excel to do arithmetic computations and lay out plans but comprehensive decision-support tools are not currently available. The logistics officer understands the commanding officer's intent and the broader planning context and then creates a collection of tentative plans over the period of a few hours. That logistics officer then creates a presentation, e.g., using PowerPoint, with several high-level plans for supporting the proposed COA(s) and briefs the commanding officer.

Because of the short timeline, the lack of full automated decision support, and difficulty in obtaining accurate real-time information about inventory and connector capabilities, these plans are typically approximate. The plans may be quite conservative, in the sense of proposing smaller or slower personnel and commodity flows than would actually be feasible. There is also a risk that the proposed flows are not feasible.

4.4 Heuristic Approaches

4.4.1 Automatic Hub-and-Spoke Generation

To reduce the memory footprint and runtime of the MIP, we use a hub-and-spoke network to sparsify the set of arcs. Commonly used in the airline industry [Bryan and O'kelly, 1999], the hub-and-spoke network designates certain centralized locations as "hubs" and partitions the remaining locations ("spokes") according to

which hub is closest. In such a network, only (1) the arcs between a hub and its corresponding spokes and (2) all arcs between hubs are kept; this dramatically reduces the total number of arcs in the network.

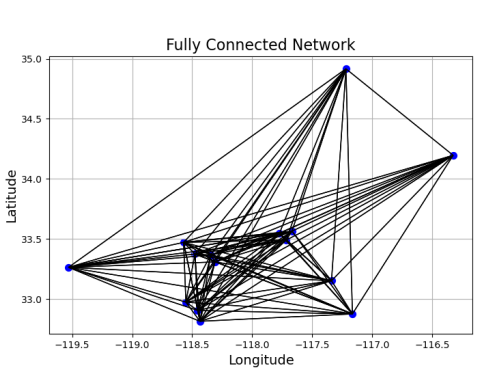
Given an instance, one can construct a hub-and-spoke network manually by letting a user specify the hubs and spokes. However, this places an additional burden on the user and allows more room for user error. Instead, we introduce methodology which leverages machine learning to automatically generate the hub-and-spoke network. In particular, we use the K-medoids clustering algorithm [Park and Jun, 2009].

Clustering is an unsupervised learning technique that groups similar data points together based on their features, partitioning the data into distinct clusters where points within each cluster are more similar to each other than to points in other clusters. Typically, each cluster contains a representative point; common choices include the centroid, as in K-means [Likas et al., 2003], or the medoid, as in K-medoids. In our application, we define the similarity as geographical distance, and let the medoids serve as "hubs" and the other points in their surrounding cluster as "spokes." Note that in such methods, typically the number of clusters K is a hyperparameter chosen by the practitioner.

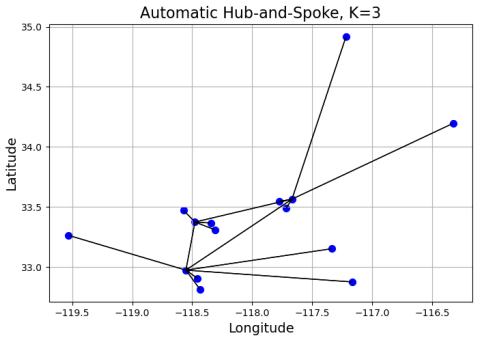
Because distances between pairs of locations can vary depending on the routing domain (sea, air, or ground), we construct one hub-and-spoke network per domain. This allows the sparsification of the network to depend on the domain; in particular, ground networks are generally more sparse than that of sea or air and therefore may not require additional sparsification through the hub-and-spoke design.

Figure 4.1 shows an example of the Southern California scenario before and

after applying the automatic hub-and-spoke network generation. The hub-and-spoke network has many fewer arcs than the original fully connected network.



(a) Original Network (fully connected)



(b) Automatically generated hub-and-spoke for $K = 3$

Figure 4.1: The result of applying automatic hub-and-spoke to the Southern California scenario.

After a solution is generated using the hub-and-spoke network, we also "polish" the solution by collapsing suitable arcs which go through hubs, as the network is not an actual constraint on the true movement of connectors. In particular, if a connector travels on arc (ℓ_1, ℓ_H) to a hub ℓ_H then on a second arc (ℓ_H, ℓ_2) carrying the exact same load of commodities along the way, we replace these two arcs with a single arc (ℓ_1, ℓ_2), adjusting the M, I, and Z variables accordingly.

4.5 Optimization-Based Logistics Planning

We consider a military logistics planner aiming to provide logistics support to an operational plan. Over the course of a time horizon, the logistics planner must transport commodities from their storing locations to satisfy demand requested by the operational planner at given locations (e.g., advanced bases) by using available connectors (i.e., transportation vehicles). We develop a decision-support tool based

on a mixed-integer program (MIP), for which the detailed formulation is described in Section 4.6.

Specifically, we consider a military logistics network consisting of warehouses, transportation hubs (e.g., railheads, sea and air ports), and advanced bases. In this network, we aim to design a logistics plan over a time horizon, discretized into finite time periods (e.g., every six hours). We assume that the operational planner has emitted a set of requests to be satisfied by the logistics planner. Each request is characterized by quantities of each commodity type (e.g., fuel, ammunition) that must be fulfilled by certain time periods. Requests include commodities for initial assaults and sustainment throughout the time horizon. To fulfill these requests, commodities must be transported from their storing locations via connectors of various types (e.g., railcar, landing craft, tanker aircraft). Each connector type faces a payload capacity, measured in terms of the number of units of each commodity type, as well as the total commodity volume and weight, it can carry. In addition, each location faces capacity constraints regarding the maximum amount of commodities it can store, as well as the maximum number of connectors of each type it can hold at any given time period. At the beginning of the planning horizon, the logistics planner is aware of the time period and geographical location at which each connector and commodity become available.

The main decisions of the logistics planner then consist of routing the connectors and commodities throughout the logistics network. To reduce the number of variables in the MIP, we aggregate the decisions per commodity type and connector type. Specifically, we decide for each geographical location and at each time period the number of connectors of each type that depart to travel to another location. Similarly, at each location and time period, we decide the number of each commod-

ity type that depart on each connector type traveling to another location. While these variables keep track of the movement of the commodities and connectors, we also need to define variables to keep track of the number of commodities and connectors on ground at each location and time period. Finally, since requests may not be satisfied on time, we use decision variables that keep track of the amount of commodities of each type satisfied (resp. still to be satisfied) for each request, location, and time period.

We model the objective of the logistics planner as minimizing the penalties for delayed request fulfillment. Specifically, we assume that for each request, a penalty is added for each unit of unmet commodity demand at each location and time period. The goal is to select a logistic plan that minimizes this objective function, while satisfying all the physical constraints of the system. In particular, the logistics plan must ensure that connector and location capacities are not exceeded. Furthermore, at each location and time period, the number of connectors and commodities departing cannot exceed the available amount. Importantly, conservation constraints are needed to keep track of the number of commodities and connectors of each type at each location and time period. For connectors, such constraints account for the connectors (i) stationed at the same location at the preceding time period, (ii) becoming available at that location and time period, (iii) leaving during that time period, and (iv) arriving from another location, given their travel time and the required processing and maintenance time on ground. Similarly, commodity conservation constraints account for the commodities previously available, arriving, leaving, and used to satisfy requests at each location and time period.

We note that this modeling framework permits to account for mobile logistics locations, such as large war units that may move throughout the planning horizon

as dictated by the operational plan. Furthermore, it can model some locations being closed off to some or all connector types at given time periods (e.g., due to extreme weather events). Finally, the penalties may vary with the amount of delay, to model aggravating situations as a result of missing commodities.

4.6 Mixed-Integer Programming Formulation

In this section we describe in detail the mixed-integer programming formulation of the military logistics planning problem presented in Section 4.5. The planning horizon is discretized into T time periods. We denote as $\mathcal{L}$ the set of logistics locations, $\mathcal{J}$ the set of commodity types, and $\mathcal{H}$ the set of connector types. The operational planner emits a set of requests $\mathcal{I}$, where each request $i \in \mathcal{I}$ specifies a set of demands $D_{ij\ell t}$ for each commodity of type $j \in \mathcal{J}$ at each location $\ell \in \mathcal{L}$ by each time period $t \in [\![1, T]\!]$. Each unit of commodity type $j \in \mathcal{J}$ occupies Q_j units of capacity (e.g., volume, weight) and each connector of type $h \in \mathcal{H}$ has a capacity given by A_h. Furthermore, during each time period, each location $\ell \in \mathcal{L}$ has a total commodity capacity given by B_ℓ and can hold up to $C_{h\ell}$ units of each connector type $h \in \mathcal{H}$. Commodities and connectors may become available for routing at different time periods; we then denote as $V_{h\ell t}$ and $W_{j\ell t}$ the respective number of connectors of type $h \in \mathcal{H}$ and commodities of type $j \in \mathcal{J}$ that newly become available at location $\ell \in \mathcal{L}$ at time period $t \in [\![1, T]\!]$.

When a connector of type $h \in \mathcal{H}$ departs from a location $\ell \in \mathcal{L}$ at time period $t \in [\![1, T]\!]$, we let $N_{h\ell kt}$ denote the number of time periods needed to reach location $k \in \mathcal{L}$. We note that this parameter accounts for the speed of each connector type and the distance between pairs of locations, which may be varying over the

planning horizon. This feature is useful to model instances where a sea connector may adjust its speed when facing difficult maritime conditions, or where a location is itself mobile along a prefixed route. When a connector of type $h \in \mathcal{H}$ arrives at a location $k \in \mathcal{L}$, we assume it can immediately unload its cargo, but must undergo some maintenance for M_{hk} time periods before being able to depart. In addition, we denote as $\mathcal{L}_{h\ell t}$ the set of locations that a connector of type $h \in \mathcal{H}$ can travel to when departing from location $\ell \in \mathcal{L}$ at time period $t \in [\![1, T]\!]$. This set encompasses travel restrictions and infeasibility due to weather, traveling range, or closing of locations.

To model the objective of the logistics planner, we denote as $P_{ij\ell t}$ the penalty incurred by the logistics planner for each demand unit of commodity type $j \in \mathcal{J}$ in request $i \in \mathcal{I}$ that is unmet by time period $t \in [\![1, T]\!]$ at location $\ell \in \mathcal{L}$. The goal of the logistics planner is to minimize the total penalty incurred. We note that $P_{ij\ell t}$ can account for request and commodity priorities, and may vary with the amount of delay. As delay increases, the per unit penalty can be increased and even be set to a large value if the demand has not been met by the end of the time horizon.

The decisions of the logistics planner can be described as follows: For each connector type $h \in \mathcal{H}$, each starting location $\ell \in \mathcal{L}$, each time period $t \in [\![1, T]\!]$, and each destination location $k \in \mathcal{L}_{h\ell t}$, $x_{h\ell kt} \in \mathbb{Z}_{\geq 0}$ represents the number of connectors of type h departing location ℓ at time period t to travel to location k. Similarly, for every commodity type $j \in \mathcal{J}$, $y_{jh\ell kt} \in \mathbb{R}_{\geq 0}$ represents the amount of commodity of type j departing location ℓ on a connector of type h at time period t to travel to location k. We then define the following auxiliary variables that keep track of the available connectors and commodities at each location $\ell \in \mathcal{L}$ and each

time period $t \in [\![1, T]\!]$: $v_{h\ell t} \in \mathbb{Z}_{\geq 0}$ (resp. $w_{j\ell t} \in \mathbb{R}_{\geq 0}$) represents the number of connectors of type $h \in \mathcal{H}$ (resp. commodities of type $j \in \mathcal{J}$) on ground at location ℓ and ready to depart at time period t. Finally, we define variables $s_{ij\ell t} \in \mathbb{R}_{\geq 0}$ (resp. $r_{ij\ell t} \in \mathbb{R}_{\geq 0}$) that represent the amount of commodities of type $j \in \mathcal{J}$ used to satisfy (resp. still needed to satisfy) demand from request $i \in \mathcal{I}$ at location $\ell \in \mathcal{L}$ at time period $t \in [\![1, T]\!]$. We can then formulate the following MIP for determining the optimal logistics plan:

$$\min_{x,y,v,w,s,r} \quad \sum_{t=1}^{T} \sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{J}} \sum_{\ell\in\mathcal{L}} P_{rj\ell t} \cdot r_{ij\ell t} \tag{4.1a}$$

$$\text{s.t.} \quad \sum_{j\in\mathcal{J}} Q_j \cdot y_{jh\ell kt} \leq A_h \cdot x_{h\ell kt}, \quad \forall t \in [\![1,T]\!],\ \forall h \in \mathcal{H},\ \forall \ell \in \mathcal{L},\ \forall k \in \mathcal{L}_{h\ell t}, \tag{4.1b}$$

$$\sum_{j\in\mathcal{J}} Q_j \cdot w_{j\ell t} \leq B_\ell, \quad \forall t \in [\![1,T]\!],\ \forall \ell \in \mathcal{L}, \tag{4.1c}$$

$$v_{h\ell t} \leq C_{h\ell}, \quad \forall t \in [\![1,T]\!],\ \forall h \in \mathcal{H},\ \forall \ell \in \mathcal{L}, \tag{4.1d}$$

$$v_{h\ell t} = v_{h\ell,t-1} + V_{h\ell t} - \sum_{k\in\mathcal{L}_{h\ell t}} x_{h\ell kt} + \sum_{\{(k,t')\in\mathcal{L}\times[\![1,T]\!]\,|\,t'+N_{hk\ell t'}+M_{h\ell}=t,\ \ell\in\mathcal{L}_{hkt'}\}} x_{hk\ell t'}, \quad \forall t \in [\![1,T]\!],\ \forall h \in \mathcal{H},\ \forall \ell \in \mathcal{L}, \tag{4.1e}$$

$$w_{j\ell t} = w_{j\ell,t-1} + W_{j\ell t} - \sum_{i\in\mathcal{I}} s_{ij\ell t} - \sum_{h\in\mathcal{H}} \sum_{k\in\mathcal{L}_{h\ell t}} y_{jh\ell kt} + \sum_{h\in\mathcal{H}} \sum_{\{(k,t')\in\mathcal{L}\times[\![1,T]\!]\,|\,t'+N_{hk\ell t'}=t,\ \ell\in\mathcal{L}_{hkt'}\}} y_{jhk\ell t'} \quad \forall t \in [\![1,T]\!],\ \forall j \in \mathcal{J},\ \forall \ell \in \mathcal{L}, \tag{4.1f}$$

$$r_{ij\ell t} = r_{ij\ell,t-1} + D_{ij\ell t} - s_{ij\ell t}, \quad \forall t \in [\![1,T]\!],\ \forall i \in \mathcal{I},\ \forall j \in \mathcal{J},\ \forall \ell \in \mathcal{L}, \tag{4.1g}$$

$$x_{h\ell kt}, v_{h\ell t} \in \mathbb{Z}_{\geq 0}, \quad \forall t \in [\![1,T]\!],\ \forall h \in \mathcal{H},\ \forall \ell \in \mathcal{L},\ \forall k \in \mathcal{L}_{h\ell t}, \tag{4.1h}$$

$$y_{jh\ell kt}, w_{j\ell t}, s_{ij\ell t}, r_{ij\ell t} \geq 0 \quad \forall t \in [\![1,T]\!],\ \forall j \in \mathcal{J},\ \forall h \in \mathcal{H},\ \forall \ell \in \mathcal{L},\ \forall k \in \mathcal{L}_{h\ell t} y_{jhk\ell t'} \tag{4.1i}$$

In this MIP, Objective (4.1a) minimizes the total penalty for delayed demand throughout the planning horizon. Constraints (4.1b) model payload capacities

of connectors, ensuring that enough connectors of each type travel to transport commodities. Constraints (4.1c) and (4.1d) ensure that stored commodities and stationed connectors do not exceed location capacities. Constraints (4.1e) update the number of connectors of each type available at each location and time period, accounting for the connectors leaving, arriving, and entering the system. Similarly, Constraints (4.1f) ensure that the variables w represent the amount of commodities of each type available at each location and time period, accounting for the commodities leaving, arriving, being supplied, and being used to satisfy requests. Constraints (4.1g) update the amount of commodities of each type that must still be satisfied for each request at each location and each time period. To ensure validity of the model, we set variable $v_{h\ell 0}$, $w_{j\ell 0}$, and $r_{ij\ell 0}$ to 0. The remaining constraints define the domain of the decision variables. In particular, the nonnegativity of Constraints (4.1h), combined with Constraints (4.1e), ensure that the number of connectors departing from a location at any time period does not exceed the number of available connectors. Similarly, Constraints (4.1i) and (4.1e) ensure that at any time period and any location, the amount of commodities that depart or are used to satisfy requests does not exceed the amount of available commodities. Finally, Constraints (4.1i) and (4.1g) ensure that the amount of commodities used to satisfy requests does not exceed the corresponding pending demand.

4.7 Case Study

4.7.1 Southern California Scenario

In this section, we test our optimization-based logistics planning tool on an expeditionary scenario designed by subject matter experts to closely reflect real-world conditions. The scenario, illustrated in Figure 4.2, is set in Southern California and has been meticulously calibrated to mirror Marine Corps Air-Ground Combat Center Twentynine Palms training scenarios used for training and certifying Marine Littoral Regiments.

