I'm ready to go

我准备好了

so it's fine

好的

I'm very happy to see so many of you here today to listen to professor for six days course on the posh man and an improvement

我很高兴看到很多人来听教授六天的课程

so before professor the fact that starts at what cost many briefly introduced so Dimitri protectors studied mechanical and electrical engineering at the National Technical University of a -piece and obtained his PhD in citizen science from Massachusetts Institute of Technology

在教授开始之前我要简单介绍一下，Dimitri教授在

he has held faculty positions mystic engineer economics district department name standard and then the electrical engineering department of the University of Illinois in Urbana since he has been teaching at the electrical engineering and computer science department of the MIT

where he is recovering in McAfee professor of engineering his research spans several years including commendation control large-scale computation and data communication networks and is close offering activities we have written numerous research papers and quality assurance they say books right now several of which are used and test those in my office missing all the expansion program started the business pieces and has continued through the present with many research papers and separate books and research monographs professor those habits was awarded with impulse binding of prize for research excellence in interface between operations research and computer science for his book nearly dynamic programming for all the way the job degrees it was awarded the national award for operations research then into a summit important American control conference jamun target for the leading educational work and the alive impulse expository writing network in he was in a kitchen a dining space [Music] [Applause]

BERTSEKAS

thank you very much Sam

非常感谢你，Sam

well it's my pleasure to be here

非常荣幸能到这里

I'm concerned a little bit with sound can you hear me in the back now

我有一点担心后面的同学能不能听到我的声音

okay we're trying to get a microphone okay

我把麦克风带上

so I'll try to do my best to the sound

我会尽量让声音效果最好

and this is six lecture course on approximate dynamic program so somewhat advanced course it's a research oriented course the subject right at the forefront of research in Tzfat subject with many applications and interesting theory

这是一个关于近似动态规划的六次课程，这是一门面向研究的课程，所以他比较前沿，前沿的研究包括很多应用与有趣的理论

there are many books now in on the subject however my lectures are going to be based

这门课程的主题依赖于很多本书

not surprisingly on my books

毫无意外地，都是我的书

the first book is the hero dynamic programming book that I've written will charge the checklist back in so I research monograph really and many parts of it still remain relevant now

第一本书是神经动态规划，,这本专著现在还是和我的很多研究相关

I have a textbook who the latest edition was published two years ago and this is more up to date and it's also more accessible than the first book in last year

我有一本书两年前出版了第二版，这本书比较新，也比上一本书容易获得

I wrote a research monograph called abstract dynamic programming that has some relevance to the course material we are going to be posting the slides my slides in a website which will be announced

这本书叫做abstract dynamic programming（抽象动态规划？），我会把slide发布到一个网页上

however if you want to see a full set of slides on dynamic programming including approximate dynamic programming you may find it here there are several sets of slides the latest being one or two years ago okay

如果你希望获得更多关于动态规划或者近似动态规划的slide，你可以在这个网页找到最近一两年的slide

BRIEF OUTLINE I

so let me give you a brief outline of what we're going to be covering

我现在要给你们看一个我们将要讲的内容的目录

okay this is our subject large-scale dynamic programming based on approximations and in part on simulation a research area of great interest for the last years

我们的主题是，基于近似（一部分是基于仿真）的大规模动态规划，这是一个最近几年受到很大关注的研究领域

and it comes under various names

这个领域有很多名字

reinforcement learning that's the that's the name that people in artificial intelligence or learning use

人工智能或者机器学习的研究者叫它强化学习

and neuro dynamic programming or approximate dynamic programming that I have been using

我一般叫他神经动态规划或者近似动态规划

they have been two lines of research which at some point converged and predicted in a very productive and fruitful way

目前有两个研究方向，在某种程度上可以用一种很有成效的方式融合与预测到

one and was from artificial intelligence the reinforcement learning line with its ideas on on feature based representations of functions and relations and also the idea of learning by observation or simulation

一个方向是人工智能，也就是强化学习，他主要依赖于基于特征表达的函数与输入输出之间的关系，或者可以叫通过观察样本或者仿真进行学习

and the other side of the the other line of research came from optimization control theory

另一个研究方向是最优控制理论

with his emphasis on formal optimization methods and algorithms such as policy duration value duration and so on

它的关注重点是传统的优化方法和算法，比如策略迭代和值迭代什么的

at some point people that worked over here people who worked over here realized that they were doing with the same problem approaching from different sides and the cross fertilization between them was very productive

这两个方向的人认识到他们正在从两个不同的方面出发解决同一个问题，这获得了很大的成果

now our subject deals with control of dynamic systems under uncertainty

我们的主题是在不确定性下的控制问题的动态规划

okay if you could okay we are going to be discussing control of dynamic systems under stochastic uncertainty

我们会讨论随机或者不确定性控制问题下的动态规划控制

but the material we are going to talk about has extension

我们将要讨论的话题被延伸了

some of which I'm going to mention the cases where the uncertainty is not stochastic it may be minimax right

我将会提到的不确定性，可能并不是指随机，有可能是一种使用minmax解决的问题

one certainty set membership uncertainty or or uncertainty produced by the actions of an antagonistic opponent

这种不确定性，有时候来源于对手做出的行为

as in games multistage games but also it has it because dynamic program is a very broadly applicable subject

比如游戏或者多阶段游戏，我们在这种问题中讨论动态规划是因为动态规划是一种应用非常广泛的方法

it has applications way B here on the control of dynamic systems such as for example solving discrete optimization problems combinatorial optimization problems in the G program it's an excellent support method for solving such problems

动态规划可能被用于求解动态系统的控制问题，比如离散优化问题，组合最优化问题这是一种特别适合解决这些问题的方法（动态规划）

dynamic programming has always been recognized as having a very very broad range of applications it was that the algorithms were not sufficiently powerful to deal with the applications in the size of the applications

