In many problems we are faced with making optimal choices. This requires either maximizing (or minimizing a function). There are methods for doing it often with the help of calculus. There are efficient numerical methods as well. They work in principle for functions of several variables. However as the number of variables increase it gets harder and soon nearly impossible.

Dynamic programming, is a natural way of facing this problem in special cases. Let us formulate the problem as follows. There is system in a state $x \in X$. We can take action $u \in \mathcal{U}$. These are called controls. The actions have rewards and consequences. The reward is a pay off $f(x,u)$ that depends on the state x as well as the control u. the consequence is that the control moves the state to a new state $x' = g(x,u)$ that depends on the control and the old state. We now go on from x' as before. Recursively

$$\begin{aligned} x_{j+1} &= g(x_j, u_{j+1}) \\ r_j &= f(x_j, u_{j+1}) \end{aligned}$$

The total reward is

$$R_n = f(x_0, u_1) + \cdots + f(x_{n-1}, u_n) + H(x_n)$$

where H is the reward or value inherent in the final state. This is finite time horizon problem involving n time steps. We could formulate the discounted infinite time horizon problem as well with a total reward of

$$R = \sum_{j=0}^{\infty} \rho^j \, f(x_j, u_{j+1})$$

R is a complicated function of x_0 and the controls $u_1, \ldots, u_n$, in the finite horizon case and a function of infinite number of variables $u_1, u_2, \ldots$, in the infinite horizon case. The idea of Dynamic programming is very simple. Suppose $n = 1$. We have only one period and we have to do the optimization problem

$$\sup_u [f(x,u) + H_n(g(x,u)]$$

where $H_n = H$. Let us denote by $u(x)$ the optimizing value and $H_{n-1}(x)$ the optimum value. Note that $H_{n-1}(x)$ is the value of a state with one period left. Now iterate

$$\sup_u [f(x,u) + H_{j+1}(g(x,u))] = H_j(x)$$

with $u = u_{j+1}(x)$ being the optimizing control. Then the problem is solved by $u = u_j(x)$ expressing the best control as a function of time and the current state. $H_0(x)$ provides the optimum value. The infinite horizon discounted problem is simpler. If we solve

$$\sup_u [\rho H(g(x,u)) + f(x,u)] = H(x)$$

for both H and u (by iteration starting from some H) then $u = u(x)$ is the best policy and provides a value $H(x)$. In the finite horizon case it is obvious that we have the answer. But the infinite horizon needs some thought. Let us suppose that $H(x)$ is the optimum value for all possible choices of controls. If we use u we have a reward of $f(x, u)$ and we end up at $g(x, u)$. The best we can do now is $\rho H(g(x, u))$ keeping in mind the discount. So the total payoff is

$$\rho H(g(x, u)) + f(x, u)$$

and we should optimize here for the best payoff. Hence H solves the equation.

$$\sup_u [\rho H(g(x, u)) + f(x, u)] = H(x)$$

Does H exist? Growth conditions are needed. If the value defined as the supremum over all payoffs is finite then H exists. Is it unique? If the solution H does not grow too fast and $\rho^n H(x_n) \to 0$ for any choice of controls, then it is unique. Any such solution is implementable.

The transformation $x \to g(x, u)$ need not be deterministic. Could involve a random component as well. So we have a transition probability $\pi(u, x, dx')$ that depends on the control. Now the problems are modified:

$$H_j(x) = \sup_u [f(x, u) + \int \pi(u, x, dx') H_{j+1}(x')]$$

Optimizing expected values. Infinite Horizon case

$$H(x) = \sup_u [f(x, u) + \rho \int \pi(u, x, dx') H(x')]$$

Examples: $f(x, u) = f(u)$ a concave function. $g(x, u) = x + a - u$. u is the consumption. a is the income. f is the utility. Add a discount that encourages consumption now while the concavity of f discourages excessive consumption.

$$H(x) = \sup_{0 \le u \le x} [f(u) + \rho H(x + a - u)]$$

If the income is random i.i.d

$$H(x) = \sup_{0 \le u \le x} [f(u) + \rho \int H(x + a - u) \phi(a) da]$$

Continuous time problems.

Consider the ODE

$$dx(t) = \sum_j u_j(t) b_j(x(t)) dt$$

where u is the control. $x(0) = x$ and at time T, the pay off is

$$\int_0^T c(x(t), \mathbf{u}(t)) dt + f(x(T))$$

Derivation of the equation: Suppose at time t we have an optimal value $V(t, x)$ if we start from x. If we start at time $t - h$ we could try the control $\mathbf{u} = \{u_j\}$ during $(t - h, h)$ and the payoff will be roughly

$$V(t - h, x) = hc(x, \mathbf{u}) + V(t, x + h \sum_j u_j b_j(x))$$

Yields

$$V_t + \sup_{\mathbf{u}} [c(x, \mathbf{u}) + V_x \sum_j u_j b_j(x)] = 0, \ V(T, x) = f(x)$$

This is a nonlinear first order PDE. Smooth solutions may not exist. If there is a smooth solution then it solves the problem. Because for any choice of $u(t)$ and the corresponding solution $x(t)$ of the ODE

$$\begin{aligned} \frac{d}{dt}[V(t, x(t)) + \int_0^t c(x(s), \mathbf{u}(s)) ds] \\ &= V_t + V_x \frac{dx}{dt} + c(x(t), \mathbf{u}(t)) \\ &\leq 0 \end{aligned}$$

Therefore

$$\begin{aligned} V(0, x) &\geq V(T, x(T)) + \int_0^T c(x(s), \mathbf{u}(s)) ds \\ &= f(x(T)) + \int_0^T c(x(s), \mathbf{u}(s)) ds \end{aligned}$$

for ANY choice of u. On the other hand with the optimizing choice of $\mathbf{u}$ we have equality.

Example: Consider the problem

$$\frac{dx}{dt} = u(t), |u| \leq 1, T = 1, f(x) = \cos x, c = 0$$

You are free to travel with speed up to 1 to get to a large value of $\cos x$. Formally the equation is

$$V_t + |V_x| = 0, V(T, x) = \cos x$$

Clearly

$$V(t, x) = \sup_{|y-x| \leq (1-t)} [\cos y]$$

In particular V is not continuously differentiable. On the other hand the verification lemma needs smoothness. Are there solutions (that are not C_1) that are spurious? How to tell the true one from spurious ones?

You can add noise and replace ODE by SDE say by adding a BM. Then the equations become

$$V_t + \sup_u [c(x, \mathbf{u}) + V_x \sum_j b_j(x) u_j] + \frac{1}{2} u_{xx}, V(T, x) = f(x)$$

These are more likely to have smooth solutions. The verification now uses Itô's formula.