(1) Online selection problems = items arrive sequentially, have to be accepted/rejected @ arrival, partial info on their val.

More specific than just "online problems"
=> Make this passage more precise.

(2) Challenge = incomplete info
Consequence = suboptimal decisions

Quotation marks = `` + '' (2x back quote, 1x double quote)

Chapter 1

Introduction

optimization

(1) Online ~~selection~~ problems are a class of ~~decision~~ optimization problems centred around designing algorithms that make ~~'nearly' optimal~~ decisions in a~~n online~~ setting, where decisions need to be made in real-time without complete information on future events. In this setting, an algorithm typically needs to make a series of irrevocable decisions, subject to a set of constraints, as it receives information ~~on~~ about the problem incrementally. The challenge being that choices we make in the present can restrict options available to us in the future, this generally makes optimal decision making impossible. The study of these problems has applications in areas where we need to be able to adapt to changing conditions and have only knowledge of the situation, ~~examples include~~ for example stock trading, resource allocation, routing in computer networks, and online advertising. Let us present two motivational examples in more detail. ← Mb. remove sentence

repetition

about

wordy sentence + avoid "we"?

(2)

not a selection problem

A classical example of an online problem is the ski rental prob., where ...

In the **ski rental problem** [1], we would like to go skiing with a group of friends and are faced with a choice between buying or renting a pair of skis. The local ski-rental shop charges a (fixed) daily rate ~~R~~. Buying, on the other hand, incurs a one-time cost ~~$C > R$~~ and allows us to ski without further expenses for the rest of the season. Unfortunately, due to the ~~unpredictable~~ weather in the mountains, we ~~have no idea of knowing~~ do not know how often we will be able to go skiing ~~with our friends~~. We end up going just the one time or every day of the season. Hoping to avoid unnecessary expenses, how do we decide what to do? Answer?

On the one hand, on the other hand

avoid "we"?

paragraph feels unnecessarily wordy

avoid "we"?

Another, more recent example is

In the so-called **adwords problem** [2], we want to help a search engine company maximise its total ad revenue, the revenue from showing targeted ads to users. Advertisers provide us with a set of keywords (adwords) they would like to advertise on and a quote of their daily advertising budget. They agree to pay us a fixed rate every time we show an ad to a user who searched for one of these keywords. We cannot, however, charge the advertisers more than

the ad is to maximize the revenue from displaying ads related to keywords in search queries. More specifically, ...

Not sure if detail about ski rental and adwords add much, but OK.

Adwords is not classical => "another classical" is slightly inaccurate

"In this thesis, we study problem settings related to ..."

or "focus on" also on

remove par. break

their daily budgets specify.
As users enter queries into the search engine, we have to decide which ads to show them in real-time. How can we do this in a way that maximises ad revenue while respecting the daily budgets of advertisers?

m.b. just [3]

which

Another classical example of an online selection problem that we will be focusing on is the so-called **secretary problem**. ~~it~~ was introduced and studied by Dynkin in 1963 [3]. Here, we are interested in hiring the most qualified secretary from a pool of n applicants. We assume all applicants can be ranked unambiguously by their 'level of qualification' which is unknown to us a priori. The applicants are interviewed in uniform random order. As they are interviewed, their 'level of qualification' is revealed to us (and we can compare it to the qualifications of previously interviewed applicants). At this point, we have to make an irrevocable decision on hiring or rejecting the applicant. / remove Dykin found an algorithm that selects the best applicant with probability at least $1/e$, this success rate turns out to be asymptotically optimal. In ~~an~~ the initial phase, the algorithm interviews ~~or samples~~ the first k applicants and rejects them all. In ~~a~~ the second phase, the algorithm ~~selects~~ (hires) the first applicant whose 'level of qualification' exceeds that of the previous applicants. ~~The optimal sample size k can be calculated explicitly for small instances and converges to n/e as $n \to \infty$.~~

m.b. value?

whenever we interview a candidate, we learn

and break / par.

to hire or reject

of all the previously seen

write $k = \lfloor n/e \rfloor$ above

why mention this? overcomplication at this stage

In his paper, Dynkin described an optimal alg...