动态规划被认为是一种应用非常广泛的算法，但是他并不能很有效地解决实际应用的规模下的问题

but now that we know how to deal with large-scale problems there's a vast horizon for applications in all kinds of fields

但是现在我们已经知道如何解决所有场景下的展望期非常长的大规模的问题了

from control theory to operations research to economics to attrition intelligence finance and all kinds of fields

比如控制理论，运筹优化，经济问题，人工智能，金融等

dynamic program is very general

动态规划是一种非常一般性的方法

and now that we are getting close to being able to realize its full potential

现在我们就要知道他的潜力了

we have a big Vista of potential payoff regarding your methodology of the subject

关于你的方法，我们有很大前景的潜在汇报

it's it has a rich variety of theory in math it also has an element of art in it

动态规划不仅在数学上有很多理论支撑，还具有艺术性

because we are dealing with challenging problems

我们在解决很有挑战的问题

it's important to approach them in a creative way there is also there are also issues relating to more abstract theoretical aspects of the subject modeling issues

如何以有创造性的方法接近这些问题的解决方法很重要，同时还有很多抽象的理论需要讨论，比如如何建模，

how do you model problems dynamic programming problems

你该如何建立一个动态规划的模型

we don't have time to cover all this so we will focus primarily on algorithms okay

我们没有时间来涵盖所有这一切, 所以我们将重点主要在算法

BRIEF OUTLINE II

so here's what we will end with ever limited the limited framework of six lectures a state-of-the-art account of some of the major topics at the graduate level

我们会以这六次有限的课程为基础，对研究生阶段的重要的课题的最前沿的工作进行介绍

okay this is fairly advanced is for graduate students primarily

这些课程对于研究生来说很好

okay it's a it's more over it's fast so if you could make an effort to study this type the slides ahead of time read a little bit it would be good

如果你们能提前学习一下这些slide就更好了

because we are going to be going somewhat fast and our aim is to show how by approximation and simulation

因为我们会讲的很快，授课的目标是展示如何使用近似或者仿真技巧求解问题

we can address the dual process of dynamic programming

我们会解决动态规划存在的两个问题

now bellman the originator of dynamic programming coined the term curse of dimensionality

提出动态规划的学者-bellman 提出了“维数灾”的概念

meaning the difficulty of addressing large dimensional problems with the dynamic programming technique

意思是使用动态规划求解大规模问题的时候很困难

he recognized this as a major limitation of dynamic programming

他意识到这是动态规划的主要局限性

well we are going to try to address this curse of dynamic programming there's another curse

of dynamic programming

现在我们要提提到另一个动态规划的灾难

if you want to apply it in its exact form you need an exact mathematical model of the system

如果你想要把动态规划应用在一个具体的问题上，你需要这个系统的数学模型

you need equations for the system transition probabilities and so on

你需要这个系统的状态转移方程之类的东西

some systems do not have an easily obtainable mathematical model

一些系统不能容易地获得数学模型

but instead you can simulate them with a simulator

但是你可以通过仿真来代替数学模型

I have a real time simulator or computer simulator and it turns out that we can deal with problems like that without a mathematical model

我有一个实时的仿真系统，它能够证明我们可以不用数学模型解决问题

a computer simulator will suffice and that's part and parcel of the nature of the techniques were going to gasps this here's what we're gonna cover two lectures today

一个计算机模拟器就足够代替数学模型, 这是动态规划技术的一部分，我们今天的课程就会包括这些内容

and Wednesday on exact diagnostic programming a review or finite horizon problems

周三要讲的内容是有穷维问题的精确动态规划综述

but with an emphasis on infinite horizon problems control of a dynamic system over an infinite number of stages

但是无穷维动态系统，也就是无穷维控制问题才是重点

and we are going to talk about algorithms and issues of large-scale computation with election on Friday

我们周五会讨论一些算法和关于大规模问题计算的问题

we are going to enter the approximation in simulation methodology

那时我们会在仿真技术中引入近似

we're going to discuss general issues for large-scale problems

我们会讨论一般性的大规模问题

and then in the second week we are going to discuss more specific approximate dynamic programming techniques

下周会讨论更多特殊的动态规划技术

one lecture an approximate policy Direction based based on methods called temporal differences

一种基于策略方向的方法，时域差分

galerkin approximation various methods of this type one election aggregation methods in one lecture and q-learning and other methods such as approximation policy space

用一次课的时间介绍时域差分的不同算法，比如q-learning和策略空间近似等方法

so the first week is going to be preparation for the second week

所以其实第一周是为了第二周的课程做准备

that doesn't mean that the first week is going to be easy okay

但这并不意味着第一周的课程很简单

just to warn you

只是提醒你们一下

but the main research content is going to come here in the second week

主要的研究内容会在第二周介绍

LECTURE OUTLINE

okay so what are we going to do today an introduction to dynamic programming and approximate dynamic program

今天我们要做的事情时介绍动态规划和近似动态规划

first focus on finite horizon problems involving decisions in a finite number of steps okay

首先我们会关注有穷维问题

make a decision now then make a decision tomorrow then again a finite number of decisions in sequence

现在决策一次，明天决策一次，然后不停地决策有限次形成一个决策序列

we're going to discuss the dynamic programming algorithm for such finite horizon problems the classical form of the algorithm

我们接下来要的就是讨论经典的解决有穷维问题的动态规划算法

then we are going to turn to infinite horizon problems

然后把这些算法扩展到无穷维问题上去

and develop some of the basic theory for the easiest type of infinite horizon problem which is discounted discounted problems this kind of problems with bounded cost per stage okay

