Maximizing Firm Profit Function with Risk-Averse Labor Supply

Zach Siegel
UCLA Anderson School of Management
zachary.edmund.siegel@gmail.com

1 Introduction

During the recent COVID-19 pandemic, a primary concern in "flattening the curve" of infection is to prevent medical facilities from being overwhelmed. Unfortunately, while demand for medical personnel is high during a pandemic, supply of personnel may be reduced. It is essential for a medical firm to maintain a sufficiently high wage to capture or retain a sufficient fraction of the workforce.

While giving an overview of the domain-specific rationale for the model, this paper explores the optimization of the objective function, which is concave in each variable but not jointly concave.

2 Model

2.1 Worker Utility

It is established in [1] that individuals tend to be risk-seeking when facing a certain loss. Furthermore, workers who are more risk-averse with respect to their income may be more disinclined than others to face an expensive accident.

We consider the income utility and injury disutility to be additively separable. Let

$$\begin{aligned} U(r) &= \text{utility of payment } r \\ V(H) &= \text{disutility of injury of magnitude } H \\ \Psi(r) &= U(r) - V(H) \\ &= \text{Utility of wage } r \text{ and} \\ &\quad \text{injury of (fixed) magnitude } H. \end{aligned}$$

where $U'(r), V'(H) > 0$, and $U''(r), V''(H) < 0$ reflecting risk-aversion in gains and risk-seeking in losses. Utility $\Psi(r)$ is the utility of income r along with the pain and suffering of an injury, which must therefore satisfy $\Psi(r) < U(r)$. We consider H to be exogenous.

In general, neither a higher or lower income necessarily induces risk-aversion, and we consider that workers respond to income increases (via $U(r)$) with constant absolute risk aversion (CARA). In the domain of loss/injury (via $V(H)$), for simplicity we consider a (slightly different) CARA function as well. However, with the motivation that risk-averse workers are also more injury-averse, we designate one risk-aversion parameter $\theta > 0$ to attenuate risk response in both domains:

$$\begin{aligned} U(r) &= \frac{1}{\theta}(1 - e^{-\theta r}) \\ V(H) &= \theta'(1 - e^{-\frac{1}{\theta'}H}) \\ \theta' &= t\theta, t > 0. \end{aligned}$$

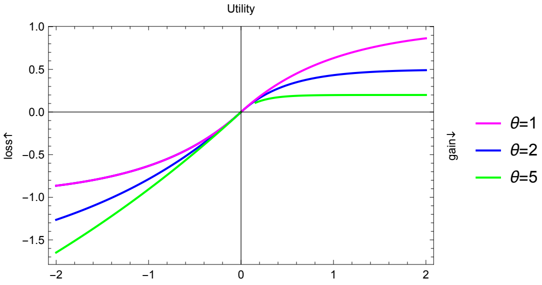

Figure 1: *Utility in gains and losses attenuated by a single risk-response parameter θ*

Note that as θ increases, $U(r)$ becomes flatter, reflecting risk-aversion in income. As θ' increases, $V(H)$ becomes *steeper*, reflecting less risk-seeking in losses. Allowing $\theta' = t\theta$ reflects that individuals' risk-aversion in income may accompany less risk-seeking behavior in losses. For simplicity, here we consider $t = 1$, or $\theta' = \theta$.

Figure 1 demonstrates workers' utility response to gains ($U(r)$) on the positive side of the horizontal axis, and utility response to losses ($-V(H)$) on the negative side of the horizontal axis. Note, as will be critical, that in addition to flattening and steepening the utility and disutility response curves, respectively, a higher risk-response factor θ also monotonically decreases utility at any income or injury level.

Remark 1. Both $U(r)$ and $-V(H)$ are decreasing in θ.

Proof. First, $U(r)$ is decreasing in θ:

$$\frac{d}{d\theta}\frac{1}{\theta}(1 - e^{-\theta r}) = \frac{(1+\theta r)e^{-\theta r} - 1}{\theta^2}$$

$$\leq 0$$

$$\Leftrightarrow$$

$$(1+\theta r)e^{-\theta r} - 1 \leq 0$$

$$\Leftrightarrow$$

$$1 + \theta r \leq e^{\theta r}$$

which is true as $1 + x \leq e^x$ for all x.