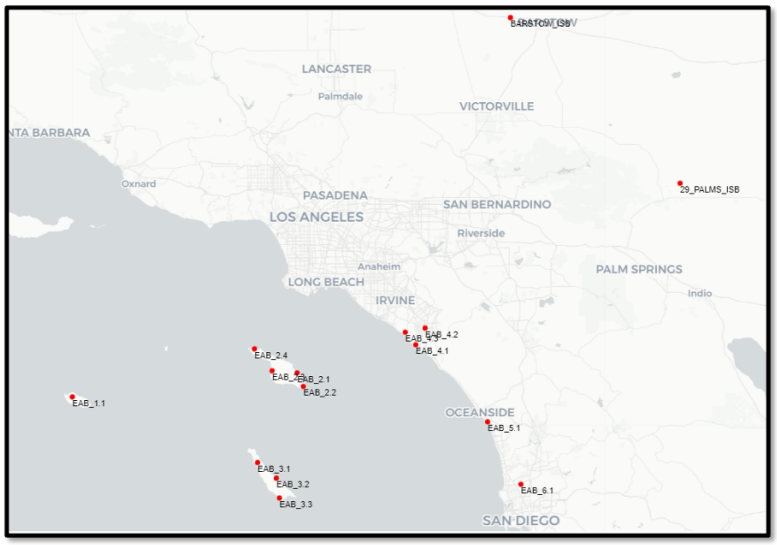


Figure 4.2: Southern California expeditionary scenario.

The logistics network consists of two supply locations in Barstow and Twentynine Palms, CA, and 13 expeditionary advanced bases (EABs) positioned along and off the Pacific coast. The supply locations initially hold 59 types of commodities, categorized into Class I (Meals Ready-to-Eat), Class II (equipment), Class III (fuel), Class V (ammunition), Class VII (vehicles), and PAX (personnel). A COA has been chosen by the commanding officer, resulting in 378 requests for

commodities to sustain operations at each EAB, referred to as EABO. Each request is associated with a set of demands for multiple commodities that must be delivered to a location by a certain due date. The planning horizon spans one week, divided into 28 periods of 6 hours each. To transport commodities from their storage location to the EABs, 46 connectors are available, as described in Table 4.1. Each connector faces capacity constraints regarding the total commodity volume and weight it can carry. In addition, each connector has limits on the amounts of personnel and fuel it can transport.

Table 4.1: Connector characteristics.

Connector Type	#	Range [mi]	Speed [mph]	PAX Cap.	Fuel Cap. [lb]	Total Volume Cap. [cu ft]	Total Weight Cap. [lb]
KC-130	3	4000	360	92	10,000	3,600	42,000
MV-22	12	380	315	24	2,000	1,025	6,000
MTVR	16	300	55	15	1,800	400	14,000
MTVR-Trailer	7	300	55	15	1,800	650	26,000
LAW	2	1000	16	75	60,000	35,000	1,700,000
LCU-1700	3	1200	13	350	6,000	15,900	340,000
Railway	1	122	10	0	0	500,000	10,000,000
Truck-5	2	93	39	0	0	3,500	35,000
Truck-6	2	103	43	0	0	3,500	35,000

4.7.2 Logistics Planning

To develop a logistics plan for sustaining EABOs, we solve our proposed MIP, which in this scenario comprises 688,795 variables and 182,065 constraints, using the Gurobi optimization solver. The solver provides a solution within a 10% optimality gap in 35 minutes.

Figure 4.3 presents the force closure rates for EABOs 4-6 in terms of PAX and Classes I, III, and V. The results indicate some delays in fulfilling requests for these EABOs. Specifically, the PAX demand for EABO 4 is only satisfied at 66%

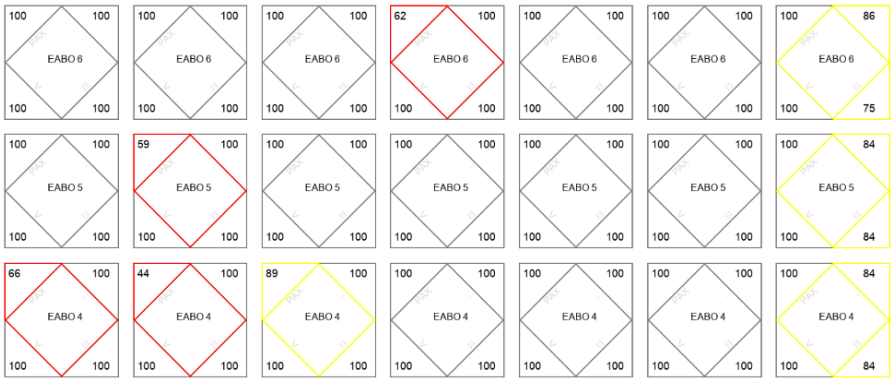


Figure 4.3: Force closure rates for three EABOs on each day of the planning horizon, with respect to four classes of supplies. Each square contains the percentage of demand for PAX (top left), Class I (top right), Class III (bottom right), and Class V (bottom left) satisfied at the corresponding EABO by that date.

by Day 1, 44% by Day 2, 89% by Day 3, and 100% by Day 4. Similar, though less severe, delays can be observed for EABOs 5 and 6. This is attributed to the large personnel request early in the planning horizon, while most connectors have small PAX capacities; although landing craft units (LCUs) have the highest PAX capacities, they suffer from very low speeds. Further requests for Classes I and III toward the end of the planning horizon cannot be fully met due to the connectors being utilized to transport commodities to the more remote EABOs 1-3.

Figure 4.4 depicts the utilization rates of the connectors throughout the planning horizon. Notably, KC-130s are constantly utilized and operate at maximum capacity. These connectors are vital due to their high range, speed, and capacity. Although MV-22s have lower range and capacity, they remain important connectors for rapidly transporting commodities and fulfilling earlier requests. The railway and trucks are used to transport commodities out of Barstow and Twentynine Palms, respectively, which explains their higher utilization at the beginning of the planning horizon. Interestingly, while MTVRs with trailers are used for ground transportation, the logistics plan does not allocate any use for the re-

Connector	03-14	03-15	03-16	03-17	03-18	03-19	03-20
KC-130	100	100	100	100	100	100	100
MV-22	98	100	100	100	100	98	85
MTVR	0	0	0	0	0	0	0
MTVR-Trailer	25	100	89	93	96	89	100
LAW	75	100	88	75	50	50	100
LCU-1700	58	83	100	92	75	92	75
Railway	100	50	25	75	0	50	50
Truck-5	100	100	100	100	100	100	75
Truck-6	100	100	100	75	50	50	50

Figure 4.4: Connector utilization rates (% of maximum capacities) per day.

maining MTVRs (which have smaller capacities). This result indicates a surplus of ground connectors that could be more effectively replaced by other connector types for the chosen COA.

Notably, KC-130s are constantly utilized and operate at maximum capacity. These connectors are vital due to their high range, speed, and capacity. Although MV-22s have lower range and capacity, they remain important connectors for rapidly transporting commodities and fulfilling earlier requests. Railways and trucks are used to transport commodities out of Barstow and Twentynine Palms, respectively, which explains their higher utilization at the beginning of the planning horizon. Interestingly, while MTVRs with trailers are used for ground transportation, the logistics plan does not allocate any use for the remaining MTVRs (which have smaller capacities). This result indicates a surplus of ground connectors that could be more effectively replaced by other connector types for the chosen COA.

Lastly, Figure 4.5 illustrates a portion of the logistics plan over the first 30 hours of the planning horizon. This snapshot highlights the complex synchronization of multiple connectors across the intermodal logistics network to transport as many commodities as possible during the first hours of the planning horizon to

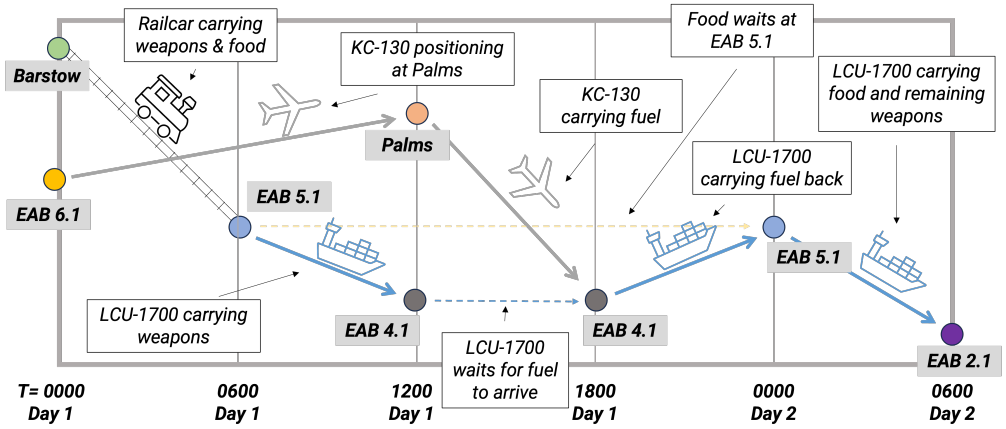


Figure 4.5: Partial logistics plan generated from the MIP.

the demand locations. To satisfy the demand for weapons at EAB 4.1, they are first transported from Barstow to EAB 5.1 by railway before being loaded on an LCU-1700 and transported to their final destination.

However, these connectors have remaining capacity, which they aim to utilize to transport additional commodities and satisfy the request at EAB 2.1. Specifically, the railcar carries additional weapons and food on its journey to EAB 5.1. Due to the maximum commodity capacity at EAB 5.1, the logistics plan only stores food at that location, while the remaining weapons are loaded onto the LCU-1700 and transported to EAB 4.1. Importantly, after the LCU-1700 has satisfied the weapons demand at EAB 4.1, it waits for 6 hours before traveling back to EAB 5.1 with the remaining weapons. This waiting period allows for fuel to arrive on a KC-130 from Twentynine Palms, destined for EAB 5.1. We note that the logistics plan does not send the KC-130 directly to EAB 5.1 with the fuel, as it is farther from Twentynine Palms compared to EAB 4.1. Since KC-130s are the most valuable connectors, the logistics plan has the KC-130 deliver fuel to a closer location, so it is readily available to travel to another location sooner, while the fuel completes its journey on the slower LCU-1700. The LCU-1700 then travels from EAB 4.1 to

EAB 5.1 with the fuel and remaining weapons, delivers the fuel at EAB 5.1, picks up the food that has been waiting at that location for 18 hours, and then travels to EAB 2.1 to satisfy its food and weapons demand.

Interestingly, we observe that the KC-130 was initially located at EAB 6.1 and needed to travel empty to reach the stocking location of commodities in Twenty-nine Springs. This demonstrates how repositioning connectors at the beginning of the planning horizon can increase the efficiency of the logistics plan and reduce fulfillment delays. Overall, this solution shows the intricate planning required to sustain operations logistically during expeditionary warfare, as it jointly depends on connector and location capacities, connector speeds and ranges, initial locations of commodities and connectors, and demand requests. This large number of factors is challenging to incorporate when creating logistics plans manually, as is currently done in practice, underscoring the value of the proposed optimization-based decision-support tool.

4.7.3 Contingency Planning

We conclude this case study by demonstrating the value of the proposed approach in generating contingency plans. To this end, we adjust the Southern California scenario to model a situation where one of the KC-130s is out of service during the planning horizon. Under this condition, we resolve the MIP and analyze the impact on the logistic plan. Figure 4.6 presents the effects on the force closure rates for EABOs 4-6 in terms of PAX, and Classes I, III, and V.

We observe that the loss of one KC-130 has a significant impact on the fulfillment of personnel demand at EABOs 4 and 5, resulting in an additional two-day

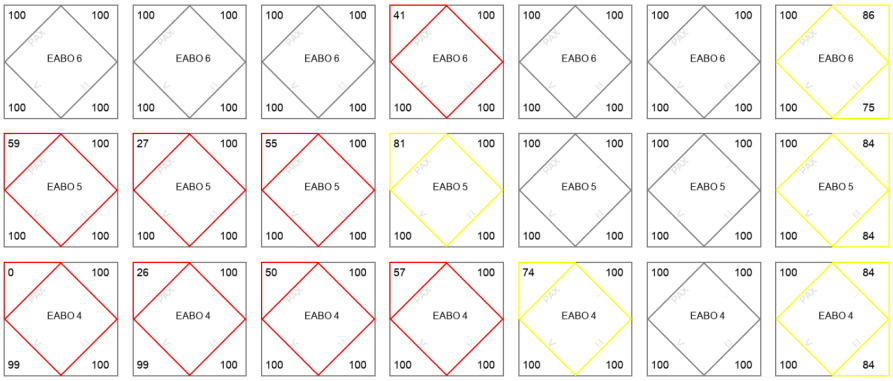


Figure 4.6: Force closure rates for three EABOs on each day of the planning horizon, with respect to four classes of supplies. Each square contains the percentage of demand for PAX (top left), Class I (top right), Class III (bottom right), and Class V (bottom left) satisfied at the corresponding EABO by that date in the contingency plan.

delay. This underscores the importance of KC-130s in transporting PAX during expeditionary warfare and highlights the logistics vulnerability in the event of unforeseen connector unavailability.

Finally, Figure 4.7 illustrates the impact on the utilization rates of the remaining connectors. The MV-22s and remaining KC-130s continue to operate at

Connector	03-14	03-15	03-16	03-17	03-18	03-19	03-20
KC-130	100	100	100	100	100	100	100
MV-22	100	100	100	100	100	98	88
MTVR	0	0	0	0	0	0	0
MTVR-Trailer	29	93	89	89	89	100	79
LAW	50	100	75	62	88	88	88
LCU-1700	67	83	92	100	83	83	100
Railway	75	100	100	75	25	50	25
Truck-5	100	100	100	100	100	100	75
Truck-6	100	75	75	100	50	25	100

Figure 4.7: Connector utilization rates (% of maximum capacities) per day in the contingency plan.

maximum capacity, and we similarly observe minor changes in the utilization of trucks. However, the railway is more heavily utilized as it compensates by trans-

porting more commodities out of Barstow that were originally carried by a KC-130. Interestingly, while MTVRs without trailers remain unused, there is a small decrease in the utilization of MTVRs with trailers. This decrease is primarily due to the reduced flow of commodities reaching the remote EABs, which subsequently decreases the need for ground transportation. Overall, this analysis demonstrates the capability of the proposed decision-support tool to rapidly analyze what-if scenarios, highlighting the vulnerability of an expeditionary logistics network and enabling the creation of contingency plans in case of unforeseen events.

4.8 Conclusion

In this article, we developed what we believe is the first optimization-based decision-support tool for expeditionary logistics planning. We formulated a mixed-integer program to optimize the routing and scheduling of various types of connectors and commodities within a complex intermodal logistics network, while accounting for critical features pertaining to connectors (speed, range, capacity, maintenance), locations (availability, capacity), and commodities (availability, due dates). We then tested our approach on an expeditionary scenario set in Southern California and designed to reflect the US Marine Corps training scenarios. Our results highlighted the model's ability to efficiently generate logistics plans that carefully synchronize connectors, ensuring timely fulfillment of most operational demands. We also showed how our model can be leveraged to conduct what-if scenarios, permitting logistics planners to swiftly create contingency plans. Our results demonstrate the value of prescriptive analytics for expeditionary logistics planning, enhancing the current qualitative and manual analyses conducted by military logistics planners. Overall, our decision-support tool can be utilized to

design complex and more efficient logistics plans to sustain expeditionary forces and improve the capabilities of maneuver course of actions.

Future research directions involve extending our model by incorporating additional factors, such as stochastic elements to account for uncertainty in logistics operations, and integrating more advanced solution techniques to handle even larger and more complex scenarios. Additionally, accounting for risks of disruptions in logistics plans due to enemy attacks would permit designing more robust solutions, ensuring the success of expeditionary operations.

4.9 Lessons Learned

The work described in this chapter was funded by the Office of Naval Research over the course of three years. Although our contribution is technical at its core, there were also many other challenges due to the applied nature of this project. In particular, the project deliverable is a software solution for use by Marine and Navy logisticians to replace their current workflow. This required frequent communication with Marine and Navy stakeholders, as well as robust, production-quality code beyond the typical requirements of academic research code. In this section, I describe some lessons I have learned in my time on this project that may be helpful for any readers that also find themselves in similar applied work.

4.9.1 Software Engineering

In the first few months of the project, the project code consisted almost entirely of Python notebooks. This was a workable solution for a brief period, as we were

still working out the MIP constraints and testing out different Python packages. However, once the scope of the project was made clear, it quickly became obvious that Python notebooks would no longer suffice and that real software engineering was necessary.

To that end, we strove towards the standard of production-ready code, to the level of what other members of the project have seen in industry. This required thinking about things that I rarely thought about in my academic research, such as how parameters should be ingested, what the optimization pipeline looks like, how the solution of the MIP should be displayed to the user, etc.

Moreover, the discipline to test code was also something that I came to appreciate. Initially, writing tests and validation checks felt like a chore. After all, in academic settings, it is often sufficient for research code to run on a known set of inputs, e.g. numerical experiments on data that we curated ourselves. However, when deploying code to production, there is no knowing what kinds of nonsensical inputs the user may try running the code with. In cases like this, it is very important for code to be able to fail gracefully, providing important information on what aspects of the inputs were wrong so that the user does not complain that the entire product is broken, but rather understands it is only a problem with the input they provided.