并且开发一些能解决这些无穷维问题，比如阶段平均成本有界的离散问题的基础理论

DP AS AN OPTIMIZATION METHODOLOGY

so let's get started first of all where does dynamic programming fit within the broad field of optimization

现在我们就开始讲能够适用于很多领域的优化问题的动态规划方法

when I started doing research in this field optimization was really an exotic subject

当我开始进行优化领域的研究的时候，发现这是一个非常奇特的主题

you know something that was at the edges of research now it has become very fundamental all encompassing all pervasive you can find optimization elsewhere everywhere and this and this there are so many different kinds of optimization that one wants what the what are the connections between all these and a more abstract level there is only one optimization problem what you see here minimize a cost function of a barrier would you you may be a scaler a vector or something even more complicated

你知道的，这是一个非常基本的研究，涵盖了很多领域，你可以从任何地方发现很多不同形式的优化问题，从更抽象的角度来看，只有一种优化问题，即最小化成本函数，或者叫最小化问题，有时候可能要优化一个标量，有时候要优化一个向量，或者更复杂的问题。

it must live within a constrained set capital u and you want to minimize this cost function overall u that are admissible are in this constraint set

优化时必须受到约束集合u的限制，你想要优化成本函数时，u从所有可行的约束集合中选，并找到令目标函数最小的u

now depending on the nature of g(u) you have different names the big divide is between discrete problems will use a finite set or continuous problems where u is not finite and g as some continuity prop

想要优化的目标函数g(u)有两个名字，最大的区别是离散问题会从一个有限集中寻找最优的u还是从连续的值中找最优的u，u是连续值时，u可以取的值的个数是无穷多个，

this is differentiable for example and so on discrete problems are the combinatorial problems

离散问题又叫组合问题，它和连续问题是不一样的

the computer scientists like to address the integer programming problems they have a different character than this continuous problems which are usually solved by calculus type of methods or convex in type of methods

计算机科学家喜欢解决整数规划问题，整数规划问题有与连续问题不同的特点（通常使用微积分类型的方法活凸优化相关的方法）

so that's one big divide discrete and continues

这就是离散问题与连续问题的一个很大的不同

another big divide is linear programming problems and non linear programming problems

另一个比较大的不同的分类方法是线性规划问题与非线性规划问题

linear programming problems are the ones linear

线性规划问题指的是线性的问题（目标函数与约束都是线性的）

nonlinear are continuous but in linear g is linear and use a polyhedral set

非线性是连续问题，连续问题指的是目标函数g是线性函数，同时定义在多边形集上

and non when they are the ones that are not linear

非线性问题指的是那些东西(目标函数与决策变量) 是非线性的情况

linear are connected to discrete and integer programming problems

线性问题与离散问题活整数规划问题相联系

but fundamentally they are continuous so that's another categorization of problems another major categorization of problems is to divide them between stochastic and deterministic

另一个基本的分类方法也是主要的分类方法是把问题分为随机问题（其实这里叫不确定性问题比较准确）与确定性问题

now stochastic problems involve a stochastic parameter W a random variable which is averaged in some way by taking expectation

随机问题包括一个随机参数w，一个以某种方式在一定程度上获得了预期的平均值(或者也可叫期望)

and g the cost function has this form here so for a given u I have a function capital G of U in some random variable W

成本函数的形式是：给定了u与随机变量w，有一个函数G(u)

I average this by taking the expected value over W

我通过w的期望计算G的平均值

and for each u I get a number here okay

对于每一个给定的u，都能计算得到一个g的值

and now I minimize this number over u so stochastic problems are really the same as these

然后我在所有给定的u内最小化目标函数，所以随机问题其实和确定性问题没什么区别

but they involve a stochastic element which very much affects the type of methodology that you need to solve them

但是随机问题包括一个随机元素，这会对你选择想要解决的问题的方法时造成很大的影响

now dynamic programming in tandem with stochastic problems as well as deterministic problems shall deal with the street can deal with continuous candy was linear and nonlinear

现在我们有动态规划，它可以同时处理随机问题和确定性问题，线性和非线性问题

but the main characteristic that distinguishes from all the others is that it is multistage

但是这么多分类方法，最主要的把问题分开的方法，是按照是否多阶段区分的

in other words I select not just are u but are u over each one of several stages

换句话说，优化的时候，我选择的不是整个问题的u，而是每一个阶段的u

I select a u now I select a u tomorrow another u after two days and so on

我选择了今天的u，又选择了明天的u，两天以后的u，一直选择下去直到所有u都被选择

and each time this w this unknown random variable reveals itself to us in the form of some information

每一次选择u的时候，不知道的(准确地说应该是不确定，uncertainly)随机变量w都会告诉我们他的一部分信息（选择u的时候这个信息并不知道）

so multiple stages following receipt of new information okay

在很多个阶段之后，收到了新的信息

that's a characteristic of dynamic programming that's unique the information about W is rebuilding stages the decisions are also made in stages and they make use of the available information

这就是动态规划的特点，独特的随机信息w会在产生新阶段的时候生成，决策也会对生成新状态产生影响，即产生新阶段的时候会用到已知的信息

in other words we have to wait to get me information in order to make better decisions so there's a feedback loop here between the session and information which is characteristic to dynamic programming

换句话说，为了做出最好的决策，我们必须等到知道必须的信息，所以这是个会话和信息之间的反馈循环，这就是动态规划的特点

for this reasons methodology is different kind of different people say ask me often I have taken linear program should I take dynamic programming is it different what's fundamentally different it uses different methods different structure different mentality okay

因为这个原因，动态规划与其它方法是不同的，经常有不同的人问我，我使用整数规划求解一个问题，我能使用动态规划求解吗，他们的本质是不同的，动态规划使用了不同的方法，不同的结构和不同的求解思路

it's it's nothing very little that you see here in terms of methodology for these problems resembles what you see in dynamic programming