In addition, $-V(H)$ is decreasing in θ:

$$\frac{d}{d\theta} - \theta(1 - e^{-\frac{1}{\theta}H}) = -1 + (1 + \frac{H}{\theta})e^{-\frac{H}{\theta}}$$

$$\leq 0$$

$$\Leftrightarrow$$

$$1 + \frac{H}{\theta} \leq e^{\frac{H}{\theta}}$$

which is true for the same reason. □

2.2 Probability of Worker Injury and Labor Supply

Let a worker normally earn wage w but with probability p earn no income income due to an accident or injury that causes pain and suffering H. It is natural to consider a worker buying insurance at the level I that maximizes their expected utility.

As in [2], consider perfect, load-free insurance, which necessarily has price $\frac{p}{1-p}$ per unit insurance, so that the expected payout of insurance is 0:

$$E(\text{payout}) = \overbrace{p(-I)}^{\text{insurer pays}} + \overbrace{(1-p)(\frac{p}{1-p}I)}^{\text{insurer is paid}}$$
$$= 0.$$

That is, in the case of an accident, the insurer pays I to the worker; in the case of no accident, the worker pays $\frac{p}{1-p}I$. The expected worker utility is then

$$E = (1-p)U(w - \frac{p}{1-p}I) + p\Psi(I).$$

The conditions described for U and Ψ (concave, increasing) ensure a first-order condition will yield the optimal I without substituting the functional forms:

$$0 = (1-p)U'(w - \frac{p}{1-p}I)\frac{-p}{1-p} + p\Psi'(I)$$
$$\Downarrow$$
$$\Psi'(I) = U'(w - \frac{p}{1-p}I).$$

In the case of the CARA functional forms assumed above for U and Ψ, this implies $I = (1-p)w$ and the worker's expected utility E is

$$E = \frac{1}{\theta}(1 - e^{-\theta(1-p)w}) + p\theta(1 - e^{-\frac{H}{\theta}}).$$

We consider that a worker is willing to work at wage w if expected utility $E > 0$.

2.3 Firm Profit

Consider a firm that hires L workers at wage w at risk level p. As in [2], probability of accident p is endogenous, and the firm attempts to maximize profit as defined by production minus wages and cost of maintaining a desirable safety level.

The cost of maintaining safety level $(1-p)$ is $C(1-p)$, where $C(0) = 0$ and $\lim_{(1-p)\to 1} C(1-p) = \infty$. We assume C to be convex and strictly increasing. A natural candidate is

$$C(1-p) = c \cdot \frac{1-p}{p}$$

for some *cost of safety* $c > 0$.

Let the firm's (monetized) *production* function $\phi(L,p)$ represent the output of the firm with L employees at risk level p. *Production* could be represented by revenue or a monetization of services provided, such as patients served.

We consider the case of labor scarcity in which productive output may simply be proportional to labor level L and independent of probability of injury p:

$$\phi(L,p) = \gamma L$$

where γ represents the productive output per employee. Note that while $\phi(L,p)$ does not depend on p, the number of workers L does depend on p, which assumes that a worker can be equally productive at any probability p of injury such that their utility is positive. More risk-averse workers refuse to work at high p, but when they work they are equally productive.

The firm's profit can then be written

$$\begin{aligned}\Pi &= \phi(L,p) - wL - C(1-p)\\ &= (\gamma - w)L - c\frac{1-p}{p}.\end{aligned} \tag{1}$$

2.4 Worker Type and Population

Consider a heterogeneous population of workers characterized by their risk-response parameter $\theta \sim F(\cdot)$ where CDF F has positive support. Suppose there are $\overline{L}$ total potential workers, and a fraction β of them decide to work at a given p and w. That is, L in the firm profit function can be written

$$L = \beta\overline{L}. \tag{2}$$

As noted above, worker utility at any wage is strictly decreasing in the risk-response parameter θ: a higher θ entails a flatter and lower utility from wages, and a steeper and higher disutility from injury H. So, for a fixed wage w, probability of accident p, and injury level H, there is a risk-response $\tilde{\theta}$ above which workers will not work.