4.9.2 Technical Debt

A more broad concept tied to the previous one is technical debt. A common phrase in the technology industry (and indeed I have learned it as an Uber intern as well), technical debt is the cost incurred by rapidly and/or lazily writing "something that

works" when first developing a solution, as opposed to more carefully implementing code that prevents future code refactoring but may require more time. Taking the financial metaphor even further, interest is also accrued, in the sense that using the rapidly developed code as a foundation requires suboptimal design decisions further down the line, compounding the inefficiency and adding unnecessary complexity. Finally, the debt is "paid back" when this first iteration of the code usually has to be scrapped and rewritten due to poorly structured and inefficient code.

Of course, technical debt is hard to avoid when pressing deadlines exist, as was the case in this project as well as in industry. For instance, as I mentioned in the previous section, Python notebooks were initially useful for testing changes and having quick feedback cycles. However, when deadlines are not as urgent and you can spend a a day today to design and write good code to prevent a week of headache in a few months, I think the choice is clear.

4.9.3 Stakeholder Engagement

When developing a new tool for a group of people to use, it is crucial to maintain an open channel for communication with stakeholders. Whether we like it or not, not everything is set in stone at the beginning of a three-year project. This was one aspect of this project that I felt was particularly difficult to navigate, as the set of stakeholders and their understanding of use cases changed with time. In addition, aspects of this project were sufficiently sensitive as to require government security clearance (which most of the academic team did not have), making direct communication with uniformed personnel more difficult.

To that end, it was very important for us to always be looped in to the conver-

sation between our program officers and the stakeholders. This allowed to rapidly incorporate feedback as soon as possible and to make sure our work was aligned with expectations. It is always a bad feeling to be working hard on some feature that ends up being deemed useless by the person paying for your product.

CHAPTER 5

EFFICIENT DUAL DECOMPOSITION FOR VEHICLE ROUTING PROBLEMS WITH DELIVERY

This chapter is based on joint work with Peter Frazier, Matthew Ford, and Huseyin Topaloglu.

5.1 Introduction

The transportation of heterogeneous commodities across complex intermodal networks is an important problem in both industry and military applications. In such settings, it is crucial to coordinate the timely delivery of commodities across a multitude of road, air, and sea routes. The previous chapter discussed an example use case for the United States Marine Corps and Navy, which involved the routing and scheduling of commodities like fuel and food for expeditionary warfare. Beyond military applications, such logistics problems also arise in daily operations of commercial shipping companies like FedEx and UPS, which must deliver a multitude of commodities across air and ground routes to customers.

In the previous chapter, we presented a MIP formulation for this problem, which relies on tracking the movements of vehicles and commodities through time and space. However, the size of the resulting formulation, in terms of the number of variables and constraints, grows as the cardinality of the cross product of the set of vehicles $\mathcal{H}$, time steps $[\![1, T]\!]$, the set of commodities $\mathcal{J}$, and the set of arcs in the network, which is generally proportional to $\mathcal{L} \times \mathcal{L}$. In other words, for large networks with many commodity types, vehicle types, locations, and a long time horizon, the time and memory requirements for solving the MIP directly become

impractical.

To that end, we present an efficient algorithm for solving these problems that leverages Lagrangian duality and dual decomposition [Everett III, 1963]. We decompose the MIP into many smaller subproblems, each of which is efficiently solvable. By iteratively solving these subproblems, we aim to derive an approximately optimal solution to the original MIP with a smaller memory footprint and in less time. In this way, we permit our MIP model to be more widely applicable across large-scale problems, avoiding the need for large compute.

5.2 Related Work

5.2.1 Military Airlift

The logistics MIP formulation that we solve is closely related to the LP formulations used in military airlift [Morton et al., 1996, Rosenthal et al., 1997, Killingsworth and Melody, 1997, Baker et al., 2002] as well as the MIP formulations used in optimizing the Navy's Combat Logistics Force (CLF) [Powell, 2004, Brown and Carlyle, 2008, Brown et al., 2017, Brown and Kline, 2021]. See Chapter 4 for a more detailed discussion.

5.2.2 Lagrangian Relaxation for Transportation

There exists a large literature within operations research that use Lagrangian relaxation in various ways to solve transportation problems more efficiently. In the

context of the airline industry, Topaloglu [2009] relax constraints linking flight legs to enable solving a network revenue management problem over individual flight legs, producing tighter upper bounds on revenue and higher expected revenues than competing LP approaches. Also for airlines, Aykin [1994] propose a solution method based on relaxing capacity constraints to solve the hub-and-spoke network design problem. Carlyle et al. [2008] use Lagrangian relaxation on path-weight constraints to solve constrained shortest path problems. For the multi-facility production, inventory, and distribution problem, Karakitsiou and Migdalas [2008] relax inventory balance constraints to enable solving the joint problem separately across production facilities and distributors. For the problem of routing container ships, Rana and Vickson [1991] relax a coupling constraint across container ships in order to decompose the Lagrangian across individual ships, where the joint problem is too large to solve directly. For the capacitated facility location problem, Klincewicz and Luss [1986] relax the capacity constraints to yield the easily solvable uncapacitated facility location as a subproblem. Plotkin et al. [1995] relax the capacity constraints in fractional packing problems (a geneneralization of multicommodity flow) to derive fast approximation algorithms.

The vehicle routing problem (VRP) literature is also ripe with Lagrangian relaxation methods. For VRP with time windows, Kohl and Madsen [1997] relax the constraint that all customers are serviced to enable solving the joint problem separately across each vehicle. Similarly, for the pickup and delivery problem with time windows, Mahmoudi and Zhou [2016] relax passenger pickup constraints to decompose the multi-vehicle problem across individual vehicles. For the multi-depot location routing problem, Özyurt and Aksen [2007] relax flow conservations constraints to decompose the joint problem into a facility-location problem and a minimum spanning forest problem.

The paper closest to ours in approach is Crainic et al. [2001], which considers multicommodity capacitated fixed-charge network design. Multicommodity capacitated fixed-charge network design denotes the problem of sending flows of commodities across a network with capacitated arcs, additionally with a fixed cost per arc used. Based on the relaxations introduced in Gendron and Crainic [1994] and Gendron et al. [1999], the authors consider two sets of constraints to relax: the capacity constraints and flow balance constraints. In the case that flow balance constraints are relaxed, the Lagrangian relaxation yields subproblems across arcs that are continuous knapsacks, resembling the continuous knapsacks that arise in our vehicle subproblem. As in our problem, they also find that the optimal value of the Lagrangian matches the optimal value of the LP relaxation. Finally, they also use the bundle method to solve the Lagrangian, finding that it outperforms the subgradient method, as we do. However, the main contribution of this paper is an empirical analysis of the quality of the lower bound produced by solving the Lagrangian dual across a large test suite of problems, depending on the choice of constraints to relax and the choice of dual optimization method. In contrast, we focus on leveraging the Lagrangian relaxation to produce feasible solutions to the original MIP, not just lower bounds. Additionally, the problems considered by this line of work and our problem are different; for instance, the networks considered in these problems are static (compared to time-expanded, as is the case for us) and so their continuous knapsacks are not solved as subroutines within a dynamic program.

5.2.3 Dual Decomposition

Dual decomposition [Everett III, 1963] is a popular decentralized method for solving large-scale optimization problems based on Lagrangian relaxation. There is a large literature on using dual decomposition to efficiently solve various optimization problems, such as distributed control [Rantzer, 2009], wireless network routing Xiao et al. [2004], and multi-agent optimization [Terelius et al., 2011]. There also exist works exploits the structure of particular problems so that these subproblems can be solved efficiently, e.g. faster than solving them as linear programs (LPs) or MIPs [Torresani et al., 2012, Agrawal et al., 2022]. We view our contribution as similarly exploiting the problem structure of the MIP formulation for our logistics problem to find efficient solution efficiently.

5.3 Generic MIP Formulation

We first present a simplified version of the MIP from the previous chapter. In particular, we make the following simplifications:

1. the commodity variables x are continuous,
2. there is only one capacity constraint,
3. the penalty P only depends on j, ℓ (not on t),
4. the set of request IDs $\mathcal{I}$ and its indices i are removed.

The first three simplifications accelerate our approach for optimizing the dual, as we will explain in 5.4.1 and 5.4.2. In particular, (1) and (2) allow us to quickly solve a continuous knapsack problem as a subroutine, while (3) allows us to solve

a different subproblem analytically. On the other hand, (4) is only a simplification for convenience, and can be easily adapted for this problem by defining a different commodity type per request.

As the MIP has many constraints, we present it in multiple parts. First, we present the objective:

$$\min_{x,y,v,w,s,r} \sum_{t=1}^{T} \sum_{j \in \mathcal{J}} \sum_{\ell \in \mathcal{L}} P_{j\ell} \cdot (r_{j\ell t} - s_{j\ell t}), \tag{5.1}$$

where $x_{h,\ell,k,t}$ is the number of vehicles of type h traveling arc (ℓ, k) starting at time t, $y_{jh\ell kt}$ is the number of commodities of type j traveling on vehicles of type h on arc (ℓ, k) starting at time t, $v_{h\ell t}$ is the number of vehicles of type h at location ℓ at time t, $w_{h\ell t}$ is the number of commodities of type j at location ℓ at time t, $s_{j\ell t}$ is the amount of demand satisfied at location ℓ at time t for commodity type j, and $r_{j\ell t}$ is amount of demand, including previously unsatisfied demand, to be satisfied at location ℓ at time t, for commodity type j.

We now present constraints which involve only the x, y, v variables:

$$\sum_{j \in \mathcal{J}} Q_j \cdot y_{jh\ell kt} \leq A_h \cdot x_{h\ell kt} \qquad \forall t \in [\![0, T]\!],\ \forall h \in \mathcal{H},\ \forall \ell \in \mathcal{L},\ \forall k \in \mathcal{L}_{h\ell t},$$

$$v_{h\ell t+1} = v_{h\ell,t} + V_{h\ell t} - \sum_{k \in \mathcal{L}_{h\ell t}} x_{h\ell kt} + \sum_{\{(k,t') \in \mathcal{L} \times [\![1,T]\!] \,|\, t' + N_{hk\ell} = t,\ \ell \in \mathcal{L}_{hkt'}\}} x_{hk\ell t'}$$
$$\forall t \in [\![0, T-1]\!],\ \forall h \in \mathcal{H},\ \forall \ell \in \mathcal{L}$$

$$0 \leq v_{h\ell T} + V_{h\ell T} - \sum_{k \in \mathcal{L}_{h\ell T}} x_{h\ell kT} + \sum_{\{(k,t') \in \mathcal{L} \times [\![1,T]\!] \,|\, t' + N_{hk\ell} = T,\ \ell \in \mathcal{L}_{hkt'}\}} x_{hk\ell t'},$$
$$\forall h \in \mathcal{H},\ \forall \ell \in \mathcal{L}$$

$$v_{h\ell 0} = 0 \qquad \forall h \in \mathcal{H},\ \forall \ell \in \mathcal{L}$$

$$x_{h\ell kt}, v_{h\ell t} \in \mathbb{Z}_{\geq 0} \qquad \forall t \in [\![0, T]\!],\ \forall h \in \mathcal{H},\ \forall \ell \in \mathcal{L},\ \forall k \in \mathcal{L}_{h\ell t}$$

$$y_{jh\ell kt} \in \mathbb{R}_{\geq 0} \qquad \forall t \in [\![0, T]\!],\ \forall j \in \mathcal{J},\ \forall h \in \mathcal{H},\ \forall \ell \in \mathcal{L},\ \forall k \in \mathcal{L}_{h\ell t}$$

Note that, in comparison to the MIP from Chapter 4, we also slightly change the way time is indexed. In particular, at a time period t, $v_{h\ell t}, w_{j,\ell t}$ now measures the number of vehicles/commodities remaining at a location ℓ after vehicles/commodities arrived at time t and also after vehicles/commodities departed at time t. A consequence is that we have an additional constraint which ensures that after the last time period T, there is still a positive number of vehicles/commodities at each location.

We also have the following constraints over only the w, r, s variables:

$$w_{j\ell 0} = 0 \qquad \forall j \in \mathcal{J}, \forall \ell \in \mathcal{L}$$

$$r_{j\ell t} = r_{j\ell t-1} + D_{j\ell t} + s_{j\ell t-1} \qquad \forall t \in [\![1, T]\!],\ \forall j \in \mathcal{J}, \forall l \in \mathcal{L}$$

$$s_{j\ell t} \leq r_{j\ell t} \qquad \forall t \in [\![0, T]\!],\ \forall j \in \mathcal{J}, \forall l \in \mathcal{L}$$

$$w_{j\ell t}, s_{j\ell t}, r_{j\ell t} \in \mathbb{R}_{\geq 0} \qquad \forall t \in [\![0, T]\!],\ \forall j \in \mathcal{J}, \forall l \in \mathcal{L}$$

Also observe the objective only involves these variables.

Finally, we have the following constraints which involve both sets of variables:

$$w_{j\ell t+1} = w_{j\ell,t} + W_{j\ell t} - s_{j\ell t} - \sum_{h \in \mathcal{H}} \sum_{k \in \mathcal{L}_{h\ell t}} y_{jh\ell kt} + \sum_{h \in \mathcal{H}} \sum_{\{(k,t') \in \mathcal{L} \times [\![1,T]\!] \,|\, t' + N_{hk\ell} = t,\ \ell \in \mathcal{L}_{hkt'}\}} y_{jhk\ell t'}$$
$$\forall t \in [\![0, T-1]\!], \forall \ell \in \mathcal{L}, \forall j \in \mathcal{J} \qquad (5.2)$$

$$w_{j\ell,T} + W_{j\ell T} - s_{j\ell T} - \sum_{h \in \mathcal{H}} \sum_{k \in \mathcal{L}_{h\ell T}} y_{jh\ell kT} + \sum_{h \in \mathcal{H}} \sum_{\{(k,t') \in \mathcal{L} \times [\![1,T]\!] \,|\, t' + N_{hk\ell} = T,\ \ell \in \mathcal{L}_{hkt'}\}} y_{jhk\ell t'} \geq 0$$
$$\forall \ell \in \mathcal{L}, \forall j \in \mathcal{J} \qquad (5.3)$$

Again, the second constraint above is a consequence of re-indexing the time periods compared to Chapter 4, ensuring that a positive number of commodities are leftover at each location, after the last time period.

In this formulation, the number of variables scale linearly with the number of vehicles and time steps and quadratically with the number of locations. When the problem is small, the problem can be solved with standard off-the-shelf MIP software. However, in real scenarios, planning horizons can be 30 days, the set of locations may span an entire ocean, and there may be hundreds of different vehicles and commodities to keep track of. In such problems, there are millions of decision

variables and constraints, and it becomes infeasible to solve such problems with MIP software under reasonable memory and time constraints.

5.4 Decomposition

We now introduce the decomposition. We partition the variables into two sets: (1) "vehicle variables", x, y, v, which are variables tracking vehicles and inventory moving on them; and (2) "commodity-location variables", w, r and s, which are variables related to demand and inventory at nodes. This will allow us to define separate problems that can be solved in parallel. To enable this, we dualize constraints 5.2 and 5.3 (our commodity balance constraints) . This gives us the following additional penalty terms in the objective:

$$\sum_{j\ell t} \lambda_{j\ell t} \left(w_{j\ell,t+1} - w_{j\ell t} + \sum_{k,h} y_{jh\ell kt} - \sum_{k,h} y_{jhk\ell,t-N_{hk\ell}} + s_{j\ell t} - W_{j\ell t} \right)$$

for $t = 1, \ldots, T-1$ and

$$\sum_{j\ell} \lambda_{j\ell T} \left(-w_{j\ell T} + \sum_{\ell',h} y_{jh\ell kT} - \sum_{\ell',h} y_{jhk\ell,T-N_{hk\ell}} + s_{j\ell T} - W_{j\ell T} \right)$$

Here, each $\lambda_{j\ell t} \in \mathbb{R}$ for $t < T$ and $\lambda_{j\ell t} \in \mathbb{R}_{\geq 0}$ is a Lagrange multiplier that penalizes violations of the flow balance constraint at node j in time t for commodity ℓ. Positive values for $\lambda_{j\ell t}$ penalize having the flow out be larger than the flow in.

In the decomposed form, each constraint only involves a single set of variables, either the vehicle variables or the commodity and location variables, and the objective function is linear. Moreover, each constraint involving vehicle variables involves just one vehicle. This lets us further divide the vehicle problem into one problem per vehicle. Similarly, each constraint involving commodity and loca-

tion variables involves just one commodity-location, allowing us to divide into one problem per commodity-location.

Note that the commodity flow balance constraint is not the only choice of constraint to dualize. In particular, many methods in the transportation literature choose to dualize capacity constraints in the case that the uncapacitated subproblems are easily solvable [Crainic et al., 2001, Aykin, 1994, Klincewicz and Luss, 1986]. For our problem, in the case that the capacity constraints are dualized (i.e. $\sum_{j\in\mathcal{J}} Q_j y_{jh\ell kt} \leq A_h x_{h\ell kt}$), there is a different decomposition across the variables, where now we can separate across the x, v variables and the w, y, r, s variables. The objective would then include a term $\nu_{xh\ell t}\left(A_h x_{h\ell kt} - \sum_{j\in\mathcal{J}} Q_j y_{jh\ell kt}\right)$, where ν are nonnegative dual variables. One could then solve the decomposed subproblems as MIPs in a dual ascent or bundle method approach as well. However, it is not obvious that the decomposed subproblems admit efficient solutions as in the case of dualizing the commodity balance constraints, as we discuss below. A future research direction is to explore efficient solution methods for this alternative choice of Lagrangian.