你在动态规划中看到的方法与其他方法是不同的

another big difference is that dynamic programming focuses on global minima okay

另一个很大的区别是动态规划关注的是获得全局最优解

global of globally optimal solution some of these methods may get stuck into local minima

一些方法难以获得全局最优解，只能获得局部最优解

so so if you've taken any one of these courses expect to hear something different here today in this in this course okay so if this is where it fits

如果你想听到一些不同的东西，那么今天的课程很适合你

BASIC STRUCTURE OF STOCHASTIC DP

and let me give you now a basic model that we are going to be using throughout this course

我会给你一个我们将会在这个课程中使用的基本动态规划模型

we have a discrete time dynamic system

现在有一个离散时间动态系统

a system that evolves over time K indexes the discrete time and K ranges from time to time n plus minus and terminates after X n is generated okay

离散时间系统包括一个时间k的索引，k的取值范围从0到n，终止状态在X\_n出现时产生

what is this X K XK is the state of the system what is the state

X\_k是系统状态

well it's hard to describe it but in the candy some very strange things but its main property is it summarizes all the past information that's relevant for future optimization

这很难描述，总之它主要的的特点是总结了过去所有的信息，并对未来进行优化

in other words the system evolves over time if we look at its current state in order to make future optimal decisions

换句话说，这个系统包括了时间，如果我们只看到了当前的系统状态，我们需要做出一个让未来也最优的决策

that's all I need to know just the current state in the data of the problem

这就是我需要知道的事情，这个问题当前的系统状态

I don't need to look at how the system came to this state

我不需要知道系统是如何达到当前系统状态的

I don't need to look at the past

我并不需要知道过去发生了什么给

it's something that mathematically speaking it separates the parts from the future from the purposes of optimization UK is the control or decision

从数学上讲，它把问题分成两部分，过去的信息和对未来的优化，U\_k表示控制或这决策

this is what we are interested in

这就是我们感兴趣的

we want to know how to make popped in all decisions while observing the current state

我们想要知道如何在观测到当前系统状态的时候从所有决策中选出一个决策

W K is a random parameter also can be viewed as a noise in some some engineering systems or we can call it a disturbance depending on the context

W\_k是一个随机参数，在一些工程系统中被称作噪声，我们也可以叫他依赖上下文的概率分布

and n is what we call horizon for the number of times control is applied

n是我们说的展望期，表示时间的数量，即需要决策的数量或者控制被执行的次数

later we're going to talk about infinite horizon problem where n is infinity

接下来我们会讨论无穷维问题，即时间维n是正无穷的问题

but let's focus first at the case of a finite horizon okay

但是我们首先要关注有穷维的问题

so we have a system starts at X then controls applied

所以我们有一个从系统状态x开始的系统，然后控制被执行

some random W occurs and x first then a new control is applied

一些随机变量w在产生新状态的时候出现，系统状态x更新，新的控制又一次被执行

uw occurs and u is applied and then we'll go to x and so on sequential application control it

控制u与随机变量w出现，u被执行，然后我们又观测到系统状态x就这样不断地进行下去，这样我们就可以使用一个决策序列来控制这个系统

sequential generation of states now with every transition with each J there's a certain cost that occurs at time K

每一次状态转移都会顺次产生新的系统状态，每一个时间k都可以观测到一个确定的成本J

the cost gk is the cost that is incurred when we are at state xk applied control UJ and also W occurs

我们在系统状态x\_k时采取控制u\_k，同时随机变量w\_k产生，他们共同产生成本g\_k

so at time there's a g cost of the curse then a G then Z and all of this is added up and at the terminal state there is another terminal cost which we denote by this function here

随着时间增加，把所有g累加起来，终止状态x\_N的成本时我们定义成本函数的时候定义的

these are scalar functions real valued function these are numbers here however they are random numbers Y random because no matter what what whatever you that you put here W makes this number a random number

这是一个标量实函数，因为函数的参数中有一个随机参数w，因此他的函数值是随机的

so the whole cost is random even if you fix the use the controls therefore in order to make this meaningful criterion for optimization you need to take expected value

所以整个成本是随机的，为了采用一个有意义的标准获得决策，你需要用函数的期望值作为目标函数进行优化

the expected value here is with respect to all the random values so if I apply some use in here then I get a number and I can ask the question what are the use that make this cost minimum that's our problem okay

这个期望值是所有随机变量共同作用获得的，所以如果我执行一些控制u，我可以得到一个成本值，这时候我就可以回答用什么控制可以最小化成本的问题了，这就是我要解决的问题

now one way to describe dynamic systems is by an equation like this

现在描述动态系统的一种方式是用这样的方程组

this is an alternative way but depending on the context may be more convenient

这是另一种依赖于上下文但是更方便的方式

and that is through a transition through transition probabilities

这种方式是通过状态转移概率描述随即系统的

this p here is a conditional probability that gives you for the current value of x in the current value of u the probability distribution of the next state

p是一个条件概率，给出了当先系统状态x下执行控制u产生新状态的概率分布

this is an alternative but equivalent description of a system

这是一种等价的系统描述方式

because he come back if I have a system like this

如果我们有一个这样的系统

I can generate these transition probabilities from this equation given the distribution of wk

我可以从给定的方程组的随机变量w\_k的分布生成这些转移概率

I can get the distribution of XK plus alternatively if I give you this and then you can define a system of this form in a very simple way

然后就可以获得x\_{k+1}的状态分布，如果我把这些信息告诉你，你就可以用一种非常简单的方式定义一个这种形式的动态系统

consider this special equation where XJ plus is just the W and give us distribution of W to be the distribution of XK plus this is not going to be so important in our course