Then the fraction of workers who will work is $\beta = F(\tilde{\theta})$ where, at $\tilde{\theta}$, expected utility $E = 0$. That is, $\tilde{\theta}$ (> 0) satisfies the equation

$$\begin{aligned}0 &= E\\ &= \frac{1}{\tilde{\theta}}(1 - e^{-\tilde{\theta}(1-p)w}) + p\tilde{\theta}(1 - e^{-\frac{H}{\tilde{\theta}}}).\end{aligned}$$

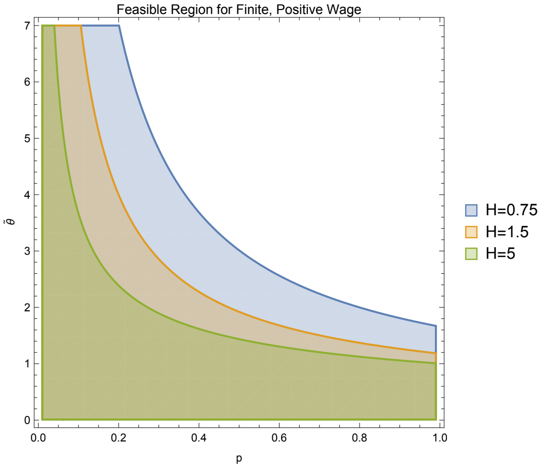

Figure 2: *The feasible region on probability of accident p and cutoff risk-resonse $\tilde{\theta}$ resulting in a finite cutoff wage.*

Unfortunately, the cutoff risk-response $\tilde{\theta}$ cannot be written analytically in terms of firm decisions p and w. However, the wage that induces cutoff risk-response $\tilde{\theta}$ can be written

$$w = \frac{-1}{\tilde{\theta}(1-p)} \ln\left(1 - p\tilde{\theta}^2(1 - e^{-\frac{H}{\tilde{\theta}}})\right) \tag{3}$$

where the obvious restriction that $0 < w < \infty$ corresponds to

$$p < \frac{1}{\tilde{\theta}^2(1 - e^{-\frac{H}{\tilde{\theta}}})}.$$

This is a non-convex constraint, with feasible region illustrated in Figure 2. Outside the shaded region, the risk and cost of an injury are sufficiently high to require an infinite wage to induce workers at risk type $\tilde{\theta}$ to work.

Finally, we assume a functional form for the worker type distribution function: $\theta \sim \text{Unif}[0, \overline{\theta}]$, and

$$\beta = F(\tilde{\theta}) = \frac{\tilde{\theta}}{\overline{\theta}}, \tag{4}$$

where $\overline{\theta}$ is the maximum worker risk-response.

Remark 2. Note that the wage strictly increases in $\tilde{\theta}$, meaning that aiming to capture more risk-averse workers requires a higher wage. The wage is also strictly increasing in p, meaning that a riskier job requires a higher wage to capture equally risk-averse individuals.

Proof.

$$\begin{aligned}\frac{\partial w}{\partial \tilde{\theta}} = & \frac{\ln(1 - p\theta^2(1 - e^{-\frac{H}{\tilde{\theta}}}))}{\tilde{\theta}} \\ & + \frac{2p\tilde{\theta}(1 - e^{-\frac{H}{\tilde{\theta}}}) - pHe^{-\frac{H}{\tilde{\theta}}}}{1 - p\tilde{\theta}^2(1 - e^{-\frac{H}{\tilde{\theta}}})} \\ \geq & \, 0\end{aligned}$$

$$\Leftrightarrow$$

$$\begin{aligned}&\ln(1 - p\tilde{\theta}^2(1 - e^{\frac{-H}{\tilde{\theta}}})) \\ &\qquad \geq - \frac{2p\tilde{\theta}^2(1 - e^{-\frac{H}{\tilde{\theta}}}) - pH\tilde{\theta}e^{-\frac{H}{\tilde{\theta}}}}{1 - p\tilde{\theta}^2(1 - e^{-\frac{H}{\tilde{\theta}}})}\end{aligned}$$

The left-hand side is bounded below by:

$$\frac{-p\tilde{\theta}^2(1 - e^{-\frac{H}{\tilde{\theta}}})}{1 - p\tilde{\theta}^2(1 - e^{-\frac{H}{\tilde{\theta}}})}$$

for the simple reason that for any $x > -1$, $\ln(1 + x) \geq \frac{x}{1+x}$.