More recently, various generalisations of the secretary problem have been studied, where the algorithm is allowed to to select a set of multiple elements subject to some combinatorial constraints. Motivated by applications in online mechanism design, Babaioff et al ~~2007~~ introduced the class of so-called matroid secretary problems [4]. Mechanism design is a field in economics focused on designing mechanisms to allocate items to (bidding) agents in a way that maximises total revenue or social welfare - the study mechanism design problems on matroid domains is of particular economic interest.

defining clause + awkward position of "where"

??? expected more details on MSP, not on online mech. design.

+ reference to a paper on o.m.d.?

/ remove par. break / ⊛ The goal is to

In this problem setting, there is a weighted matroid $M = (E, \mathcal{I})$, whose ground set E is of known size n. And we are interested in designing an algorithm that selects an independent set $P \in \mathcal{I}$ of approximately maximal weight. The algorithm has to select from a sequence of elements $e \in E$ arriving one by one in uniform random order. As elements arrive, they reveal their (unique, positive) weight to the algorithm and an irrevocable decision on whether or not to include them in the selection P has to be made. To ensure the inclusion of an element does not violate the independence constraint on P, the algorithm can access an independence oracle in every step. /remove par break/
We ~~call~~ denote the offline maximum weight independent ~~the~~ set as OPT and we call an algorithm with the property that the expected weight of the selection, $w(P) =$

as large weight as possible

not a sequence

requires elaboration, so mention separately in prelim. section

trivially wlog, so remove details, depends on precise setting

rephrase

[⊛ "there is" — where? how given? raises too many questions. Better: "in [MSP], elements of [M] are revealed..."]

the "we" used too often => tone becomes too informal.
vary by using different subjects (preferred) or passive voice (OK if not too frequent)

n.b. try "the performance of the algorithm is measured by the competitive ratio", "an alg. is called c-competitive if", etc.

Better paragraph break: setting vs performance?

informal $c \cdot w(P) \geq w(\mathrm{OPT})$ call x2 extremely informal

$\sum_{e \in P} w(e)$, is within a ~~factor~~ c of $w(\mathrm{OPT})$ is called c-competitive. Note: The algorithm for the classical secretary problem outlined above is e-competitive.

for instance, ... also n.b. unnecessary

Dynkin's

conjectured

algorithms/procedures

Due to certain nice properties of matroid domains such as the fact that we can find optimal solutions greedily offline, Babaioff et al. formulated a conjecture on the existence of $O(1)$-~~competitive methods~~[1] for general matroid~~s domains~~. They also provided ~~an initial result of~~ a $O(\log(\rho))$-competitive algorithm ~~for matroid secretary on general matroid domains,~~ where ρ is the rank of the underlying matroid. [This result has seen several improvements, Chakraborty and Lachish in 2012 [5] and Lachish in 2014 [6] provided methods with stronger competitive ratios. The current best competitive ratio is obtained by a $O(\log\log(\rho))$-competitive algorithm presented by Feldman et al. in 2014 [7], their method is simpler and has better constants than the $O(\log\log(\rho))$-competitive method by Lachish.] /remove par break/

a general $O(\log \rho)$-comp. alg.

be more precise about contributions

Lachish and Feldman et al. both have current best $O(\log\log \text{rank})$

Also mention specific matroid classes?

The existence of a constant-competitive algorithm for general matroids remains an open question.

Nevertheless, constant CRs have been achieved for some [cases]. One such variation is

For some special cases and variations of the matroid secretary model, however, constant-competitive algorithms have been found. In the free order model, a setting where the structure of the matroid is known beforehand and the arrival order is determined by the algorithm. Jaillet et al. ~~were able~~ designed a constant-competitive method in 2013 [8] [2]. [Specifically, they proved their method includes every element in OPT with probability at least 1/4, this lower bound is tight in some cases [3].] Note their algorithm does not match the competitive ratio $1/e$ of Dynkins algorithm for classical secretary, this suggests the possibility of further improvements.

upfront

that

be more precise

the algorithm determines which element is revealed next at each step.

alg./procedure

n.b. "one can observe that many of the aforementioned results ..."