考虑到这些特殊的方程组，x\_{k+1}是随机变量w，同时给了我们随机变量w的分布，我们就可以知道x\_k的分布和w\_k的分布是一样的，但是这不是我们课程的重点

but if you see in the literature a subject developed in terms of a model with transition probabilities be aware that's equivalent

但是如果你在文章中看到这种转移概率模型，你要知道他在描述一个动态系统

it can all the results can be transformed into our forward okay

这种描述方式可以等价转换为我们之前提到的动态系统

now what I'd like to do is give you an example of a system like this

接下来我想要给你举一个这样的例子

and that's what we will do is develop the dynamic programming algorithm for this problem

这就是我们要做的，为这样的问题开发一个动态规划算法

INVENTORY CONTROL EXAMPLE

So this is an inventory control problem

这是一个库存控制问题

a warehouse where you store face and then customers come and you sell them and then you have to order more to replenish and so on [Music]

你在一个仓库存了一些货物，然后来了一些客户，你把货物卖掉满足这些客户的需求，然后你不得不下订单补充货物

nobody ever calls me here okay

没有人给我打电话

okay okay so this is an inventory system okay

这就是一个库存系统

XJ is the stocker period J

x\_k是阶段k的库存状态

so let's say this is a system call we store cards exchange the number of cars at pancake

举个例子，我们把汽车存在仓库里，改变汽车的数量

now had during this time period K customers come and they demand a certain wk quantity which has to be subtracted from XK at the same time at this period we order stock

现在阶段k来了几个客户，他们对汽车有需求w\_k，这样新的库存数量就是x\_k减需求w\_k，同时下订单补充库存

we order new cars UK okay that's UK and therefore the number of the stock at period k plus one evolves according to this equation

我们补充了u\_k量汽车，新阶段的库存数量根据这个方程计算获得

what we have what additional we get minus what we give away XJ plus it is an equation of the form that I gave you a discrete time equation at the same time

there's some cost to be paid for our ordering decisions when I order UK

我们下订单u\_k的时候需要支付订货成本

then I have to pay a certain amount C is the price of that UK UK price per car that I pay so I have to pay this cost

现在有一个确定的汽车价钱c，c是u\_k的单价，我需要支付订货成本

and there is also some additional cost at the end of the period when I look at the inventory

在这个阶段末我查看库存的时候，还有一些成本

I want the inventory to be roughly 0 okay

我希望这时候的库存差不多是0

I don't want to have excess inventory what I have shortage of inventory so there is some costs associated with having having too much or too little at the end of the period

我不希望库存过多，同样不希望库存太少，所以这部分成本包括本阶段结束时货物过多的成本和货物过少的成本

so that's a common formulation of the problem minimize over any time periods

所以这是一个一般性的所有时间阶段最小化问题

the total ordering costs plus the cost associated with having too much or too little inventory

总成本包括订货成本加库存成本或者缺货成本

and my discrete time system is this

这就是我的离散时间系统

so this is an example G n sub X n is the value of the leftover inventory at the end okay

所以这就是一个例子，g\_N(x\_N)是最后一个阶段的剩余库存成本

it's it's a problem that fits the classical form

这是一个经典的动态规划问题

you can take GN equals to if you throw away the final Nitori

你可以让g\_N等于0，如果最后你把剩余库存扔掉的话

so it's a classical example wki random variable okay

w\_k是随机变量

that's where the stochastic nature of the problem comes in

这就是问题的随机性

UK has to obey some constraints

U\_k必须满足约束

you cannot order negative inventory it has to be zero or or poor positive okay

你不能让库存量是负的，它必须是0或者正数

sometimes there might be constraints that depend on the state

有些时候约束可能依赖于系统状态

if your warehouse cannot hold more than a hundred cars that you can't order more than a hundred right

如果你的仓库不能存放超过100量汽车，那么你不能定超过100量汽车的订单

ADDITIONAL ASSUMPTIONS

so here are some additional assumptions in this model

这个模型还有一切假设

the probability distribution of WK does not depend on past values

w\_k的概率分布不依赖于过去的信息

you assume that demands are not correlated but we may depend on XK and UK

你会假设需求是不相关的，但是他们可能依赖于系统状态x\_k和控制u\_k

you don't want dependence of the the uncertainty on the past

你不希望不确定信息以来过去的信息

because if there was such dependence then it would provide information past past demands would provide information about the current demand and that could be exploited in in the optimization

因为如果不确定信息依赖于过去的信息，它可能会提供过去的信息，需求也会提供一些关于当前需求的信息，并在优化中加以利用

so so to make the formulation simpler

所以为了让公式更简单

we assumed that the current w's do not depend on past seven years okay

我们假设当前的随机变量w不依赖于过去的信息

now here's something that's very important for a subject what do we optimize over do we optimize over sequences of orders of cars or do we optimize over something more complex

现在有一些对我们的主题非常重要的东西，我们优化汽车订单的序列或者更复杂的东西

indeed we optimize of over something more complex called policies for feedback control laws

事实上，我们优化更复杂的东西被叫做反馈控制律的策略

which are rules that tell us how much to order given the current value of inventory

这个策略告诉我们如何在当前库存状态下进行订货

so we optimize over functions functions of the current state that tell us if you have a hundred cars open or there's so much if you have five cards or there's so much clearly it is important to adapt our decisions to the current information to the current number that we have - what has happened in the past it would not be a good idea at the beginning of the horizon to make all our orders all our decisions about orders because it is clearly useful to take advantage of information whether we sell a lot or we sell a little how much we have and so on

so that's a major distinction we minimize over sequences of functions mapping each one mapping inventory to order

所以这是一个很主要的区别，我们优化一个序列的函数映射，每一个函数都从库存到订单量的映射

so our search space is over objects like this, a function of the initial inventory, then at time one a function of the inventory at that time X1 all the way to the end not over sequence of so control sequences of workers