The right-hand side is bounded *above* by this same expression: they have the same denominator, and comparing their numerators reduces to $1 + \frac{H}{\tilde{\theta}} \leq e^{\frac{H}{\tilde{\theta}}}$, which is true because $1 + x \leq e^x$ for all x.

The proof that $\frac{\partial w}{\partial p} > 0$ is trivial. □

2.5 Firm Profit Maximization

Substituting the wage equation (3), the labor supply equations (4) and (2), in the firm profit (1) yields the following maximization problem for the firm:

$$\begin{array}{lll}\underset{p,\tilde{\theta},w,L}{\text{Maximize}} & \Pi & = (\gamma - w)L - c\dfrac{1-p}{p} \\ \text{s.t.} & w & = \dfrac{-1}{\tilde{\theta}(1-p)} \ln\left(1 - p\tilde{\theta}^2(1 - e^{-\frac{H}{\tilde{\theta}}})\right) \\ & L & = \dfrac{\overline{L}}{\overline{\theta}}\tilde{\theta} \\ 0 \leq & p & \leq 1\end{array}$$

where w and L are clearly just splitting variables. The problem can be written succinctly as:

$$\begin{aligned}&\underset{p,\tilde{\theta}}{\text{Maximize}}\,\Pi = \qquad (5)\\ &\frac{\overline{L}}{\overline{\theta}}\left(\gamma\tilde{\theta} + \frac{1}{1-p}\ln\left(1 - p\tilde{\theta}^2(1 - e^{-\frac{H}{\tilde{\theta}}})\right)\right) - c\frac{1-p}{p} \\ &0 \leq p \leq 1 \\ &0 \leq \tilde{\theta} \leq \overline{\theta}\end{aligned}$$

Remark 3. Note that Π in (5) is concave separately in both p and $\tilde{\theta}$. While the (implicit) feasible region is not convex, this may still permit fruitful second-order optimization methods. Furthermore, this feasible region is compact due to the bounds on p and $\tilde{\theta}$, allowing straightforward sampling for numerical methods.

It would remain to show that Π is *jointly* concave in p and $\tilde{\theta}$ on its feasible set. In fact, it is not: it clearly violates Jensen's inequality. Note that the wage in (3),

$$w = \frac{-1}{\tilde{\theta}(1-p)} \ln\left(1 - p\tilde{\theta}^2(1 - e^{-\frac{H}{\tilde{\theta}}})\right),$$

approaches ∞ when the argument of the log approaches 0 (meaning $\Pi \to -\infty$) or when

$$p \to \frac{1}{\tilde{\theta}^2(1 - e^{-H/\tilde{\theta}})}.$$

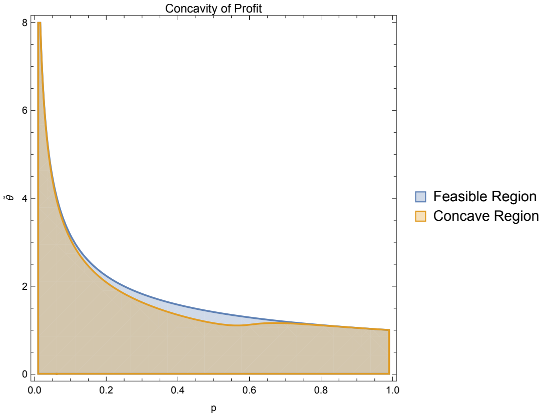

Figure 3: *The profit* Π *seems to be concave on its entire domain.*

This defines a nonconvex region in the $(p, \tilde{\theta})$ plane on which profit is finite. Figure 3 illustrates this region, which is implicitly the domain of optimization, as the objective function is not defined outside this region.

Still, numerical experimentation seems to show that Π is in fact jointly concave on the interior of its feasible set. A function of two variables with negative second partial derivatives is jointly concave if and only if its Hessian has positive determinant. Figure 3 also illustrates the region on which this condition holds, which seems to be the entirety of the feasible region (noting numerical instability on the boundary where $\Pi \to -\infty$).