Most of the existing work on secretary problems is based on combinatorial and linear programming approaches. Moreover, many previous methods are static in the sense that their behaviour depends only to a very limited degree on newly revealed information. This somewhat limited scope encouraged us to look at matroid secretary~~, in particular the classical secretary problem and the free order model~~, from a new perspective. More specifically, we wanted to use tools from information theory such as the concept of **Shannon entropy** to develop novel and dynamic approaches. In information theory, Shannon entropy is used as a measure of the amount of information contained in random systems.

add citations

add citations

wdym? be more precise

separate citations by approach

save for later

the main goal of this thesis is

+ motivate why use entropy (connect to ☐)

include in text?

[1] A stronger version of the conjecture claims there exist e-competitive methods for general matroid domains, i.e. methods matching the performance of Dynkins algorithm for the classical secretary problem.

[2] The algorithm is included in the appendix.

[3] We refer to the appendix for a proof.

△ Proposal: Jaillet et al [8] gave a $\frac{1}{4}$-comp. alg. for this setting, which was [motivation/value of result]. However, this alg. is not optimal, since there is a trivial lower bound on the CR of $1/e$ from the classical secretary case and there are instances on which Jaillet et al.'s alg. has CR of exactly $\frac{1}{4}$ (as we show in the appendix). ← (m.b. switch order to $\frac{1}{4}$ instances, then $\frac{1}{e}$ LB)

① feels disrespectful and vague.
better: motivate via intuition of "learning" and "information", m.b. cite mentions of such words in literature

Let us elaborate on our plans for classical secretary problem and the free order model, respectively.

this sounds conversational

The optimal algorithm for classical secretary problem, as outlined above, relies on an initial sampling phase to make hiring decisions later on. This initial phase could, alternatively, be described as an information gathering phase, as it is precisely the knowledge of the applicants in the sample that is used to make a hiring decision. We were interesting in formalising this notion, that is, we wanted to find an alternate derivation of the optimal sample size by finding an optimal-information threshold to start the hiring process.
For the free order model we wanted to analyse the performance of algorithms that dynamically reveal elements depending on the information gained from previously revealed elements. For the sake of simplicity, we restricted ourselves to the analysis of a special class of graphical matroids, so-called hat graphs.

Our contributions can be summarised as follows. In the first chapter we present our work on the classical secretary problem. Though we were unable to find an explicit connection between the optimal sample size and information contained in the system, we are able present several entropy-calculations that allow us to measure the relevant information over time that is contained in the setting and present a number of observations.
In the second chapter we work on the free order model. Specifically we analyse the performance of an entropy-inspired, dynamic algorithm on hat graphs. Due to computational complexity and correspondingly difficult analysis we resorted to numerical tests and present a conjecture on the behaviour of the algorithm.

The particular goals we set out to achieve in this thesis were as follows. The first aim was to recontextualize the analysis of the classical secretary problem in terms of entropy. More specifically, [measure info gathered during sampling phase to determine optimal sampling threshold (i.e., <brief definition>)].
Our second aim was to design an alg. for the free order model with better CR than $\frac{1}{4}$ using entropy. In particular, [reuse ideas from class. sec. for better selection rule; moreover, use entropy to "better" choose arrival order].

M.6. merge w/ previous paragraph?
Be more precise when stating results
First list exact results (verbs: computed, proposed alg., implemented, measured performance, etc.),
only then give opinion (e.g., "unable to find explicit connection"; also mention upsides)

CSP:
relevant info content,
have conjecture(s)
about how to utilize (in analysis)

FO-MSP:
numerics show
improved CR
on small examples

Overall: * quite decently written, clear structure
* tone a bit too informal for a thesis.
OK to be down-to-earth, but try to
a) make it sound less like a conversation w/ a friend
b) be more precise w/ your statements/claims
* weird paragraph breaks — reassess topics
* missing key motivation behind project — see 1
* need to make one more quick pass (1 hr max), proofread along the way (definite/indefinite articles, defining/non-defining clauses, missing words, signposting words)