5.4.1 Vehicle Subproblem

For each individual vehicle h, we have the following subproblem. For notational simplify, we drop the h index for the variables.

$$\min_{x,y,v} \quad \sum_{t=1}^{T} \sum_{j \in \mathcal{J}} \sum_{\ell \in \mathcal{L}} \lambda_{j\ell t} \left(\sum_{k} y_{j\ell k t} - \sum_{k} y_{jk\ell, t-N_{hk\ell}} \right) \tag{5.4a}$$

$$\text{s.t.} \quad \sum_{j \in \mathcal{J}} Q_j \cdot y_{j\ell k t} \leq A_h \cdot x_{\ell k t}, \quad \forall t \in [\![1, T]\!],\ \forall \ell \in \mathcal{L},\ \forall k \in \mathcal{L}_{h\ell t}, \tag{5.4b}$$

$$v_{\ell t} = v_{\ell, t-1} + V_{\ell t} - \sum_{k \in \mathcal{L}_{h\ell t}} x_{\ell k t} + \sum_{\{(k,t') \in \mathcal{L} \times [\![1,T]\!] \,|\, t' + N_{hk\ell} = t,\ \ell \in \mathcal{L}_{hkt'}\}} x_{k\ell t'},$$

$$\forall t \in [\![1, T-1]\!],\ \forall \ell \in \mathcal{L}, \tag{5.4c}$$

$$v_{\ell T} + V_{\ell T} - \sum_{k \in \mathcal{L}_{h\ell T}} x_{\ell k T} + \sum_{\{(k,t') \in \mathcal{L} \times [\![1,T]\!] \,|\, t' + N_{hk\ell} = T,\ \ell \in \mathcal{L}_{hkt'}\}} x_{k\ell t'} \geq 0,$$

$$\forall \ell \in \mathcal{L}, \tag{5.4d}$$

$$v_{\ell 0} = 0 \tag{5.4e}$$

$$x_{\ell k t}, v_{\ell t} \in \mathbb{Z}_{\geq 0}, \qquad \forall t \in [\![1, T]\!],\ \forall \ell \in \mathcal{L},\ \forall k \in \mathcal{L}_{h\ell t}, \tag{5.4f}$$

$$y_{j\ell k t} \in \mathbb{R}_{\geq 0} \qquad \forall t \in [\![1, T]\!],\ \forall j \in \mathcal{J},\ \forall \ell \in \mathcal{L},\ \forall k \in \mathcal{L}_{h\ell t}, \tag{5.4g}$$

This integer program optimizes the trajectory of a single vehicle across our network as it carries inventory from node to node. In this integer program, the objective corresponds to a problem where the vehicle can buy and sell inventory of commodity j at each node ℓ at time t at a price of $\lambda_{j\ell t}$ without facing constraints on the amount of inventory available at that node. Its goal is to chart a trajectory and a plan for buying and selling to maximize the total profit.

This integer program is not a particularly efficient way to solve this problem for problems with many nodes in light of the fact that we need to have flow variables for each arc and commodity. The problem can be solved more efficiently using dynamic programming, where the state of the dynamic program is the time and vehicle's location.

Dynamic Programming Approach

Consider solving the vehicle MIP (5.4) as a dynamic program, where we follow the movement of a single vehicle h over time. The path of a vehicle through the network is determined by the binary valued $x_{\ell kt}$ variables, and the contribution to the objective is determined by the prices $\lambda_{j\ell t}$ and the commodities $y_{j\ell kt}$ brought along these arcs. We consider deriving the optimal solution for each vehicle h by choosing arcs x and commodities to carry y, forward in time from the vehicle's starting location. In this way, the integrality of $x_{\ell kt}$ and $v_{\ell t}$ are naturally preserved, as are the balance constraints 5.4c and 5.4d.

Formally, suppose we have an agent which represents a single vehicle h as it traverses the network buying and selling commodities. The agent's state is $s := (\ell, t)$, the current location and time. At state $s = (\ell, t)$, the agent can take actions $a := (\{y_j\}_j, k)$, which represents the amount of each commodity j that the vehicle brings from ℓ to a next location k, starting at time t. Note that $(\{y_j\}_j, k)$ corresponds to $y_{j\ell kt}$ in the MIP. The action $(\{y_j\}_j, k)$ incurs **costs** $\lambda_{j,t,\ell} y_j$ from "buying" commodity j at price-per-unit $\lambda_{j,t,\ell}$ and transporting it, as well as **rewards** $\lambda_{j,t+N_{h\ell k},k}$ from "selling" commodity j at price-per-unit $\lambda_{j,t+N_{h\ell k},k}$ at the destination. Therefore we may write the reward function $r(s, a)$ as

$$r\left((\ell, t), (\{y_j\}_j, k)\right) = \sum_j (\lambda_{j,t+N_{h\ell k},k} - \lambda_{j,t,\ell}) y_j.$$

Now we can construct a dynamic program (DP) based on the following value function $V(\cdot)$.

$$V(\ell, t) = \max_{\{x_j\}_j \in \mathcal{X}, k} r\left((\ell, t), (\{x_j\}_j, k)\right) + V\left(k, t + N_{h\ell k}\right),$$

where $\mathcal{X}$ encodes the unit-count, volume, and weight capacity constraints of the MIP. Then we see that $V(\ell_{start}, t_{start})$ for vehicle h's start location-timetime ℓ_{start}, t_{start} corresponds to the optimal solution of the MIP.

To solve this dynamic program, we use backward recursion as shown in Algorithm 3, where each iteration reduces to a knapsack problem, for which standard solution methods exist (for continuous $\{x_j\}_j$). In particular, at one location-time ℓ, t, for each other location k, we solve the following continuous knapsack:

$$\begin{aligned} \max_{y} \quad & \sum_{j \in \mathcal{J}} y_{j\ell kt} \left(\lambda_{jkt+N_{h\ell k}} - \lambda_{j\ell t}\right) \\ \text{s.t.} \quad & \sum_{j \in \mathcal{J}} Q_j \cdot y_{j\ell kt} \leq A_h \\ & y_{j\ell kt} \in \mathbb{R}_{\geq 0} \end{aligned} \tag{5.5}$$

Here the value of each unit of commodity type j is $\lambda_{jkt+N_{h\ell k}} - \lambda_{j\ell t}$, so the well-known polynomial time solution to this knapsack is by taking the commodity with the largest (positive) ratio $\frac{\lambda_{jkt+N_{h\ell k}} - \lambda_{j\ell t}}{Q_j}$, and filling up to capacity [Dantzig, 1957]. This yields the value of $r\left((\ell, t), (\{x_j\}_j, k)\right)$, and we then calculate the maximum value of $V\left(k, t + \text{travel}_{h\ell k}\right) + r\left((\ell, t), (\{x_j\}_j, k)\right)$ over the choice of k.

Algorithm 2: Vehicle Problem Dynamic Program

```
for T = N to 1 do
    for L = 1 to L do
        Calculate V(L, T)
```

We may then solve this single-vehicle problem for each vehicle, and combine the solutions to yield an overall solution for the multi-vehicle problem.

5.4.2 Commodity-location Subproblem

For each commodity type j and location ℓ, we have the following subproblem (where $w_{j\ell,T+1}$ is defined to be 0). For notational simplicity, we simplify the notation for variables and only index over the remaining set, t:

$$\min_{w,s,r} \quad \sum_{t=1}^{T} P_{j\ell} \cdot (r_t - s_t) + \lambda_{j\ell t} \cdot (w_{t+1} - w_t + s_t) \tag{5.6a}$$

$$\text{s.t.} \quad w_0 = 0 \tag{5.6b}$$

$$r_t = r_{t-1} + D_t - s_{t-1}, \quad \forall t \in [\![1, T]\!], \tag{5.6c}$$

$$s_t \leq r_t, \quad \forall t \in [\![1, T]\!], \tag{5.6d}$$

$$w_t, s_t, r_t \in \mathbb{R}_{\geq 0} \quad \forall t \in [\![1, T]\!] \tag{5.6e}$$

In this problem, we have a node that can buy and sell inventory at a time-varying price $\lambda_{j\ell t}$, that pays a penalty for unmet demand, and that chooses how much of the unmet demand to meet in each period.

This problem can be solved via integer programming with $3T$ variables (s_t, r_t, w_t), where T is the number of time periods. Though this may solve quickly in practice, below we provide an even faster method for solving 5.6.

Analytic Solution

Observe that the penalty $P_{j\ell}$ does not depend on t, so that the penalty for missing a unit of demand is linear in the number of time steps by which the demand satisfaction is delayed. For the following, we remove the j, ℓ indices since there is a node problem for each j, ℓ.

We constrain the set of λ_t considered to only those for which the solution is bounded (below); we can ensure this by initializing our dual variables to a known, bounded solution and constraining our dual optimization. First, we argue that the prices λ_t should decrease monotonically with t. Observe the coefficient on w_t in the objective is $\lambda_{t-1} - \lambda_t$. Suppose there is a time period t such that $\lambda_{t-1} < \lambda_t$. Then we can send w_t to infinity, holding everything else the same, and achieve $-\infty$ objective. (This corresponds to buying an infinite amount at time $t-1$ and then selling it at time t.) Thus, we can constrain our search for dual variables to those that satisfy $\lambda_{t-1} \geq \lambda_t$ for each commodity and location.

We also claim that each λ_t must be non-negative. Suppose that there exists some t for which $\lambda_{t'} < 0$. Then for each $t > t'$, including the final time step T, we have $\lambda_t < 0$ by the previous paragraph. Then since the w_T term in the objective only has weight λ_T, if λ_T is negative, we can send w_T to ∞ to again yield $-\infty$ objective. Therefore we only consider prices λ_t which are non-negative.

Now consider a process which matches time periods t at which there are demand to time periods s when inventory is acquired to satisfy demand. This corresponds to setting the values of w_s, r_s, s_s, given demand signal D_t.

- If $t \geq s$ then the cost is λ_t. Because prices are declining over time, if $t \geq s$, it will be optimal to satisfy the demand as late as possible, which costs λ_t per unit (corresponding to increasing s_t by 1). Note that it would cost $\lambda_{t-1} - \lambda_t \geq 0$ to increment w_t, but we may also leave $w_t = 0$ and assume the demand is met by incoming commodities (from y variables). No late penalty is incurred.
- If $t < s$, then the cost is $(s-t) * P_{j\ell} + \lambda_s$; the demand is satisfied $s-t$ time steps late, so we incur the r_t cost for each step, and pay λ_s per unit for s_s

(again, leaving $w_s = 0$).

- If we never satisfy the demand, the cost is $(T - t + 1) * P_{j\ell}$ where T is the last period, and no cost for setting w, s.

From the above reasoning, the lowest-cost way to satisfy a unit of demand at time t is the minimum of these, i.e. $\min\{\lambda_{t-1}, \{P_{j\ell}(s-t) + \lambda_{s-1}\}_{s=t+1,\ldots,T}, P_{j\ell}(T-t)\}$. We can rewrite this as

$$\min_{t \le s \le T+1} P_{j\ell}(s-t) + \lambda_s,$$

where we define $\lambda_T := 0$ (note in our formulation the prices only go up to time $T-1$).

The minimizer does not depend on the individual unit of demand among D_t, so the total contribution to the objective would then be

$$\sum_t \mathrm{D}_t \cdot \left(\min_{t \le s \le T+1} P_{j\ell}(s-t) + \lambda_s \right).$$

5.5 Algorithm

To solve the joint problem over all variables, we propose the following algorithm.

1. Propose an initial set of dual variables $\lambda_{j\ell t}$.
2. Repeat the following until some termination criteria is reached (e.g. number of iterations, runtime, dual convergence)
   (a) Solve the dualized problem by solving, in parallel, the vehicle problem for each vehicle and the node problem for each node.

(b) Calculate the violation of the flow balance constraint. This gives us the subgradient of the dual problem. Use this to update the dual variables as in gradient ascent. This is a standard technique and is called *dual ascent*, or specifically *dual decomposition* where we decompose across the vehicle variables x, y, v and commodity-location variables w, s, r.

As a reminder for the reader, dual decomposition iterates between the two following steps, where we denote $f_i(x_i)$ the objective function for the vehicle problem and node problems for $i = \{x, y, v\}, \{w, s, r\}$ respectively, x_i represents the corresponding decision variables (in this notation, $x_i = i$), $\mathcal{X}_i$ is the corresponding feasible region, and $Ax = b$ is the flow balance constraint.

For $k = 1, 2, \ldots$

$$x_i^{(k)} \in \arg\min_{x_i \in \mathcal{X}_i} f_i(x_i) + \lambda^{(k-1)} A_i x_i, \quad i = \{x, y, v\}, \{w, s, r\}$$
$$\lambda^{(k)} = \lambda^{(k-1)} + t_k \left(\sum_i A_i x_i^{(k)} - b \right).$$

Much of the language we use relies on the price coordination interpretation of dual decomposition, where prices are updated based on whether resources are over- or under-utilized (i.e. the sign of the slack in the flow balance constraint). See Boyd et al. [2011] and Vandenberghe [2022] for reference.

3. Use the last solve of the vehicle and node problems to get a feasible solution. See section 5.6.

Warm Starting

In the event that an optimal dual solution λ^* and set of primal variables $x^*, y^*, v^*, w^*, s^*, r^*$ from a related problem are available, one can generally plug in λ^* as an initial value for the dual variable in the new problem if the set of commodities, locations, and time steps are the same. Moreover, the first iteration of dual optimization (solving for the primal variables) can also be further accelerated in certain cases by leveraging the existing primal variables.

In particular, if for a certain commodity j and location ℓ, the only change is in the supply value $W_{j\ell t}$, then w^*, s^*, r^* will remain optimal in the new problem with dual variables λ^* because the objective for the commodity-location problem only changes by a constant (the difference in the supply).

If all commodity weights Q_j and, for a certain vehicle h, the weight capacity A_h and supply $V_{\ell t}$ remain the same, then the same x^*, y^*, v^* will remain optimal for the subproblem associated that vehicle h for dual variables λ^*, since all parameters for that vehicle subproblem will remain the same.

5.5.1 Accelerating Convergence with the Bundle Method

In step (2) above, we mention dual ascent as the algorithm for optimizing (maximizing) the dual. However, in our experiments, we found using the bundle method of Hiriart-Urruty and Lemaréchal [1996] accelerates the dual optimization.

The bundle method works by bounding the dual function $f(\lambda)$ above by an approximation $\hat{f}(\lambda)$ constructed from a "bundle" $\mathcal{B}$ of function value, subgradient, and dual variable tuples $(f(\lambda^k), g^k, \lambda^k)$. The approximation satisfies $f(\lambda) \leq \hat{f}(\lambda)$

and $f(\lambda^k) = \hat{f}(\lambda^k)$, and is formally defined as

$$\hat{f}(\lambda) = \min_{(f(\lambda'),\lambda',g') \in \mathcal{B}} f(\lambda') + \langle g', \lambda - \lambda' \rangle.$$

This approximation is used to iteratively generate the next set of dual variables λ^{k+1} by solving a quadratic program:

$$\lambda^{k+1} = \operatorname*{argmax}_{\lambda} \hat{f}(\lambda) - \frac{w}{2}\|\lambda - \overline{\lambda}\|_2^2, \tag{5.7}$$

where $\overline{\lambda}$ is the current iterate. Here $w > 0$ is a constant which controls how far away the next price can be.

Solving 5.7 can be done efficiently by a slight reformulation:

$$\begin{aligned} &\operatorname*{argmax}_{\lambda, v} v - \frac{w}{2}\|\lambda - \overline{\lambda}\|_2^2 \\ &\text{s.t.} f(\lambda') + \langle g', \lambda - \lambda' \rangle \geq v \quad \forall (f(\lambda'), \lambda', g') \in \mathcal{B} \end{aligned} \tag{5.8}$$

The solution to 5.8, λ^{k+1}, is then plugged into f to see if there is a sufficient increase relative to $f(\overline{\lambda})$. If so, we set $\overline{\lambda} = \lambda^{k=1}$, and otherwise do not update $\overline{\lambda}$. In any case, we append $f(\lambda^{k+1}, g^{k+1}, \lambda^{k+1})$ to $\mathcal{B}$ and iterate until convergence.

This is summarized in Algorithm 3. In practice, we initialize $\lambda_0 = \vec{0}$ and $w = 1$; we found the results were not particularly sensitive to these choices. Additionally, as discussed in Section 5.4.2, we added constraints to ensure the commodity-location problem had a bounded solution; we also found these to improve performance.

Algorithm 3: Bundle Method for Dual Optimization

Initialize λ_0, set $\overline{\lambda} = \lambda_0$.
$\mathcal{B} = \{f(\lambda_0), g_0, \lambda_0\}$
for $k = 0, 1, \ldots$ **do**
    Solve Equation (5.8) for λ^{k+1}
    if $f(\lambda) > f(\overline{\lambda})$ **then**
        $\overline{\lambda} = \lambda^{k+1}$
    end
    $\mathcal{B} \leftarrow \mathcal{B} \cup \{f(\lambda^{k+1}, g^{k+1}, \lambda^{k+1}\}$
end

5.5.2 Theoretical Properties

We now discuss theoretical properties of the dual solution.