所以我们搜索的空间是这些决策，初始化库存的函数，时间为1时关于库存x1的函数，一直这么搜索下去直到最后一个函数，而不是搜索整个控制序列

so the search space is a lot more complicated than this

所以搜索整个控制决策的空间要比一个一个搜索的空间更大

and that's why a lot of these other techniques like linear programming nonlinear programming techniques do not really apply in their entirety

这就是线性规划、非线性规划等技术没法应用在这整个问题上的原因

it's because we minimize over feedback laws and clearly there's a lot of benefit for doing that

因为我们在反馈规律下进行最小化显然更有优势

if we restrict ourselves to optimize over sequences would clearly we're not going to do as well as if we exploit the information okay

如果我们能够利用现有的信息，那么我们优化整个控制序列显然不会比一个一个搜索效果更好

so now we are going to go back to the general case remember a general problem involves a discrete-time system state control disturbance an additive cost function

现在我们再回去看一下一般性的情况，还记得一般性的有扰动的离散时间系统控制的累加成本函数

GENERIC FINITE-HORIZON PROBLEM

and now let's look at a formal description

现在我们看一下正式的描述

we have a system of this type

有一个这种类型的动态系统

we have control constraints at each time

每一个时间点都有控制约束

we are restricted to choose constraint from some set that depends on the current state

我们被限制根据当前系统状态从约束集合中选择控制

we are given the probability distribution of W depending on XK and UK the conditional distribution

依赖系统状态x\_k和控制u\_k的随机变量w的条件概率分布已知

we consider policies that is sequence of functions each of this news mapping state to control

我们考虑使用策略，即序列函数进行控制，策略指的是从系统状态到控制的映射

and these functions must start satisfy this control constraint and for any one of these policies

对于任何一个策略，函数必须满足控制约束

we formulate a cost that depends on the initial state x and on the policy and it is the cost incurred if we use this policy

我们设计依赖于初始状态和策略的成本函数，当使用策略进行控制时，可以观测到成本

in other words if we plug in the control back we will use at each one of these states, then this becomes a random number which is averaged over all the W's and give us this number okay

换句话说，如果我们把每一个系统状态插入策略中，然后累加成本就是一个观测所有随机变量w后求平均计算获得的的随机数，最后给我们一个总成本值

so this is the cost associated with a policy starting at the initial state x this is called the cost function of the policy for every initial state is a value okay so we're dealing with a function

所以这个总成本和策略，初始状态都有关系，这个函数被叫做策略的成本函数，对于任意给定的初始状态，这个函数都可以计算出一个数值，所以我们处理的是这些函数

and the dynamic programming aims to compute the optimal cost function given by this expression

动态规划的目标是最小化给定的成本函数

so given this cost of policies minimize over all policies to get J star that's the optimal cost heart state Excel starting from x0 it is a function of the initial state x0

即在给定初始状态后在所有策略中找到让J最小的那个策略，最优策略对应的成本叫做J\*，这就是最优化成本的核心目标，J\*也是初始化状态的函数

okay now here's something interesting

下面有一些有趣的东西

if you look at this expression

如果你看一下这个表达式

you would think that the policy depends on X0 there is no reason superficially looking at this

你会认为这个策略依赖于x0，但是实际上并不是这样的

why if you plug in a different x0 here you would use that you would use the same policy

如果你换一个输入，你还是使用相同的策略

it turns out however that because we're dealing with functions there exists you typically an optimal policy PI star which satisfies this equation for all X0 pi star the optimal policy is independent of the initial state

这是因为我们处理这些函数的时候，我们找到的最优策略pi\*对于所有初始化状态x0，都是最优的，就是说，最优策略与初始状态无关

this will come out as a consequence from the dynamic programming algorithm okay

这就是动态规划计算出来的结果

so this is a mathematical description of a generic finite horizon problem

这就是通用的有穷维问题的数学描述

notice I've made no assumptions about the spaces where this access use and abuse live X may take values from a Euclidean space

注意，我没有对动态规划的状态和控制进行任何假设，对于任意的x，都可以算出相应的价值

like for example you have a car on airplane moving in space okay

比如现在的问题是一辆车在飞机上移动

typically the motion of this plane is defined by six state values okay

通常这架飞机的运动是由六个状态值定义的

the positions the three coordinates of the position and the three coordinates of the velocity

关于位置的三个坐标和速度的三个坐标

so X is six numbers okay for such a port for such a motion and X lives in six dimensional space a continuous space

所以状态x是由六个数构成的，这样的港口和这样的运动状态在六维连续空间内取值

however there are other problems where X lives in a much simpler space like a discrete space a finite space

然而还有另一个问题，x在一个更简单的状态空间中取值，比如有限离散空间

think for example of a queuing system

举个例子，排队系统

a queuing system where typically the state is the number of customers that are in the queue

一个排队系统的状态一般由队列中的客户数量构成

assuming that your queue has a finite number of spaces for customers

假设你的队列中的客户数量是有限的

then you have a finite number of states a very different much simpler space

这时候你就有一个有限个数的状态，这是一个更简单的状态空间

you could have a network of queues which is more complicated multiple queues and then the Cartesian product of finite sets still a finite set

更复杂的情况，你面对的是一个排队网络，这些单个队列的笛卡儿积得到的排队网络的状态集合还是一个有限集合

the nice thing about dynamic programming is that it's spirit does not depend on where the main barrier to live whether they live in a continuous phase where a discrete space the same reasoning the same algorithms apply

动态规划的好处是，它不依赖于状态和控制的空间，连续空间和离散空间都可以使用相同的算法

ok that's a big thing because makes the techniques the technique very general, similarly the control can be in a Euclidean space can be in a finite set like an on/off decision and the same thing with W