Proof. (Π concave in p and $\tilde{\theta}$) Note that the second three terms (excluding $\frac{-2c}{p^3}$) of

$$\begin{aligned}\frac{\partial^2 \Pi}{\partial p^2} = \frac{-2c}{p^3} &+ \frac{\overline{L}}{\overline{\theta}} \frac{-\tilde{\theta}^4(1-e^{-H/\tilde{\theta}})}{(1-p)(1-p\tilde{\theta}^2(e-e^{-H/\tilde{\theta}}))^2} \\ &- \frac{\overline{L}}{\overline{\theta}} \frac{-2\tilde{\theta}^2(1-e^{-H/\tilde{\theta}})}{(1-p)^2(1-p\tilde{\theta}^2(e-e^{-H/\tilde{\theta}}))} \\ &+ \frac{\overline{L}}{\overline{\theta}} \frac{2\ln(1-p\tilde{\theta}^2(1-e^{-H/\tilde{\theta}}))}{(1-p)^3}\end{aligned}$$

are negative if and only if

$$\begin{aligned}\ln(1-p\tilde{\theta}^2(1-e^{-H/\tilde{\theta}})) \leq &\frac{(1-p)\tilde{\theta}^2(1-e^{-H/\tilde{\theta}})}{1-p\tilde{\theta}^2(1-e^{-H/\tilde{\theta}})} \\ &+ \frac{\frac{1}{2}(1-p)^2\tilde{\theta}^4(1-e^{-H/\tilde{\theta}})}{1-p\tilde{\theta}^2(1-e^{-H/\tilde{\theta}})}.\end{aligned}$$

For all values at which the left-hand side is defined, the right-hand side is positive, and the left-hand side is negative (as $\ln(1+x) \leq x \; \forall \; x$, the left-hand side is bounded above by $-p\tilde{\theta}^2(1-e^{-H/\tilde{\theta}}) < 0$).

Similarly, $\frac{\partial^2 \Pi}{\partial \tilde{\theta}^2} < 0$:

$$\begin{aligned}\frac{\partial^2 \Pi}{\partial \tilde{\theta}^2} = &-\frac{\overline{L}}{\overline{\theta}} \frac{\left(Hpe^{-H/\tilde{\theta}} - \tilde{\theta}pe(1-e^{-H/\tilde{\theta}})\right)^2}{(1-p)\left(1-p\tilde{\theta}^2(1-e^{-H/\tilde{\theta}})\right)^2} \\ &- \frac{\overline{L}}{\overline{\theta}} \frac{2p(1-e^{-H/\tilde{\theta}}) - \frac{H^2 p}{\tilde{\theta}^2}e^{-H/\tilde{\theta}} - \frac{2Hp}{\tilde{\theta}}e^{-H/\tilde{\theta}}}{(1-p)\left(1-p\tilde{\theta}^2(1-e^{-H/\tilde{\theta}})\right)}.\end{aligned}$$

The first term is clearly negative because of the squared terms in the numerator and denominator. The second term is negative if and only if

$$2p(1-e^{-H/\tilde{\theta}}) \geq (\frac{H^2 p}{\tilde{\theta}^2} + \frac{2Hp}{\tilde{\theta}})e^{-H/\tilde{\theta}}$$

which reduces to

$$e^{H/\tilde{\theta}} \geq \frac{1}{2}\left(\frac{H}{\tilde{\theta}}\right)^2 + \frac{H}{\tilde{\theta}} + 1,$$

which holds because $e^x \geq \frac{1}{2}x^2 + x + 1$ for all $x \geq 0$.

□

3 Optimization

As a benchmark, and to conduct sensitivity analysis for this model for another paper for another course, Mathematica's built-in `NMinimize` was able to optimize this objective over a wide range of parameters. According to [3], `NMinimize` uses a combination of a differential evolution algorithm and interior point methods to maximize over arbitrary regions with possibly many local optima.

The optimization implementations for this project were done in MATLAB, and MATLAB's `fmincon` function was also used as a benchmark.