Theorem 5.5.1. *The objective value of the optimal dual solution to 5.1 is equal to the primal optimal objective of the LP relaxation of 5.1.*

In other words, the duality gap of 5.1, a mixed integer program, is simply the integrality gap between 5.1 and its LP relaxation.

Proof. First, we observe that if we relax the v, x variables to be continuous, then the MIP becomes an LP. Moreover, observe we can easily construct the following feasible primal solution. We set $x_{h\ell kt} = y_{jh\ell kt} = s_{j\ell t} = 0$ for all indices, corresponding to a solution which does not move any vehicles or commodities. In this case, we incur the finite cost $\sum_{j\ell t} P_{j\ell}(r_{j\ell t} - s_{j\ell t})$ associated with not satisfying any of the demand over the entire time horizon. Moreover, the LP is clearly bounded below by 0, since we have a constraint $s_{j\ell t} \leq r_{j\ell t}$ and the objective is minimizing

a sum of positively weighted $r_{j\ell t} - s_{j\ell t}$ terms. Therefore the primal is feasible and bounded, and by standard LP theory [Williamson, 2014], we know strong duality holds in that the optimal dual and primal objectives match.

Now consider applying the same decomposition to the corresponding LP. Note we have the same vehicle and commodity-location subproblems, except that in the vehicle subproblem, x and v variables lose the integer constraints.

Let λ^* be the optimal dual variables for the LP relaxation, and consider the corresponding optimal x^*, v^* for one particular vehicle h in the inner minimization problem.

We make the following observation. Instead of viewing the movement of the vehicle h through the x variables which index over ℓ, k, t, consider each entire path p through the network (where the balance constraints for v are implicitly satisfied). The flow of the continuous-valued x can be expressed as a convex combination of binary-valued x_p, i.e. $\sum_p \alpha_p x_p$ where $\alpha_p \geq 0, \sum_p \alpha_p = 1$ and x_p corresponds to the movement of the binary-valued vehicle h.

In particular, the LP-relaxed vehicle subproblem is equivalent to the following path-based formulation. Let $\mathcal{P}_h$ denote the set of all (combinatorially many) paths from the start location of vehicle h to each other location, which terminate by the time horizon T. Denote $P_{h\ell kt}$ as the set of paths p of vehicle h which travel the arc (ℓ, k) starting at time t. Then, consider the following LP

$$\min_{x,\alpha} \quad \sum_{t=1}^{T} \sum_{j\in\mathcal{J}} \sum_{\ell\in\mathcal{L}} \lambda_{j\ell t} \left(\sum_{k} y_{jh\ell kt} - \sum_{k} y_{jhk\ell,t-N_{hk\ell}} \right) \tag{5.9a}$$

$$\text{s.t.} \quad \sum_{j\in\mathcal{J}} Q_j \cdot y_{jh\ell kt} \le A_h \sum_{p\in P_{h\ell kt}} \alpha_p, \quad \forall t \in [\![1,T]\!],\ \forall \ell \in \mathcal{L},\ \forall k \in \mathcal{L}_{h\ell t}, \tag{5.9b}$$

$$\sum_{p\in\mathcal{P}_h} \alpha_p = 1, \tag{5.9c}$$

$$\alpha_p \in \mathbb{R}_{\ge 0}, \qquad \forall p \in \mathcal{P}_h, \tag{5.9d}$$

$$y_{jh\ell kt} \in \mathbb{R}_{\ge 0} \qquad \forall t \in [\![1,T]\!],\ \forall j \in \mathcal{J},\ \forall \ell \in \mathcal{L},\ \forall k \in \mathcal{L}_{h\ell t}, \tag{5.9e}$$

Therefore we can view the optimization over x, v, y as instead an optimization problem over α_p, y, again all real. The objective remains linear, corresponding to the accumulated $\sum_j (\lambda_{j,t+N_{h\ell k},k} - \lambda_{j,t,\ell}) y_j$ along each path x_p between consecutive locations h, k. By standard LP theory, we know an optimal solution exists at an extreme point. It remains to show that the set of extreme points for the feasible region of 5.9 are comprised of standard basis vectors $\vec{e}_i$, which in this case corresponds to single paths.

To show this, consider some solution $\{\alpha_p\}, \{y_{jh\ell kt}\}$ to 5.9 which has some $\alpha_i \in (0,1)$ for some path i. We construct two solutions $\{\beta_p\}, \{x_{jh\ell kt}\}$, $\{\gamma_p\}, \{z_{jh\ell kt}\}$ such that $\{\alpha_p\}, \{y_{jh\ell kt}\}$ is a convex combination of these two solutions with nonnegative weights.

First, consider the set of β_p defined by defined by $\beta_p = \frac{1}{1-\alpha_i}\alpha_p$ for all $n \in \mathcal{P}_h$, $p \ne i$ and $\beta_i = 0$. Also, consider $x_{jh\ell kt} = \frac{1}{1-\alpha_i} \frac{\sum_{p\ne i, p\in P_{h\ell kt}} \alpha_p}{\sum_{p\in P_{h\ell kt}} \alpha_p} x_{jh\ell kt}$ for all $j \in \mathcal{J}, \ell \in L, k \in \mathcal{L}_{h\ell t}, t = 1, \ldots, T$.

Observe that this satisfies the constraints: 5.9b is satisfied because

$$\begin{aligned}\sum_{j\in\mathcal{J}} Q_j \cdot x_{jh\ell kt} &= \frac{1}{1-\alpha_i}\frac{\sum_{p\neq i, p\in P_{h\ell kt}}\alpha_p}{\sum_{p\in P_{h\ell kt}}\alpha_p}\sum_{j\in\mathcal{J}} Q_j \cdot y_{jh\ell kt} \\ &\leq \frac{1}{1-\alpha_i}\frac{\sum_{p\neq i, p\in P_{h\ell kt}}\alpha_p}{\sum_{p\in P_{h\ell kt}}\alpha_p} A_h \sum_{p\in P_{h\ell kt}}\alpha_p \\ &= A_h \frac{1}{1-\alpha_i}\sum_{p\neq i, p\in P_{h\ell kt}}\alpha_p \\ &= A_h \sum_{p\in P_{h\ell kt}}\beta_p,\end{aligned}$$

where the first inequality comes from $\{\alpha_p\}, \{y_{jh\ell kt}\}$ being a feasible solution to 5.9, and 5.9c is satisfied because

$$\sum_{p\in\mathcal{P}_h}\beta_p = \sum_{p\neq i, p\in\mathcal{P}_h}\frac{1}{1-\alpha_i}\alpha_p = \frac{1-\alpha_i}{1-\alpha_i} = 1,$$

and the nonnegativity constraints are immediate due to all of the multipliers being nonnegative.

Next, we consider $\{\gamma_p\}, \{z_{jh\ell kt}\}$ where $\{\gamma_p\} = \vec{e_i}$ (the vector with only a 1 for index i and 0 for all other paths p). This obviously satisfies the constraints 5.9c and 5.9d. Then consider $z_{jh\ell kt}$ defined as follows: for h, ℓ, k, t such that h travels arc ℓ, k at time t in path i, we set $z_{jh\ell kt} = \frac{y_{jh\ell kt}}{\sum_{p\in P_{h\ell kt}}\alpha_p}$ for all j. For all other arcs, we set $z_{jh\ell kt} = 0$. This clearly satisfies the nonnegativity constraint.

We verify 5.9b. For h, ℓ, k, t such that h travels arc ℓ, k at time t in path i, we have

$$\begin{aligned}\sum_{j\in\mathcal{J}} Q_j \cdot z_{jh\ell kt} &= \frac{1}{\sum_{p\in P_{h\ell kt}} \alpha_p} \sum_{j\in\mathcal{J}} Q_j \cdot y_{jh\ell kt} \\ &\leq A_h \frac{1}{\sum_{p\in P_{h\ell kt}} \alpha_p} \sum_{p\in P_{h\ell kt}} \alpha_p \\ &= A_h \\ &= A_h \sum_{p\in P_{h\ell kt}} \gamma_p,\end{aligned}$$

where we use the feasibility of $\{\alpha_p\}, \{y_{jh\ell kt}\}$ for 5.9, and the last equality holds since $i \in P_{h\ell kt}$.

For all other arcs, $\sum_{j\in\mathcal{J}} z_{jh\ell kt} = 0$, which is certainly less than a sum of non-negative numbers β_p.

Finally, we observe that for all p, $\alpha_p = \alpha_i\gamma_p + (1-\alpha_i)\beta_p$, and furthermore for all j, h, ℓ, k, t,

$$y_{jh\ell kt} = \alpha_i z_{jh\ell kt} + (1-\alpha_i) x_{jh\ell kt},$$

For h, ℓ, k, t such that h travels arc ℓ, k at time t in path i, we have

$$\alpha_i z_{jh\ell kt} + (1-\alpha_i) x_{jh\ell kt} = \frac{\alpha_i}{\sum_{p\in P_{h\ell kt}} \alpha_p} y_{jh\ell kt} + \frac{\sum_{p\neq i, p\in P_{h\ell kt}} \alpha_p}{\sum_{p\in P_{h\ell kt}} \alpha_p} y_{jh\ell kt} = y_{jh\ell kt},$$

and for all other arcs,

$$\alpha_i z_{jh\ell kt} + (1-\alpha_i) x_{jh\ell kt} = \frac{\sum_{p\in P_{h\ell kt}} \alpha_p}{\sum_{p\in P_{h\ell kt}} \alpha_p} y_{jh\ell kt} = y_{jh\ell kt}.$$

It follows by definition that the solution $\{\alpha_p\}, \{y_{jh\ell kt}\}$ is not an extreme point. Therefore the only possible extreme points are the standard basis vectors over the

set of paths P, i.e. flows along a single path.

It follows that optimizing over these paths of a single, binary-valued vehicle yields a solution with the same optimal solution as optimizing over the continuous variables representing the flow of the vehicle.

By further observing that the commodity-location problem 5.9 does not change between the MIP and its LP relaxation, we can therefore conclude that the optimal dual solution to 5.1 equals the optimal dual solution to the LP relaxation, which equals the optimal LP solution of the LP relaxation of 5.1.

□

We also have the following negative result on strong duality.

Theorem 5.5.2. *Strong duality does not hold in general between the dual solution and the primal solution to the MIP.*

Proof. To prove this negative result, it is sufficient to provide a counterexample. To that end, consider the following problem.

Suppose there are three locations, a, b, c, such that the only arcs are (a, b) and (a, c), both with distance 1 unit. Suppose there is only one vehicle h which starts at a at time 0, has capacity $A_h = 1$, and travels at the speed of 1 unit per time step. Let there be one commodity type j, with "weight" 1. Let $P_{j\ell} = 1$ for all j, ℓ. Finally, suppose there is a supply of 1 unit of j at a at time 0, and two demands: $\alpha \in (0, 1)$ due at b at time 1, and $1 - \alpha \in (0, 1)$ due at c at time 1.

First, observe the primal MIP solution is $\min(\alpha, 1 - \alpha) > 0$, since the vehicle can either satisfy no demand (incurring cost 1), satisfy the demand at b (incurring

cost $1-\alpha$), or satisfy the demand at c (incurring cost α).

Now consider the dual. We first consider the vehicle subproblem's contribution to the objective.

The vehicle can either stay at location a, incurring cost $\lambda_{j0a} - \lambda_{j1a}$, move to location b, incurring cost $\lambda_{j0a} - \lambda_{j1b}$, or move to location c, incurring cost $\lambda_{j0a} - \lambda_{j1c}$.

Now consider the commodity-location subproblem. Based on the closed-form solution described above, the $j-b$ subproblem incurs $\alpha \min(1, \lambda_{j1b})$, and the $j-c$ subproblem incurs $(1-\alpha)\min(1, \lambda_{j1c})$. Finally, the $j-a$ subproblem objective evaluates to the constant $-\lambda_{j0a}$.

Now consider three cases for the values of λ^*, the optimal set of prices which maximize the dual. First, suppose $\lambda^*_{j1a} \geq \max(\lambda^*_{j1b}, \lambda^*_{j1c})$, so that an optimal solution for the vehicle is to stay at location a. In this case, the total cost is $\lambda^*_{j0a} - \lambda^*_{j1a} + \alpha \min(1, \lambda^*_{j1b}) + (1-\alpha)\min(1, \lambda^*_{j1c}) - \lambda^*_{j0a} = -\lambda^*_{j1a} + \alpha \min(1, \lambda^*_{j1b}) + (1-\alpha)\min(1, \lambda^*_{j1c})$. Since the dual optimization is a maximization problem, the objective in this case is maximized by minimizing λ_{j1a} and maximizing λ_{j1b} and λ_{j1c}. However, based on our analysis of the commodity-location problem, we know λ^*_{j1a} is at most λ^*_{j0a}, so the optimal λ^* in this case has $\lambda^*_{j0a} = \lambda^*_{j1a}$. Moreover, $\lambda^*_{j1a} \geq \max(\lambda^*_{j1b}, \lambda^*_{j1c})$ by assumption, so the maximum value of λ_{j1b} and λ_{j1c} is also $\lambda^*_{j1a} = \lambda^*_{j0a}$. Therefore the total objective in this case is at most $-\lambda^*_{j0a} + \alpha \min(1, \lambda^*_{j0a}) + (1-\alpha)\min(1, \lambda^*_{j0a}) \leq -\lambda^*_{j0a} + \alpha 1, \lambda^*_{j0a} + (1-\alpha)\lambda^*_{j0a} = 0$.

Now suppose $\lambda^*_{j1b} \geq \max(\lambda^*_{j1a}, \lambda^*_{j1c})$ so that an optimal solution for the vehicle is to go to location b. The total cost is $\lambda^*_{j0a} - \lambda^*_{j1b} + \alpha \min(1, \lambda^*_{j1b}) + (1-\alpha)\min(1, \lambda^*_{j1c}) - \lambda^*_{j0a} = -\lambda^*_{j1b} + \alpha \min(1, \lambda^*_{j1b}) + (1-\alpha)\min(1, \lambda^*_{j1c}) \leq -\lambda^*_{j1b} + \alpha\lambda^*_{j1b} + (1-\alpha)\lambda^*_{j1c} \leq -\lambda^*_{j1b} + \alpha\lambda^*_{j1b} + (1-\alpha)\lambda^*_{j1b} = 0$.

By symmetry, the case that $\lambda^*_{j1c} \geq \max(\lambda^*_{j1a}, \lambda^*_{j1b})$ also has 0 as an upper bound on the objective.

Therefore in all cases, 0 is an upper bound on the dual objective. Finally, observe that setting $\lambda_{jt\ell} = 0$ for all $t \in \{0, 1\}$ and $\ell \in \{a, b, c\}$ trivially achieves this objective, since we can freely set $s_{jb1} = r_{jb1}$ and $s_{jc1} = r_{jc1}$. Therefore the maximum of the dual is 0, which is strictly less than the primal objective of $\min(\alpha, 1-\alpha)$.

□

Note that the counterexample from Theorem 5.5.2 can be easily verified to adhere to Theorem 5.5.1, since the continuously relaxed vehicle can split into α going to location b, and $1 - \alpha$ going to location c, satisfying both demands.

5.6 Producing a feasible solution with Multicommodity Flow

In practice, it is difficult for the solution of the inner minimization problem of the dual to yield a primal feasible solution to the original MIP.

In particular, given a set of dual variables λ, in solving the knapsack for the vehicles, the vehicles either move nothing, or fill up to maximum capacity. However, this will generally violate the balance constraint that is relaxed for the Lagrangian, since there is likely not enough inventory at the location for the vehicle to actually fill up to capacity.

However, in order for this method to be practically useful, it is important that we are able to derive a primal feasible solution to the MIP, so that we can

still ultimately provide a sensible solution to a logistics planner. To that end, we propose the following heuristic: fix the vehicle variables x, v, and solve the remaining MIP over the y, w, s, r variables. We call this the multicommodity flow (MCF) method, since one can view the vehicles as the capacitated edges over which the flow of commodities travel.

Practically speaking, this drastically sparsifies the network over which the logistics problem is optimized, as compared with the MIP. Section 5.7 includes experiments which demonstrate that this method can yield reasonable solutions even in large instances.

We also note that by deriving a primal feasible solution, we provide an upper bound on the optimal MIP solution, complementing the corresponding lower bound from the dual solution.

5.7 Numerical Results

On the same Southern California case study scenario used in Chapter 4, we demonstrate the effectiveness of our approach, which can find a dual solution matching the primal feasible objective value within minutes. Figure 5.1 shows the convergence of the dual objective to the primal optimal objective on this instance. We expect our solution to scale better with problem instance than solving with the MIP.