这是一个很重要的性质，因为它让这种技术能够通用，可以简单地在欧几里得空间、有限集合，比如开关决策中应用

PRINCIPLE OF OPTIMALITY

okay there's something called principle optimality people attribute to it magical magical qualities lives indeed at the heart of our dynamic programming algorithm

现在有一个叫做最优性原理的东西，这是动态规划最核心的东西，他很神奇

but it's also a trivial statement a very very simple statements and it it may surprise you that it is the foundation on which just about all of dynamic programming is is built

但是它也是一个很不重要的东西，它可能会让你惊讶，因为它只是一个很简单的概念，整个动态规划都是在这个基础上建立起来的

what is the principle of optimality the name by the way is due to bellman principle man's associate bellman he was the one who who put it forth and showed that underlies many different models and algorithms dynamic programming type ok

最优性原理又叫bellman原理，bellman是提出动态规划的人，许多不同的模型和算法都可以用动态规划解决

what's the principle of optimality remember that we are looking for an optimal policy a Pollock's is a sequence of functions

还记得么，我们想要寻找一个序列函数的最优策略

suppose we have an optimal sequence of functions which are called P star PI star okay

假设我们有一个最优序列函数，这个序列的函数就叫做最优策略，记作pi\*

now let's our problem starts at time 0 and goes all the way to time n and write a finite horizon

现在我们要解决的问题从0时刻开始，n时刻结束，这就是要给有穷维问题

suppose that I look at a subproblem whereby I have arrived at some intermediate state X K at time K and I look at the tail problem of how to go from XK to the end optimally ok

假设现在我们有一个子问题，从中间状态x\_k开始，到终止状态结束，要寻找从时间k到时间n的最优策略

so I have an optimal policy for the entire problem and I also have formulated state sub problems which involve the same system the same control constraints and so on

现在我找到了一个整个问题的最优策略，我建立了子问题的表达式，这个子问题在同样的系统下受到同样的控制约束

and we assume that we are at X K at time K and we wish to minimize the cost to go from here to the end what is this cost to go

假设我们现在在时间k观测到系统状态x\_k，我们希望最小化从当前时刻到终止时刻的累加成本，这个累加成本叫做cost to go

it is the sum of the states course starting at time K to the end and having also this terminal cost so the total cost and the tail sub-problems costs and the principle of math is the following rather obvious things think the tail policy okay

从成本包括从时刻k到终止时刻n的状态曲线的成本和终端成本，因此总成本和子问题成本依据数学原理，很显然子问题更好解决

this is the tail policy that starts from XK to the end the tail portion of this optimal policy is also optimal for the tail sub problem

最优策略从x\_k到终端状态x\_n的那部分策略同时也是这个子问题的最优策略

in other words if I find myself at XK using this optimal policy and someone tells me forget about the past do optimally from here to the future then I would continue to use the optimal policy the tail portion of PI star is optimal for this problem

换句话说，如果我发现现在处于k时刻，系统状态x\_k，如果我不管之前的状态和控制，想要从当前状态到终止状态寻找最优策略，找到的最优策略对应的pi\*和这个问题的最优策略从x\_k到x\_n的最优策略是一样的

now the idea is that dynamic program is solved simultaneously all the tail sub problems doesn't solve just the problem of going optimally from here to here starting from a given initial State but it also solves all the tail subproblems no matter what well you may end up at any intermediate State XK find an optimal policy for this tail sub problem

现在可以知道，动态规划能够解决所有tail子问题，寻找从时刻0到时刻n的最优策略，也就找到了你从任意一个中间状态到终端状态的最优策略

so all of them simultaneously and all of them can be put together their optimal policy into a single big policy how does it do that at the generic step

这些子问题的解可以放进同一个完全的策略中

it solves all the tails are problems of given time length using the solution of the tail subproblems of shorter time lengths okay very simple idea

它使用更短时间长度的方案解决了所有给定时间长度的的子问题，一个很简单的思路

let me explain it

让我继续解释一下

we start over here one step to the end we solve this tail sub problem then we go to two steps from the end and we solve this state sub problem but the solution simplifies because we have already solved the shorter tails our problem then we go to three three three steps and using the solution for the two steps and keep going backwards until we get to the end that's the dynamic programming algorithm

我们从倒数第二个状态出发，解决这个子问题，然后从倒数第三个状态出发解决这个子问题，但是会更简单因为我们已经获得了更短的子问题的解，然后我们从倒数第四个状态开始，使用倒数第三个状态的策略计算这个策略，然后一直往前推，直到初始时刻，这就是动态规划算法

is this principle or melody here clear and any questions that you might want to ask just to give one more example

这个原则很清晰，如果你还有问题想要问，我就用一个例子来说明

suppose that I want to start there I want to go by car from my home which is in Massachusetts in the United States I want to go to California the opposite cause got Los Angeles and suppose that I calculate the optimal route the fastest route and it goes through Chicago okay

假设我想要开车从我在美国马萨诸塞州的家到加利福尼亚洛杉矶，还想要路过芝加哥，此时我需要计算路过芝加哥的最快的路线

so I have this route suppose now that I fly to Chicago and I want to find the shortest route by car from Chicago to Los Angeles what would be the optimal route well it would be the tail portion of the route that I would have taken had I started from Massachusetts shorter tail portions or fruits are optimal of optimal overall routes are optimal for the tail subproblems okay

所以我现在假设这样一条路线，飞往芝加哥，然后找开车从芝加哥去洛杉矶的最短的路线，这就是一个子问题，也是我采用的路线的尾部，在所有路线中找到从马萨诸塞州出发的更短的最优路线，这条路线的尾路线也是尾子问题的最有路线