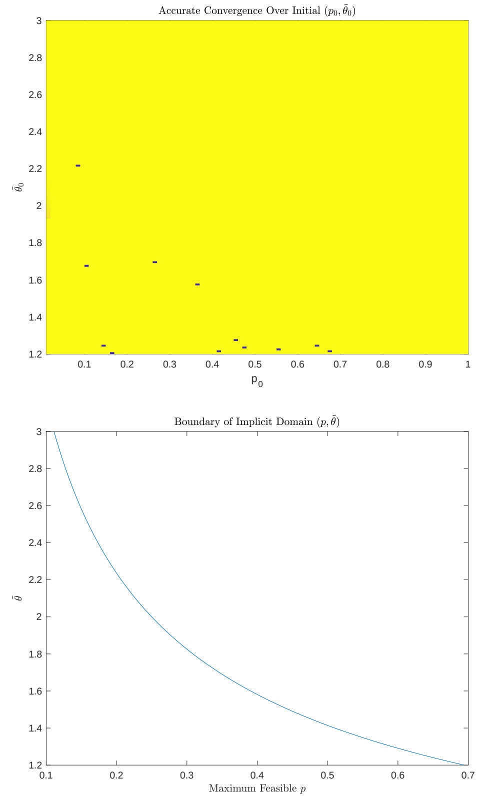

Figure 4: *Newton's Method converges to (close to) a globally optimal value of $(p, \tilde{\theta})$ only for very few inital points. The curve defines the points in the domain of problem (5) at which the objective, gradient, and Hessian are defined. The grid depicts the (very few) initial points at which a near-global optimum is attained.*

3.1 Newton's Method

Newton's Method is ideal for unconstrained optimization of twice-differentiable, convex functions.

The optimization problem (5),

$$\underset{p,\tilde{\theta}}{\text{Maximize}}\, \Pi = \frac{\overline{L}}{\overline{\theta}}\left(\gamma\tilde{\theta} + \frac{1}{1-p}\ln\left(1 - p\tilde{\theta}^2(1 - e^{-\frac{H}{\tilde{\theta}}})\right)\right) - c\frac{1-p}{p}$$
$$0 \leq p \leq 1$$
$$0 \leq \tilde{\theta} \leq \overline{\theta},$$

does not have a concave objective, and is constrained. Still, as the objective is separately concave in p and $\tilde{\theta}$, and Newton's Method may be able to find a solution when initialized at a feasible point. Furthermore, as illustrated in Figure 3, the function seems to have joint convexity on (much of) the interior of its domain.

Newton's Method can only proceed from an initial point at which a function, its gradient, and its Hessian are defined. As can be seen in Figure 4, even within this region, Newton's Method fails to converge to a near-global optimum from almost any initial points (with the near-global optimum determined by MATLAB's built-in `fmincon` function, which agreed with Mathematica's built-in `NMinimize` for several parameter values examined). Both of these perform *constrained* optimization.

The resulting values p^* from naively using Newton's Method were often outside of the feasible region $[0,1]$, though interestingly the unconstrained optimal $\tilde{\theta}$ seemed to be within $[0,\overline{\theta}]$ in all experiments.

The implementation of Newton's Method can be found in Section 5.1.

3.2 Newton's Method with Logarithmic Barrier

To enforce constraints, we employ a logarithmic barrier function, and maximize

$$\tilde{\Pi} = t\Pi + \ln(1-p) + \ln(p) + \ln(\tilde{\theta}) + \ln(\overline{\theta} + \tilde{\theta})$$

which means

$$\nabla\tilde{\Pi} = t\nabla\Pi + \begin{bmatrix} \frac{1}{p} - \frac{1}{1-p} \\ \frac{1}{\tilde{\theta}} - \frac{1}{\overline{\theta}-\tilde{\theta}} \end{bmatrix}$$

$$\nabla^2\tilde{\Pi} = t\nabla^2\Pi + \begin{bmatrix} \frac{-1}{p^2} - \frac{1}{(1-p)^2} & 0 \\ 0 & \frac{-1}{\tilde{\theta}^2} - \frac{1}{(\overline{\theta}-\tilde{\theta})^2} \end{bmatrix}$$

This optimization was performed with multiple values of t (from 0.5 to 1000) with comparable results. The accuracy of this method seems to be hindered more by the lack of joint concavity of Π than by the relative accuracy of the optima resulting from the barrier functions.