Table 5.1 further demonstrates the value of the decomposition method in solving large instance. We compare the MIP and decomposition code on the same Southern California instance, scaling up the size of the instance linearly (with the

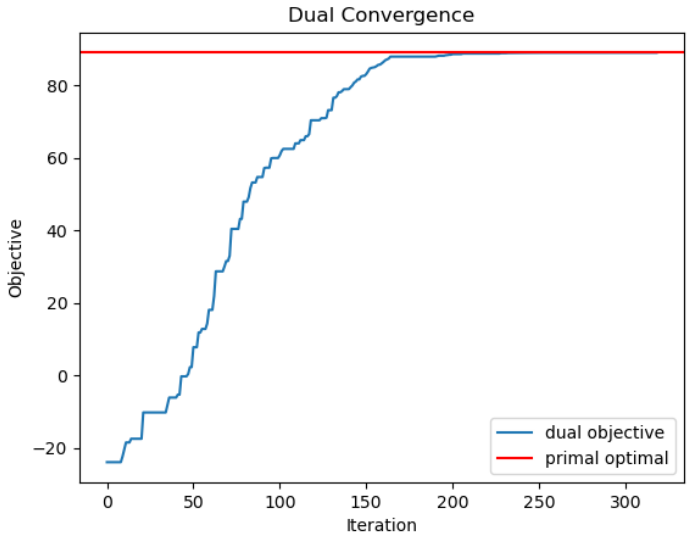


Figure 5.1: Convergence of our method on Southern California instance.

number of time steps). The decomposition code is run with 100 iterations, and then fed into the MCF algorithm from Section 5.6 to generate a primal feasible solution. We see that, although the derived primal solution is not as good as the MIP optimal solution, the time to solve does not scale nearly as poorly. In particular, for the 12 hour instance, the MIP got stuck at a feasible solution for 30 minutes which did not reach the specified optimality gap, and for the 24 hour instance, the MIP did not even find a feasible solution within the time limit.

Table 5.1: Comparison of MIP and Decomposition (100 iterations) + MCF across time step sizes on Southern California Instance. * MIP found feasible solution but did not solve to completion. ♯ no feasible solution found within time limit.

Hours per timestep	**MIP**		**Decomposition + MCF**	
	Obj	**Time (s)**	**Obj**	**Time (s)**
24	1.16×10^4	194.71	2.23×10^5	224.63
12	5.65×10^4	1840.00*	7.15×10^4	456.76
6	n.a.♯	n.a.♯	3.57×10^4	1178.73

5.8 Conclusion

In summary, our work makes three key contributions. First, we develop a fast, memory-efficient dual decomposition algorithm tailored to large-scale military logistics planning, leveraging problem structure to enable efficient subproblem solutions. Second, we show that our dual optimization achieves the LP relaxation objective. Third, we propose an effective heuristic for recovering a primal feasible solution, yielding upper and lower bounds on the optimal MIP objective. On realistic scenarios, we demonstrate that our approach converges to primal optimal objectives within minutes and enables solving instances that are computationally intractable as MIPs.

Future work includes achieving further speedup by implementing parallelization, as well as developing more intelligent and theoretically justified methods for deriving a primal feasible solution from a converged (or unconverged) dual solution.

APPENDIX A

CHAPTER 2 APPENDIX

A.1 Proofs and Other Technical Details

A.1.1 Justification of Assumption 5.2

We restate Theorem 14.5 from Anthony and Bartlett [2009].

Theorem A.1.1. *Assume a class F of functions computed by a feed-forward real-output multi-layer network with the following properties:*

- *The network has $\ell \geq 2$ layers with connections only between adjacent layers.*
- *There are W weights in the network.*
- *For some b, each neuron maps into the interval $[-b, b]$, and each neuron in the first layer has a non-decreasing activation function.*
- *There are constants $V > 0$ and $L > 1/V$ so that for each neuron in all but the first layer of the network, the vector w of weights associated with that neuron has $\|w\|_1 \leq V$ and its activation function $s : \mathbb{R} \mapsto [-b, b]$ is L-Lipschitz.*

Then, if $\alpha \leq 2b$,

$$N_\infty(\alpha, F, n) \leq \left(\frac{4enbW(LV)^\ell}{\alpha(LV-1)}\right)^W.$$

As stated in the main paper, consider a neural network with all neurons having non-decreasing, L-Lipschitz activation functions which map into $[-b, b]$, and the constraint on weights $\|w\|_1 \leq V$. Then in this case $B = b$ in Assumption 5.2a.

We conclude from Lemma 14.6 of Anthony and Bartlett [2009] that all functions F are $(VL)^\ell$-Lipschitz, i.e. $L' = (VL)^\ell$, satisfying the second part of Assumption 5.2a.

Finally, we consider the architecture of the neural network to be fixed, so that W is fixed (it is allowed to depend on the input dimension d, which we also consider fixed). Then we see this satisfies Assumption 5.2b with $C = \frac{4ebW(LV)^\ell}{LV-1}$ and $D = W$.

A.1.2 Proof of Proposition 6.1

Proof. We have that

$$\begin{aligned}\hat{V}(\pi) &= \frac{\sum_{i=1}^n \mathbb{1}\{\pi(x_i) = a_i\} r_i}{\sum_{i=1}^n \mathbb{1}\{\pi(x_i) = a_i\}} \\ &= \frac{\frac{1}{n}\sum_{i=1}^n \mathbb{1}\{\pi(x_i) = a_i\} r_i}{\frac{1}{n}\sum_{i=1}^n \mathbb{1}\{\pi(x_i) = a_i\}}. \end{aligned} \tag{A.1}$$

Now consider the denominator to Equation (A.1). We can express the event that $\pi(x_i) = a_i$ as the union of disjoint events $(\pi(x_i) = a) \cap (a_i = a)$, where the union is over the space of actions (articles) $a \in \{1, \ldots, 20\}$. Then since the event $a_i = a$ is independent of $\pi(x_i)$ (as the logging policy is known to be uniform), we can write

$$\begin{aligned}\mathbb{P}(\pi(x_i) = a_i) &= \sum_{a=1}^{20} \mathbb{P}\left(\pi(x_i) = a\right) \mathbb{P}\left(a_i = a\right) \\ &= \frac{1}{20} \sum_{a=1}^{20} \mathbb{P}\left(\pi(x_i) = a\right) \\ &= \frac{1}{20}.\end{aligned}$$

Therefore we can apply the (weak) law of large numbers on each $\mathbb{1}\{\pi(x_i) = a_i\}$, since they are independent and identically distributed Bernoulli random variables with $p = \frac{1}{20}$. So we have that

$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{\pi(x_i) = a_i\} \xrightarrow{p} \frac{1}{20}. \tag{A.2}$$

Now consider the inverse-propensity score (IPS) estimator,

$$\hat{V}_{IPS}(\pi) := \frac{\sum_{i=1}^{n} \mathbb{1}\{\pi(x_i) = a_i\} r_i}{\mathbb{P}(\pi_0(x_i) = a_i)},$$

where π_0 is the uniform sampling policy.

From Agarwal and Kakade [2019], this estimator is unbiased for $V(\pi)$ and has variance which scales as $O(\frac{1}{n})$. Therefore by Chebyshev's inequality,

$$\mathbb{P}\left(|\hat{V}_{IPS}(\pi) - V(\pi)| \geq \epsilon\right) \leq \frac{\mathrm{Var}(\hat{V}_{IPS}(\pi))}{\epsilon^2} \overset{n\to\infty}{\to} 0,$$

and we conclude that $\hat{V}_{IPS}(\pi)$ is consistent for $V(\pi)$.

Since $\mathbb{P}(\pi_0(x_i) = a_i) = \frac{1}{20}$, it follows that the numerator for Equation (A.1) is consistent for $\frac{V(\pi)}{20}$. We can finally combine this with Equation (A.2) and apply Slutzky's theorem to conclude that $\hat{V}(\pi)$ is consistent for $V(\pi)$. □

A.1.3 Example where ESR should outperform policy-based methods.

Consider a simple example: let $r(x, a) = g(x) + \mathbb{1}(a = a^*)$ for some fixed a^* where g is continuous but has a large $\mathrm{Var}(g(x))$ due to variability in x. Consider

the asymptotic regime where the number of datapoints (n) and the smoothing parameter (k) are both large. Because n is large and g is continuous, the nearest neighbors $N(i)$ will have similar contexts and the ESR (2.3) will be close to (2.2). Moreover, when k is large, (2.2) is near 0 for any θ that is able to predict a larger reward $\hat{r}_\theta(x_i, a)$ for $a = a^*$ than for the other actions $a \neq a^*$. As long as the model class contains such a θ, minimizing ESR will produce a model whose regret is small. On the other hand, a policy-based method based on inverse propensity weights may struggle to learn a good policy because large $\text{Var}(g(x))$ creates variability in the reward, obscuring the effect of the action a_i and making it more difficult to learn the best policy.

Moreover, (2.2) has a simple form – it is just $1 - \frac{\exp(k\hat{r}_\theta(x_i,1))}{\sum_{a' \in \mathcal{A}} \exp(k\hat{r}_\theta(x_i,a'))}$. so that nearest neighbors are close. Because the ESR loss function takes differences between rewards for different actions, the nearest neighbors will form a good approximation to the difference between actions. Indeed, the difference in rewards will be either 1 or 0 plus some vanishing error term due to the use of neighbors. In other words, we are essentially training a reward model where regret is (approximately) 0 only when $\hat{r}_\theta(x, a) \leq \hat{r}_\theta(x, \text{argmax}_{a'} r(x, a'))\ \forall a$ and (approximately) 1 otherwise. For general model classes, this is easy to learn since there likely exist many possible reward models in the class which satisfy this. On the other hand, using a policy-based method still relies on having $r_i = g(a_i) + \mathbb{1}(a_i = 1)$ in the loss function, and given $\text{Var}(g(x))$ is large, the large variance of g may obscure the effect of the action a_i and thereby make it more difficult to learn the best policy.

.

Table A.1: CTR (95% CI) for ESR Loss Trained NN vs. other offline contextual bandit methods. The mean click-through rate (CTR) for most days is highest for the model trained on the ESR loss. The performance improvement is statistically significant over all ten days.

Day	ESR	Direct	IPW	DR	BanditNet	Pseudo-Loss	LS	LSE
1	[4.12, 5.13]	[3.50, 4.45]	[3.88, 4.24]	[3.87, 4.23]	[4.02, 4.40]	[3.86, 4.56]	[4.01, 4.39]	[3.84, 4.20]
2	[3.13, 4.06]	[2.54, 3.54]	[3.03, 3.22]	[3.04, 3.28]	[3.13, 3.34]	[3.06, 3.26]	[3.10, 3.33]	[3.04, 3.26]
3	[3.38, 4.55]	[2.75, 3.51]	[3.06, 3.37]	[3.09, 3.40]	[3.12, 3.43]	[3.08, 3.38]	[3.16, 3.46]	[3.07, 3.40]
4	[3.78, 4.82]	[3.62, 4.48]	[3.75, 4.15]	[3.73, 4.16]	[3.90, 4.33]	[3.76, 4.16]	[3.88, 4.32]	[3.76, 4.17]
5	[4.25, 5.46]	[3.69, 4.64]	[3.97, 4.26]	[4.04, 4.35]	[4.13, 4.43]	[3.99, 4.28]	[4.13, 4.46]	[4.05, 4.30]
6	[3.97, 5.29]	[3.38, 4.20]	[3.53, 3.90]	[3.50, 3.88]	[3.60, 3.98]	[3.51, 3.88]	[3.64, 4.00]	[3.51, 3.89]
7	[3.34, 4.42]	[3.13, 4.18]	[3.66, 3.99]	[3.68, 4.03]	[3.78, 4.12]	[3.69, 4.06]	[3.80, 4.17]	[3.66, 4.00]
8	[3.22, 4.09]	[3.28, 4.34]	[3.28, 3.56]	[3.31, 3.57]	[3.44, 3.74]	[3.32, 3.59]	[3.41, 3.70]	[3.30, 3.58]
9	[3.41, 4.36]	[2.89, 3.83]	[3.11, 3.36]	[3.10, 3.37]	[3.24, 3.51]	[3.10, 3.35]	[3.24, 3.51]	[3.11, 3.38]
10	[3.14, 4.01]	[3.13, 3.95]	[3.09, 3.43]	[3.13, 3.48]	[3.19, 3.56]	[3.08, 3.45]	[3.19, 3.56]	[3.09, 3.42]
All	[3.97, 4.32]	[3.55, 3.86]	[3.58, 3.71]	[3.60, 3.73]	[3.71, 3.84]	[3.59, 3.72]	[3.71, 3.84]	[3.59, 3.71]

A.2 Experimental Setup

The IHDP dataset was run on a M1 Pro Macbook Pro with 16GB RAM. The news recommender dataset was run on a cluster allocated 1 Intel(R) Xeon(R) Silver 4214R CPU @ 2.40GHz core, 1 RTX 3090 GPU, and 64GB RAM.

To ensure reproducibility, random seeds were set for both `numpy` and `torch`.

A.3 Additional Experimental Results

A.3.1 Day-by-day Results

Table A.1 shows a more detailed view of the results in Figures 1 and 2.

A.3.2 BanditNet and Pseudo-Loss Hyperparameters

In the main test, we compared against BanditNet [Joachims et al., 2018] and the Pseudo-Loss regularizer [Wang et al., 2024]. We chose the best-performing setting of hyperparameter out of a grid; here we show performance of all hyperparameters tested.

For BanditNet, we test λ in the range $[0.65, 1.05]$ and for Pseudo-Loss, we test β in the range $[0.001, 1]$ to match the range the authors test in their own experiments. Figure A.1 shows that the performance of these methods is not particularly sensitive to the choice of the hyperparameter.

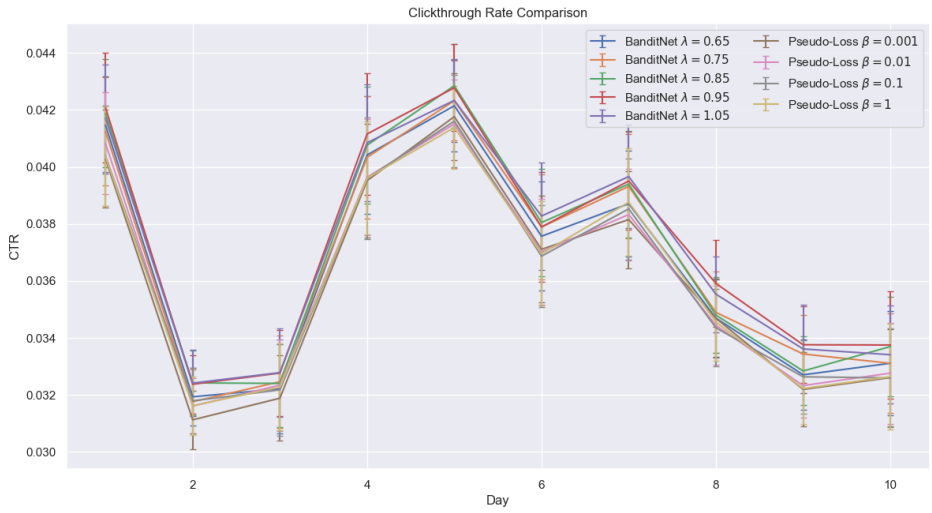


Figure A.1: Effect of hyperparameters for BanditNet and Pseudo-Loss

A.3.3 DR with Shrinkage

Figure A.2 shows results on the Yahoo dataset when using the optimistic shrinkage approach from Su et al. [2020]. Su et al. [2020] uses the same equation for estimat-

ing the policy value as in the DR approach. In particular, defining $w(x,a) = \frac{\pi(a|x)}{\pi_0(a|x)}$, standard DR uses $\frac{1}{n}\sum_{i=1}^{n}\left(\sum_{a\in\mathcal{A}}\hat{r}(x_i,a)\pi(a|x_i) + w(x_i,a_i)(r_i - \hat{r}(x_i,a_i)\right)$, while DR with optimistic shrinkage uses $\frac{1}{n}\sum_{i=1}^{n}\left(\sum_{a\in\mathcal{A}}\hat{r}(x_i,a)\pi(a|x_i) + \hat{w}_\lambda(x_i,a_i)(r_i - \hat{r}(x_i,a_i)\right)$ where $\hat{w}_\lambda(x,a) := \frac{\lambda}{w(x,a)^2+\lambda}w(x,a)$. We show results for $\lambda \in \{0.1, 1, 10, 100, 1000, \infty\}$ ($\lambda = 0$ is the same as the direct method).

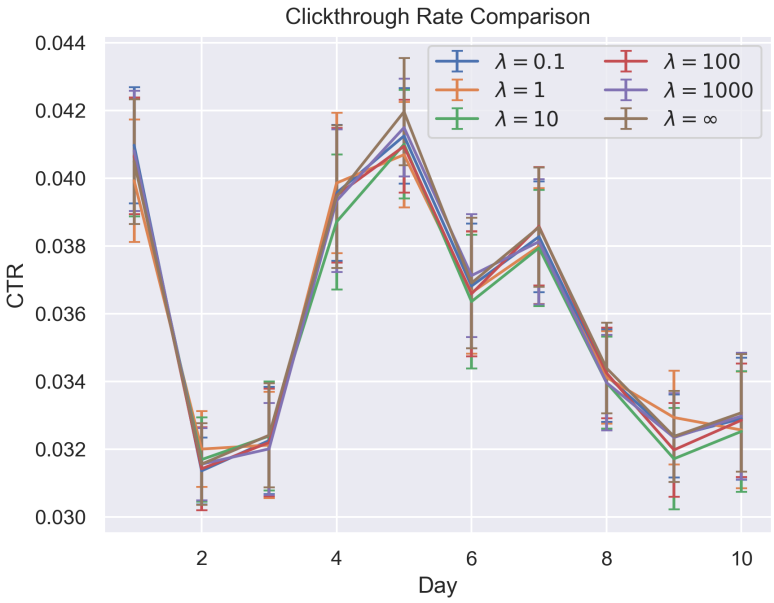


Figure A.2: Comparing choices of λ on DR with optimistic shrinkage.

A.3.4 ESR with KD-Tree

Figure A.3 shows results when replacing the Ball tree algorithm [Omohundro, 1989] used in the results shown in the main body with KD tree [Bentley, 1975]. The results are different but the overall performance is similar.