DP ALGORITHM

so now let's give the dynamic programming algorithm

现在我要给出动态规划算法了

let us call JK of XJ the optimal cost of the tail subproblem that starts at xj

我们把从x\_k开始的尾子问题的最优成本函数叫做J\_k(x\_k)

now we want to have an algorithm that gives this JK and I start from this initial condition at the very end the last tails a problem where I'd landed at XJ this is next and there's not nothing to optimize this is the cost

现在我需要一个算法能够计算J\_k的值，我要从最后一个状态的初始化条件开始，这没有什么可以优化的，这是最后一个状态的成本

and now we go backwards one step at a time having computed JK plus one the optimal cost of the tail subproblem starting at the next state

现在我一次往前一个阶段计算当前状态的下一个状态的最优成本J\_{k+1}

we add the cost of the additional of the current stage and we minimize the sum of the two

我们把当前阶段的所有成本累加，然后最小化这两项的和

so given all the JK plus once I calculate all the JK's Nibin all the xn I calculate J and minus one for every possible value of the state then J n minus 2 J n minus 3 all the way to 0

计算出J\_{k+1}以后我要计算所有的J\_k，然后计算所有情况下J\_{k-1}的值，然后计算J\_{k-2}、J\_{k-3}直到J\_0

in other words the interpretation the word interpretation of this equation is to solve the tail subproblem at time J i minimize the sum of the k-th stage cost plus the optimal cost of the next pain problem starting from the next day at time k plus now

换种方法解释，这个等式想要解决尾子问题是，J\_k的值，最小化第k个阶段的成本加上下一阶段的尾子问题的最优成本

this is an algorithm that a computer can understand you give it an initial function this GN and it cracks out she has a black box that tracks out functions into functions k plus 1 functions to K and then UK minus 1 and so on each time generating a function okay

这是一个计算机能够理解的算法，你给定了一个初始函数g\_N，计算出函数k+1后，再计算函数k，这样往前计算下去，每一个阶段产生一个函数

now I see what happened I forgot an extra slide in here okay

我看看发生了什么，我忘记了，这里还有一页slide

here we are here's the algorithm to solve the ten subproblem at time k

在时间k算法求解了是个子问题

we do this and then at time zero we have jay-z rows of x generate the flag step and it is equal to the optimal cost because the last tail problem is the original problem so J0 sub X0 as generated by this algorithm is exactly J star also if I record the minimum the u that attains the minimum in each X at each X K and K I will obtain a sequence of functions of state where at XK new case of XK minimizes in the right hand side this equation

这样做了之后，在0时刻我们有 这和最优成本是相同的，因为最后一个尾问题是原问题的一部分，所以这个算法生成的J\_0(x\_0)也是J\*，同样，我记录了每个状态x令总成本最小的u，我可以获得一个状态函数的序列，这个状态序列是最小化等号右边的表达式获得的

so this algorithm simultaneously gives me the solute the optimal cost of the tail subproblems and also an optimal policy by minimization in the right hand side

所以这个算法得到的各阶段最优成本和尾子问题的最优成本是一样的，同样地，最优化等号右边的表达式，可以获得他们的最优策略

and actually the proof of this is very simple just by induction you assume that this is JK plus 1 is the optimal possibly takes a problem you prove the same property for JK it's it's not going to go into the proof of this but you may be able to just on the base of attrition figure it out yourselves it's also what you can find in the sources

事实上这个证明很简单，你可以假设J\_{k+1}是最优的，那么你可以证明J\_k也具有相同的性质（最优性），这里不会进行详细的证明，你可以在这个基础上自己弄明白，也可以在我提供的资源中找到这个证明

Q&A

so that's the algorithms have any questions here yes some go forward okay the question is whether you can execute this algorithm going backwards going forward rather than backward it turns out that you can execute it going forward but only for a special class of problems problems where there is no uncertainty where deterministic problems where the W's take a single known value then we can sort of reverse the direction of time and developed of the same algorithm but with fine reversed and so forward out there however there is no forward algorithm for stochastic problems it's just not possible to do it probably the variables we can optimize the objective over functions No to find a perhaps you can you can approximate this algorithm and do things in a different world but to get the optimal solution with no alternative really to this yes okay you're taking me five lectures forward what's the difference between this algorithm and q-learning to learning is is an algorithm that uses sampling okay it does not calculate expected values like here instead it uses samples of W so it's an algorithm that's that's more complicated more general and I think it ties in with us with a connection between dynamic programming and simulation we're going to get into this in the fair amount of detail next week so I'd rather wait until then yes why is it white spiced are independent well because you're solving all the tails subproblems in this way and it comes out to be what you get is something that works for every initial state it's the nature of the algorithm that it is for okay so this is optimal for the entire thing is optimal for any initial state and every pain portion is also optimal for the tale problems okay just because dynamic programming needs to find the optimal policy of all the tales have problems at the given stage so at the end it also solves all that the poldek all the original problems from any initial state yes that's why we live far away they can't figure out it so I can't admit other types of course function okay so the question is that here we have a problem defined as as with an expected cost problem it's possible to have a different kind of problem like a max the maximum over some of over the W square W is given by some kind of set description what W may be chosen by an antagonistic opponent there is a dynamic programming algorithm for such problems as well and the rationale is similar and I'm going to touch upon it a little later but for support for simplicity and specificity in this course I'm going to focus on expected values however they there are there are generalizations certainly yes which ended under so small Hey and define a problem or smaller for isin you mean and consider that well we do that basically right we look at all the problems of smaller horizon in the process okay J and minus this pertain here and so on JK is obtained here so we solve all the intermediate problems with this algorithm ok I'd like to take a break I think hours is too much for for me and I'm sure for Yuko and so what I'd like to do is every minutes or so give or take - minutes take a break for minutes and then you have a chance to digest correlate questions and rest a little bit and I have to say so let's break and get back in exactly minutes please