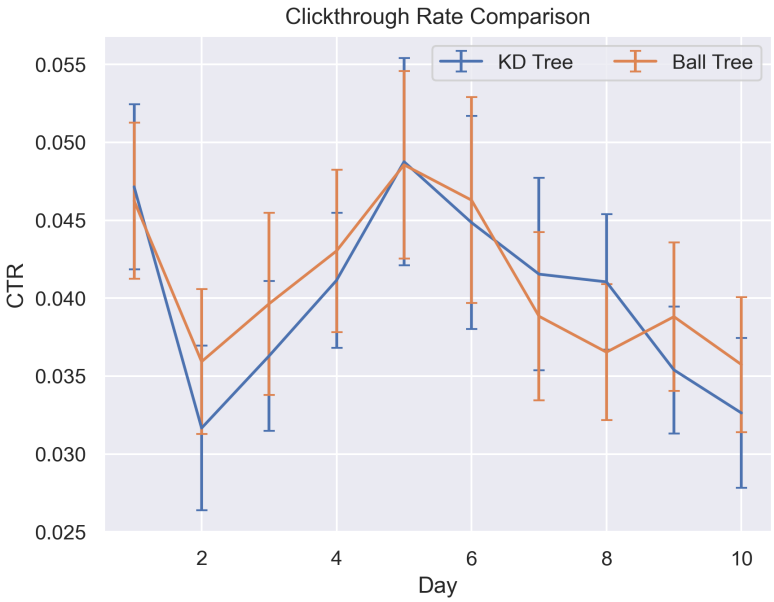


Figure A.3: Comparing effect of approximate nearest neighbors algorithm on ESR.

A.4 Code

Please find the code for the experiments at the repository here: `https://github.com/samstan/ESRLoss`

APPENDIX A

CHAPTER 3 APPENDIX

A.1 Wang-Landau Algorithm

The Wang-Landau method is a Monte Carlo algorithm designed to efficiently estimate the density of states $D(E)$ across the entire energy spectrum of a system. Unlike conventional Monte Carlo methods that sample according to the Boltzmann distribution, the Wang-Landau algorithm performs a random walk in energy space with sampling probability proportional to $\frac{1}{D(E)}$, ensuring uniform exploration of all energy levels. The method iteratively refines both the density of states estimate and a modification factor f, which is reduced as the histogram of visited energies becomes increasingly flat. The algorithm terminates when the modification factor reaches a predetermined threshold, yielding an accurate estimate of $D(E)$ that enables direct calculation of thermodynamic quantities at any temperature.

Algorithm 4: Wang-Landau Algorithm

Input: Initial configuration, modification factor $f_0 = e$, final factor f_{final}, flatness criterion δ

Output: Density of states $D(E)$

Initialize $D(E) = 1$ for all energies E; Initialize histogram $H(E) = 0$ for all energies E; Set $f = f_0$; **while** $f > f_{final}$ **do**

  while *histogram is not flat* **do**

    Propose a Monte Carlo move to new configuration with energy E_{new}; Calculate acceptance probability $p = \min\left(1, \frac{D(E_{\text{old}})}{D(E_{\text{new}})}\right)$; **if** *move accepted* **then**

      Update configuration to new state; $E \leftarrow E_{\text{new}}$;

    end

    Update: $D(E) \leftarrow D(E) \cdot f$; Update: $H(E) \leftarrow H(E) + 1$;

  end

  Check flatness: $\frac{\min(H(E))}{\langle H(E) \rangle} > 1 - \delta$; Reset histogram: $H(E) = 0$ for all E; Reduce modification factor: $f \leftarrow \sqrt{f}$;

end

return $D(E)$

APPENDIX B

CHAPTER 4 APPENDIX

B.1 Hub-and-Spoke Algorithm

Here we describe the automatic hub-and-spoke generation algorithm in detail. We run Algorithm 5 for each domain (Air, Sea, Ground) to generate hubs and spokes. Then we construct the network by including all arcs between hubs, i.e. (m_j, m_ℓ) for all $j, \ell = \{1, \ldots, k\}, j \neq \ell$ and all arcs between hubs and spokes in the same cluster C_j, i.e. $(x_i, m_j), (m_j, x_i)$ for $x_i \in C_j \setminus m_j$, for $j = 1, \ldots, k$.

Algorithm 5: Automatic Single-domain Hub-and-Spoke generation based on Park and Jun [2009]

Data: $X = \{x_1, x_2, \ldots, x_n\}$: Set of n locations

k: Number of hubs

D: Precomputed distance matrix where D_{ij} is the physical distance between x_i, x_j for all $i, j \in \{1, 2, \ldots, n\}$

Result: Set of k hubs and $n - k$ spokes

Step 1: Compute the dissimilarity measure for each object:

for $i = 1$ **to** n **do**

  $D_i \leftarrow \sum_{j=1}^{n} D_{ij}$;

end

Step 2: Select k initial hubs:

Sort location in ascending order based on D_i;

Select the first k location as initial hubs $\{m_1, m_2, \ldots, m_k\}$;

Step 3: Assign each location to the nearest hub:

for $i = 1$ **to** n **do**

  Assign x_i to cluster C_j where $j = \arg\min_{l \in \{1,2,\ldots,k\}} D_{i,m_l}$;

end

Step 4: Update hubs:

for $j = 1$ **to** k **do**

  $m_j \leftarrow \arg\min_{x \in C_j} \sum_{y \in C_j} d(x, y)$;

end

Step 5: Repeat Steps 4 and 5 until hubs no longer change;

return *Hubs* $\{m_1, m_2, \ldots, m_k\}$ *and Spokes* $C_j \setminus m_j, j = 1, \ldots, k$

BIBLIOGRAPHY

Alberto Abadie and Guido W Imbens. Large sample properties of matching estimators for average treatment effects. *econometrica*, 74(1):235–267, 2006.

Alekh Agarwal and Sham Kakade. Lecture notes in off-policy evaluation and learning. `https://courses.cs.washington.edu/courses/\\cse599m/19sp/notes/off\_policy.pdf`, May 2019.

Akshay Agrawal, Stephen Boyd, Deepak Narayanan, Fiodar Kazhamiaka, and Matei Zaharia. Allocation of fungible resources via a fast, scalable price discovery method. *Mathematical Programming Computation*, 14(3):593–622, 2022.

Mawulolo K Ameko, Miranda L Beltzer, Lihua Cai, Mehdi Boukhechba, Bethany A Teachman, and Laura E Barnes. Offline contextual multi-armed bandits for mobile health interventions: A case study on emotion regulation. In *Proceedings of the 14th ACM Conference on Recommender Systems*, pages 249–258, 2020.

J. Andersen, T.G. Crainic, and M. Christiansen. Service network design with management and coordination of multiple fleets. *European Journal of Operational Research*, 193(2):377–389, 2009.

Martin Anthony and Peter L Bartlett. *Neural network learning: Theoretical foundations*. cambridge university press, 2009.

Uday Apte. Navy expeditionary logistics. Technical report, Acquisition Research Program, 2018.

Susan Athey and Stefan Wager. Policy learning with observational data. *Econometrica*, 89(1):133–161, 2021.

Turgut Aykin. Lagrangian relaxation based approaches to capacitated hub-and-spoke network design problem. *European journal of operational research*, 79(3): 501–523, 1994.

R. Bai, S.W. Wallace, J. Li, and A.Y.L. Chong. Stochastic service network design with rerouting. *Transportation Research Part B: Methodological*, 60:50–65, 2014.

Steven F Baker, David P Morton, Richard E Rosenthal, and Laura Melody Williams. Optimizing military airlift. *Operations Research*, 50(4):582–602, 2002.

I. Bakir, A.L. Erera, and M.W.P. Savelsbergh. Motor carrier service network design. In T.G. Crainic, M. Gendreau, and B. Gendron, editors, *Network Design with Applications to Transportation and Logistics*, chapter 14, pages 427–467. Springer, 2021.

Ilyes Batatia, David P Kovacs, Gregor Simm, Christoph Ortner, and Gábor Csányi. Mace: Higher order equivariant message passing neural networks for fast and accurate force fields. *Advances in neural information processing systems*, 35: 11423–11436, 2022.

Armin Behnamnia, Gholamali Aminian, Alireza Aghaei, Chengchun Shi, Vincent YF Tan, and Hamid R Rabiee. Batch learning via log-sum-exponential estimator from logged bandit feedback. In *ICML 2024 Workshop: Aligning Reinforcement Learning Experimentalists and Theorists*.

Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.

David H. Berger. Force design 2030, March 2020. The United States Marine Corps (Report), `https://www.hqmc.marines.mil/Portals/142/Docs/CMC38%20Force%20Design%202030%20Report%20Phase%20I%20and%20II.pdf`.

Alina Beygelzimer and John Langford. The offset tree for learning with partial labels. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 129–138, 2009.

N. Boland, M. Hewitt, L. Marshall, and M.W.P. Savelsbergh. The continuous-time service network design problem. *Operations research*, 65(5):1303–1321, 2017. ISSN 0030-364X.

N. Boland, M. Hewitt, L. Marshall, and M.W.P. Savelsbergh. The price of discretizing time: a study in service network design. *EURO Journal on Transportation and Logistics*, 8(2):195–216, 2019. ISSN 2192-4376. doi: https://doi.org/10.1007/s13676-018-0119-x. URL `https://www.sciencedirect.com/science/article/pii/S2192437620300327`. Special Issue: Advances in vehicle routing and logistics optimization: exact methods.

Kevin Borden. *Optimizing the number and employment of combat logistics force shuttle ships, with a case study of the new T-AKE ship*. PhD thesis, MS thesis, Naval Postgraduate School, Monterey, CA, 2001.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1): 1–122, 2011.

James C. Bradford. The missing link: Expeditionary logistics. *Naval History*, 20 (1), February 2006.

David Brandfonbrener, William Whitney, Rajesh Ranganath, and Joan Bruna. Offline contextual bandits with overparameterized models. In *International Conference on Machine Learning*, pages 1049–1058. PMLR, 2021.

Gerald G Brown and W Matthew Carlyle. Optimizing the us navy's combat logistics force. *Naval Research Logistics (NRL)*, 55(8):800–810, 2008.

Gerald G Brown and Jeffrey E Kline. Optimizing navy mission planning. *Military Operations Research*, 26(2):39–58, 2021.

Gerald G Brown, W Matthew Carlyle, Robert F Dell, and John W Brau Jr. Optimizing intratheater military airlift in iraq and afghanistan. *Military Operations Research*, 18(3):35–52, 2013.

Gerald G Brown, Walter C DeGrange, Wilson L Price, and Anton A Rowe. Scheduling combat logistics force replenishments at sea for the us navy. *Naval Research Logistics (NRL)*, 64(8):677–693, 2017.

Deborah L Bryan and Morton E O'kelly. Hub-and-spoke networks in air transportation: an analytical review. *Journal of regional science*, 39(2):275–295, 1999.

W Matthew Carlyle, Johannes O Royset, and R Kevin Wood. Lagrangian relaxation and enumeration for solving constrained shortest-path problems. *Networks: an international journal*, 52(4):256–270, 2008.

Sougata Chaudhuri, Abraham Bagherjeiran, and James Liu. Ranking and calibrating click-attributed purchases in performance display advertising. In *Proceedings of the ADKDD'17*, pages 1–6. 2017.

Thomas W. Chavers. Operational logistics planner for the modular army: the newest update of the operational logistics planner incorporates the modular army organization into the planning system. *Army Logistician*, 41(1):51, January 2009.

Wei Chu, Seung-Taek Park, Todd Beaupre, Nitin Motgi, Amit Phadke, Seinjuti Chakraborty, and Joe Zachariah. A case study of behavior-driven conjoint analysis on yahoo! front page today module. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1097–1104, 2009.

Teodor Gabriel Crainic and Mike Hewitt. *Service Network Design*, pages 347–382. Springer International Publishing, Cham, 2021.

Teodor Gabriel Crainic, Antonio Frangioni, and Bernard Gendron. Bundle-based relaxation methods for multicommodity capacitated fixed charge network design. *Discrete Applied Mathematics*, 112(1-3):73–99, 2001.

T.G. Crainic. Service network design in freight transportation. *European Journal of Operational Research*, 122(2):272–288, 2000. ISSN 0377-2217. doi: https://doi.org/10.1016/S0377-2217(99)00233-7. URL https://www.sciencedirect.com/science/article/pii/S0377221799002337.

T.G. Crainic and J.M. Rousseau. Multicommodity, multimode freight transportation: A general modeling and algorithmic framework for the service network design problem. *Transportation Research Part B: Methodological*, 20(3):225–242, 1986. ISSN 0191-2615. doi: https://doi.org/10.1016/0191-2615(86)90019-6. URL https://www.sciencedirect.com/science/article/pii/0191261586900196.

T.G. Crainic and J. Roy. Or tools for tactical freight transportation planning. *European Journal of Operational Research*, 33(3):290–297, 1988. ISSN 0377-2217. doi: https://doi.org/10.1016/0377-2217(88)90172-5. URL https://www.sciencedirect.com/science/article/pii/0377221788901725.

T.G. Crainic, J.A. Ferland, and J.M. Rousseau. A tactical planning model for rail freight transportation. *Transportation Science*, 18(2):165–184, 1984. doi: 10.1287/trsc.18.2.165. URL `https://doi.org/10.1287/trsc.18.2.165`.

T.G. Crainic, M. Hewitt, M. Toulouse, and D.M. Vu. Service network design with resource constraints. *Transportation Science*, 50(4):1380–1393, 2016. doi: 10.1287/trsc.2014.0525. URL `https://doi.org/10.1287/trsc.2014.0525`.

George B Dantzig. Discrete-variable extremum problems. *Operations research*, 5 (2):266–288, 1957.

Bruno Alessi de Castro, Pablo Gustavo Cogo Pochmann, and Eduardo Borba Neves. Artificial intelligence applications in military logistics operations. In *Multidisciplinary International Conference of Research Applied to Defense and Security*, pages 89–100. Springer, 2023.

E. Demir, W. Burgholzer, M. Hrušovskỳ, E. Arıkan, W. Jammernegg, and T. Van Woensel. A green intermodal service network design problem with travel time uncertainty. *Transportation Research Part B: Methodological*, 93:789–807, 2016.

Vincent Dorie. Npci: Non-parametrics for causal inference, 2016. *URL https://github. com/vdorie/npci*, 2016.

Miroslav Dudík, Dumitru Erhan, John Langford, and Lihong Li. Doubly robust policy evaluation and optimization. 2014.

Adam N Elmachtoub and Paul Grigas. Smart "predict, then optimize". *Management Science*, 68(1):9–26, 2022.

A.L. Erera, M. Hewitt, M.W.P. Savelsbergh, and Y. Zhang. Improved load plan

design through integer programming based local search. *Transportation science*, 47(3):412–427, 2013. ISSN 0041-1655.

Hugh Everett III. Generalized lagrange multiplier method for solving problems of optimum allocation of resources. *Operations research*, 11(3):399–417, 1963.

Bolin Gao and Lacra Pavel. On the properties of the softmax function with application in game theory and reinforcement learning. *arXiv preprint arXiv:1704.00805*, 2017.

Bernard Gendron and Teodor Gabriel Crainic. Relaxations for multicommodity capacitated network design problems. Technical report, 1994.

Bernard Gendron, Teodor Gabriel Crainic, and Antonio Frangioni. Multicommodity capacitated network design. In *Telecommunications network planning*, pages 1–19. Springer, 1999.

Feliciano Giustino. *Materials modelling using density functional theory: properties and predictions*. Oxford University Press, 2014.

Damian A. Green. The logistics estimation workbook: 18 years and counting. *Army Sustainment*, November 2016.

Lacy M. Greening, Mathieu Dahan, and Alan L. Erera. Lead-time-constrained middle-mile consolidation network design with fixed origins and destinations. *Transportation Research Part B: Methodological*, 174:102782, 2023.

László Györfi, Michael Kohler, Adam Krzyżak, and Harro Walk. *A distribution-free theory of nonparametric regression*. Springer, 2002.

M. Hewitt. Enhanced dynamic discretization discovery for the continuous time

load plan design problem. *Transportation science*, 53(6):1731–1750, 2019. ISSN 0041-1655.

M. Hewitt. The flexible scheduled service network design problem. *Transportation Science*, 56(4):1000–1021, 2022. doi: 10.1287/trsc.2021.1114. URL `https://doi.org/10.1287/trsc.2021.1114`.

Jennifer L Hill. Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, 20(1):217–240, 2011.

Yoyo Hinuma, Hiroyuki Hayashi, Yu Kumagai, Isao Tanaka, and Fumiyasu Oba. Comparison of approximations in density functional theory calculations: Energetics and structure of binary oxides. *Physical Review B*, 96(9):094102, 2017.

Jean-Baptiste Hiriart-Urruty and Claude Lemaréchal. *Convex analysis and minimization algorithms I: Fundamentals*, volume 305. Springer science & business media, 1996.

Yichun Hu, Nathan Kallus, Xiaojie Mao, and Yanchen Wu. Contextual linear optimization with bandit feedback. *arXiv preprint arXiv:2405.16564*, 2024.

Yong Huang, Rui Cao, and Amir Rahmani. Reinforcement learning for sepsis treatment: A continuous action space solution. In *Machine Learning for Healthcare Conference*, pages 631–647. PMLR, 2022.

A. Jarrah, E.L. Johnson, and L.C. Neubert. Large-scale, less-than-truckload service network design. *Operations research*, 57(3):609–625, 2009. ISSN 0030-364X.

Olivier Jeunen and Bart Goethals. Pessimistic reward models for off-policy learning in recommendation. In *Proceedings of the 15th ACM Conference on Recommender Systems*, pages 63–74, 2021.

Thorsten Joachims, Adith Swaminathan, and Maarten De Rijke. Deep learning with logged bandit feedback. In *International Conference on Learning Representations*, 2018.

Fredrik Johansson, Uri Shalit, and David Sontag. Learning representations for counterfactual inference. In *International conference on machine learning*, pages 3020–3029. PMLR, 2016.

Fredrik D Johansson, Uri Shalit, Nathan Kallus, and David Sontag. Generalization bounds and representation learning for estimation of potential outcomes and causal effects. *Journal of Machine Learning Research*, 23(166):1–50, 2022.

Joint Chiefs of Staff. Planner's handbook for operational design. `https://www.jcs.mil/Portals/36/Documents/Doctrine/pams_hands/opdesign_hbk.pdf#:~:text=URL%3A%20https%3A%2F%2Fwww.jcs.mil%2FPortals%2F36%2FDocuments%2FDoctrine%2Fpams_hands%2Fopdesign_hbk.pdf%0AVisible%3A%200%25%20`, October 2011. Accessed: 2024-07-29.

Nathan Kallus. Recursive partitioning for personalization using observational data. In *International conference on machine learning*, pages 1789–1798. PMLR, 2017.

Nathan Kallus. Balanced policy evaluation and learning. *Advances in neural information processing systems*, 31, 2018.

Nathan Kallus. More efficient policy learning via optimal retargeting. *Journal of the American Statistical Association*, 116(534):646–658, 2021.

Nathan Kallus, Xiaojie Mao, Kaiwen Wang, and Zhengyuan Zhou. Doubly robust distributionally robust off-policy evaluation and learning. In *International Conference on Machine Learning*, pages 10598–10632. PMLR, 2022.

Athanasia Karakitsiou and Athanasios Migdalas. A decentralized coordination mechanism for integrated production–transportation–inventory problem in the supply chain using lagrangian relaxation. *Operational Research*, 8(3):257–278, 2008.

Edward H Kennedy. Towards optimal doubly robust estimation of heterogeneous causal effects. *Electronic Journal of Statistics*, 17(2):3008–3049, 2023.

Paul S Killingsworth and Laura Melody. Should c-17s be used to carry in-theater cargo during major deployments? Technical report, 1997.

Toru Kitagawa and Aleksey Tetenov. Who should be treated? empirical welfare maximization methods for treatment choice. *Econometrica*, 86(2):591–616, 2018.

John G Klincewicz and Hanan Luss. A lagrangian relaxation heuristic for capacitated facility location with single-source constraints. *Journal of the Operational Research Society*, 37(5):495–500, 1986.

Niklas Kohl and Oli BG Madsen. An optimization algorithm for the vehicle routing problem with time windows based on lagrangian relaxation. *Operations research*, 45(3):395–406, 1997.

Sören R Künzel, Jasjeet S Sekhon, Peter J Bickel, and Bin Yu. Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the national academy of sciences*, 116(10):4156–4165, 2019.

Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670, 2010.

Aristidis Likas, Nikos Vlassis, and Jakob J Verbeek. The global k-means clustering algorithm. *Pattern recognition*, 36(2):451–461, 2003.

C.C. Lin. The freight routing problem of time-definite freight delivery common carriers. *Transportation Research Part B: Methodological*, 35(6):525–547, 2001. ISSN 0191-2615. doi: https://doi.org/10.1016/S0191-2615(00)00008-4. URL https://www.sciencedirect.com/science/article/pii/S0191261500000084.

Zhexiao Lin, Peng Ding, and Fang Han. Estimation based on nearest neighbor matching: from density ratio to average treatment effect. *Econometrica*, 91(6):2187–2217, 2023.

K. Lindsey, A.L. Erera, and M.W.P. Savelsbergh. Improved integer programming-based neighborhood search for less-than-truckload load plan design. *Transportation science*, 50(4):1360–1379, 2016. ISSN 0041-1655.

A.G. Lium, T.G. Crainic, and S.W. Wallace. A study of demand stochasticity in service network design. *Transportation Science*, 43(2):144–157, 2009.

Monirehalsadat Mahmoudi and Xuesong Zhou. Finding optimal solutions for vehicle routing problem with pickup and delivery services with time windows: A dynamic programming approach based on state–space–time network representations. *Transportation Research Part B: Methodological*, 89:19–42, 2016.

Jayanta Mandi, Vıctor Bucarey, Maxime Mulamba Ke Tchomba, and Tias Guns. Decision-focused learning: through the lens of learning to rank. In *International Conference on Machine Learning*, pages 14935–14947. PMLR, 2022.

L. Marshall, N. Boland, M.W.P. Savelsbergh, and M. Hewitt. Interval-based dynamic discretization discovery for solving the continuous-time service network design problem. *Transportation science*, 55(1):29–51, 2021. ISSN 0041-1655.

Jim Mattis. Summary of the 2018 national defense strategy of the United States of America, 2018. The United States Department of Defense (Report), `https://apps.dtic.mil/sti/pdfs/AD1045785.pdf`.

Terry McKnight. Expeditionary forces are America's military crown jewel. *Proceedings*, 145(1/1,391), January 2019.

David P Morton, Richard E Rosenthal, and Captain Lim Teo Weng. Optimization modeling for airlift mobility. *Military Operations Research*, pages 49–67, 1996.

Whitney K Newey, Fushing Hsieh, and James Robins. Undersmoothing and bias corrected functional estimation. 1998.

Thanh Nguyen-Tang, Sunil Gupta, A Tuan Nguyen, and Svetha Venkatesh. Offline neural contextual bandits: Pessimism, optimization and generalization. *arXiv preprint arXiv:2111.13807*, 2021.

Xinkun Nie and Stefan Wager. Quasi-oracle estimation of heterogeneous treatment effects. *Biometrika*, 108(2):299–319, 2021.

Stephen M Omohundro. Five balltree construction algorithms. 1989.

Zeynep Özyurt and Deniz Aksen. Solving the multi-depot location-routing problem with lagrangian relaxation. In *Extending the horizons: Advances in computing, optimization, and decision technologies*, pages 125–144. Springer, 2007.

Hae-Sang Park and Chi-Hyuck Jun. A simple and fast algorithm for k-medoids clustering. *Expert systems with applications*, 36(2):3336–3341, 2009.

M.B. Pedersen, T.G. Crainic, and O.B.G. Madsen. Models and tabu search metaheuristics for service network design with asset-balance requirements. *Trans-*

portation Science, 43(2):158–177, 2009. doi: 10.1287/trsc.1080.0234. URL https://doi.org/10.1287/trsc.1080.0234.

Serge A Plotkin, David B Shmoys, and Éva Tardos. Fast approximation algorithms for fractional packing and covering problems. *Mathematics of Operations Research*, 20(2):257–301, 1995.

Donato Sherwin Powell. *An optimization model for sea-based logistics supply system for the Navy and Marine Corps*. PhD thesis, This publication is a work of the US Government as defined in Title 17 ..., 2004.

W.B. Powell. A local improvement heuristic for the design of less-than-truckload motor carrier networks. *Transportation science*, 20(4):246–257, 1986. ISSN 0041-1655.

W.B. Powell and I.A. Koskosidis. Shipment routing algorithms with tree constraints. *Transportation Science*, 26(3):230–245, 1992. doi: 10.1287/trsc.26.3.230. URL https://doi.org/10.1287/trsc.26.3.230.

W.B. Powell and Y. Sheffi. The load planning problem of motor carriers: Problem description and a proposed solution approach. *Transportation research. Part A: general*, 17(6):471–480, 1983. ISSN 0191-2607.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*, 2023.

Omid Rafieian and Hema Yoganarasimhan. Targeting and privacy in mobile advertising. *Marketing Science*, 40(2):193–218, 2021.

Krishan Rana and RG Vickson. Routing container ships using lagrangean relaxation and decomposition. *Transportation Science*, 25(3):201–214, 1991.

Anders Rantzer. Dynamic dual decomposition for distributed control. In *2009 American control conference*, pages 884–888. IEEE, 2009.

J Reith, Jennifer Van Drew, and Teresa Hines. Operational logistics planner for a leaner, more capable expeditionary army. *US Army (November 1), http://www. army. mil/article/176706*, 2016.

Matthew B Rogers, Brandon M McConnell, Thom J Hodgson, Michael G Kay, Russell E King, Greg Parlier, and Kristin Thoney-Barletta. A military logistics network planning system. *Military Operations Research*, 23(4):5–24, 2018.

Paul R Rosenbaum and Donald B Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.

Conrad W Rosenbrock, Björn Bieniek, and Volker Blum. Hands-on tutorial on cluster expansion ipam los angeles, california, july 2014.

Richard E Rosenthal, Steven F Baker, Lim Teo Weng, David F Fuller, David Goggins, Ayhan O Toy, Yasin Turker, David Horton, Daniel Briand, and David P Morton. Application and extension of the thruput ii optimization model for airlift mobility. *Military Operations Research*, pages 55–74, 1997.

Otmane Sakhi. *Offline Contextual Bandit: Theory and Large Scale Applications*. PhD thesis, Institut Polytechnique de Paris, 2023.

Otmane Sakhi, Pierre Alquier, and Nicolas Chopin. Pac-bayesian offline contextual bandits with guarantees. In *International Conference on Machine Learning*, pages 29777–29799. PMLR, 2023.

Otmane Sakhi, Imad Aouali, Pierre Alquier, and Nicolas Chopin. Logarithmic smoothing for pessimistic off-policy evaluation, selection and learning. *arXiv preprint arXiv:2405.14335*, 2024.

Y.O. Scherr, B.A. Neumann Saavedra, M. Hewitt, and D.C. Mattfeld. Service network design with mixed autonomous fleets. *Transportation Research Part E: Logistics and Transportation Review*, 124:40–55, 2019.

Y.O. Scherr, M. Hewitt, B.A. Neumann-Saavedra, and D.C. Mattfeld. Dynamic discretization discovery for the service network design problem with mixed autonomous fleets. *Transportation Research Part B: Methodological*, 141:164–195, 2020. ISSN 0191-2615. doi: https://doi.org/10.1016/j.trb.2020.09.009. URL `https://www.sciencedirect.com/science/article/pii/S0191261520304021`.

Kristof T Schütt, Huziel E Sauceda, P-J Kindermans, Alexandre Tkatchenko, and K-R Müller. Schnet–a deep learning architecture for molecules and materials. *The Journal of Chemical Physics*, 148(24), 2018.

Blake Schwartz, Brandon McConnell, and Greg H. Parlier. How data analytics will improve logistics planning, November 2019. The United States Army, `https://www.army.mil/article/223842/how_data_analytics_will_improve_logistics_planning`.

Alexander Semenov, Maciej Rysz, Gaurav Pandey, and Guanglin Xu. Diversity in news recommendations using contextual bandits. *Expert Systems with Applications*, 195:116478, 2022.

Uri Shalit, Fredrik D Johansson, and David Sontag. Estimating individual treatment effect: generalization bounds and algorithms. In *International conference on machine learning*, pages 3076–3085. PMLR, 2017.

Cosma Rohilla Shalizi. k-Nearest Neighbors I (Mostly Theory) (lecture 9). Lecture notes for Statistics 36-462/662: Data Mining, Fall 2019, Carnegie Mellon Uni-

versity, September 2019. Online at: https://www.stat.cmu.edu/~cshalizi/dm/19/lectures/09/lecture-09.html.

Nian Si, Fan Zhang, Zhengyuan Zhou, and Jose Blanchet. Distributionally robust batch contextual bandits. *Management Science*, 69(10):5772–5793, 2023.

Justin S Smith, Olexandr Isayev, and Adrian E Roitberg. Ani-1: an extensible neural network potential with dft accuracy at force field computational cost. *Chemical science*, 8(4):3192–3203, 2017.

Alex Strehl, John Langford, Lihong Li, and Sham M Kakade. Learning from logged implicit exploration data. *Advances in neural information processing systems*, 23, 2010.

Yi Su, Maria Dimakopoulou, Akshay Krishnamurthy, and Miroslav Dudík. Doubly robust off-policy evaluation with shrinkage. In *International Conference on Machine Learning*, pages 9167–9176. PMLR, 2020.

Adith Swaminathan and Thorsten Joachims. Batch learning from logged bandit feedback through counterfactual risk minimization. *The Journal of Machine Learning Research*, 16(1):1731–1755, 2015a.

Adith Swaminathan and Thorsten Joachims. Counterfactual risk minimization: Learning from logged bandit feedback. In *International conference on machine learning*, pages 814–823. PMLR, 2015b.

Adith Swaminathan and Thorsten Joachims. The self-normalized estimator for counterfactual learning. *advances in neural information processing systems*, 28, 2015c.

Håkan Terelius, Ufuk Topcu, and Richard M Murray. Decentralized multi-agent

optimization via dual decomposition. *IFAC proceedings volumes*, 44(1):11245–11251, 2011.

Huseyin Topaloglu. Using lagrangian relaxation to compute capacity-dependent bid prices in network revenue management. *Operations Research*, 57(3):637–649, 2009.

Lorenzo Torresani, Vladimir Kolmogorov, and Carsten Rother. A dual decomposition approach to feature correspondence. *IEEE transactions on pattern analysis and machine intelligence*, 35(2):259–271, 2012.

United States Marine Corps. Mstp pamphlet 4-0.2 logistics planner's guide. https://www.tecom.marines.mil/Portals/90/Docs/MSTP/MSTP%20Pamphlet%204-0_2%20Logistics%20Planner%27s%20Guide%20February%202022.pdf?ver=ah3X-lpI_q4mR0i89gJPCA%3D%3D, February 2022. Accessed: 2024-07-29.

United States Marine Corps. Tentative manual for expeditionary advanced base operations, 2nd edition. https://www.marines.mil/Portals/1/Docs/230509-Tentative-Manual-For-Expeditionary-Advanced-Base-Operations-2nd-Edition.pdf, May 2023. Accessed: 2024-07-29.

Lieven Vandenberghe. Lecture notes in optimization methods for large-scale systems, Spring 2022.

Fugao Wang and David P Landau. Efficient, multiple-range random walk algorithm to calculate the density of states. *Physical review letters*, 86(10):2050, 2001.

Lequn Wang, Akshay Krishnamurthy, and Alex Slivkins. Oracle-efficient pessimism: Offline policy optimization in contextual bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 766–774. PMLR, 2024.

Z. Wang and M. Qi. Robust service network design under demand uncertainty. *Transportation Science*, 54(3):676–689, 2020.

N. Wieberneit. Service network design for freight transportation: a review. *OR Spectrum*, 30(1):77–112, 2008. ISSN 0171-6468.

Bryan Wilder, Bistra Dilkina, and Milind Tambe. Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1658–1665, 2019.

Blair S. Williams. Heuristics and biases in military decision making. *Military Review*, 2010.

David P. Williamson. Lecture 8: Strong duality. `https://people.orie.cornell.edu/dpw/orie6300/Lectures/lec08.pdf`, September 2014. ORIE 6300 Mathematical Programming I, Cornell University.

Ping Wu, Gunnar Eriksson, Arthur D Pelton, and Milton Blander. Prediction of the thermodynamic properties and phase diagrams of silicate systems–evaluation of the feo-mgo-sio2 system. *ISIJ international*, 33(1):26–35, 1993.

Lin Xiao, Mikael Johansson, and Stephen P Boyd. Simultaneous routing and resource allocation via dual decomposition. *IEEE Transactions on Communications*, 52(7):1136–1144, 2004.

Zhouhao Yang, Yihong Guo, Pan Xu, Anqi Liu, and Animashree Anandkumar. Distributionally robust policy gradient for offline contextual bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 6443–6462. PMLR, 2023.

Baqun Zhang, Anastasios A Tsiatis, Marie Davidian, Min Zhang, and Eric Laber. Estimating optimal treatment regimes from a classification perspective. *Stat*, 1 (1):103–114, 2012.

Puning Zhao and Lifeng Lai. Minimax optimal estimation of kl divergence for continuous distributions. *IEEE Transactions on Information Theory*, 66(12): 7787–7811, 2020.

Puning Zhao and Lifeng Lai. Analysis of knn density estimation. *IEEE Transactions on Information Theory*, 68(12):7971–7995, 2022.

Yingqi Zhao, Donglin Zeng, A John Rush, and Michael R Kosorok. Estimating individualized treatment rules using outcome weighted learning. *Journal of the American Statistical Association*, 107(499):1106–1118, 2012.

Xin Zhou, Nicole Mayer-Hamblett, Umer Khan, and Michael R Kosorok. Residual weighted learning for estimating individualized treatment rules. *Journal of the American Statistical Association*, 112(517):169–187, 2017.

Zhengyuan Zhou, Susan Athey, and Stefan Wager. Offline multi-action policy learning: Generalization and optimization. *Operations Research*, 71(1):148–183, 2023.

E. Zhu, T.G. Crainic, and M. Gendreau. Scheduled service network design for freight rail transportation. *Operations Research*, 62(2):383–400, 2014. doi: 10.1287/opre.2013.1254. URL `https://doi.org/10.1287/opre.2013.1254`.