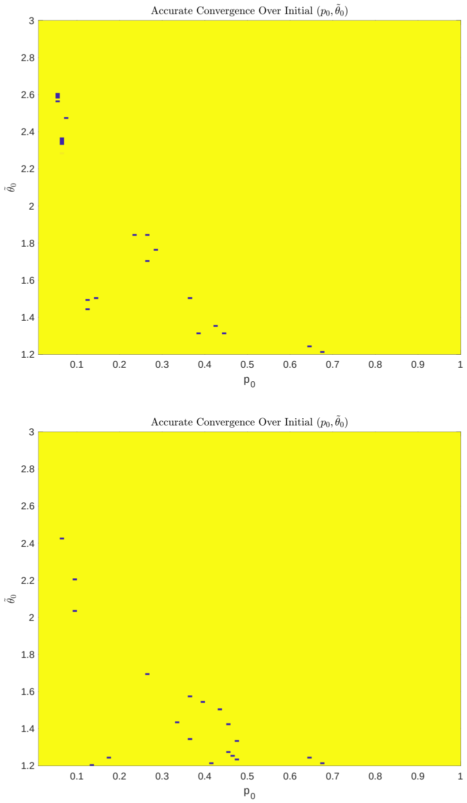

Figure 5: *The initial points from which Newton's Method with logarithmic barrier functions attained near-globally optimal objective. Parameters are the same as in Figure 4. In the first image, $t = 1$ and in the second $t = 100$.*

The accuracy of this method is examined in Figure 5. The code used to generate these plots is in Section 5.2.

3.3 Exhaustive Search Over One Variable

A natural solution to the non-concavity of (5) is to maximize over one variable while performing grid search over the other.

This seems like a suitable method for this prob-

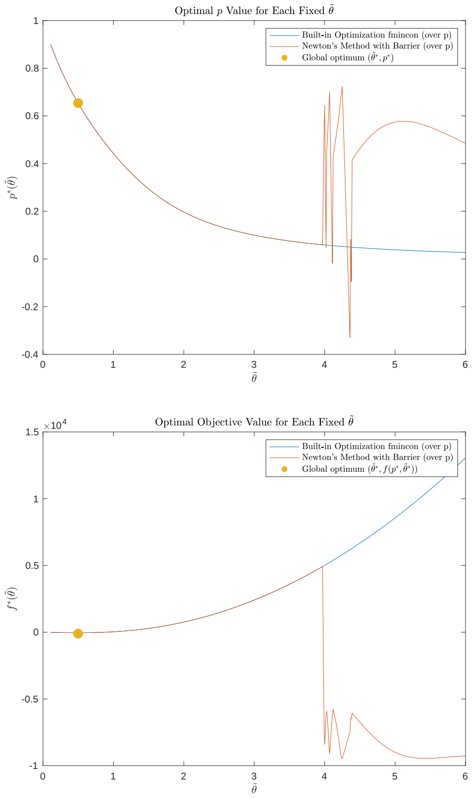

Figure 6: *The first figure depicts the optimal value $p^*(\tilde{\theta})$ found via one-variable minimization over an exhaustive grid of values of $\tilde{\theta}$. The second figure depicts the optimal value of the objective function $f(p^*(\tilde{\theta}), \tilde{\theta})$ evaluated at each value of $\tilde{\theta}$ and the optimal $p^*(\tilde{\theta})$*

lem, as the domain is compact, and the objective is concave in each of its variables.

As can be seen in Figure 6, this method achieves the following:

- Agrees with the output of the built-in `fmincon` over each one-variable optimization problem (except where numerically unstable).
- Coincides with the global optimum $(p^*, \tilde{\theta}^*)$ when the grid search reaches $\tilde{\theta}^*$.

Clearly, in Figure 6 there is numerical instability when $\tilde{\theta}$ is close to $\overline{\theta}$ (= 6 in the figure). This is an unfortunate by-product of the numerical technique, which seems to work extremely well over the region on which it is numerically stable.

The implementation of this method can be found in Section 5.3

4 Conclusion

Even on a compact, low-dimensional domain, and even with concavity separately in each variable, it is not always straightforward to maximize a given function made up of simple parts. The function Π in (5) is not (jointly) concave, and Newton's Method with logarithmic barrier functions fails to converge close to an optimum from almost any initial points in the domain.

Grid search over a (very) small number of dimensions is a practical and easy-to-implement solution to almost any optimization problem with a low-dimensional domain. Optimizing over the remaining variable(s) using second-order methods should be far more efficient than exhaustive grid search over the entire domain.

Comparing any technique with mature solvers such as `CVX`, `fmincon` (MATLAB), and `NMinimize` (Mathematica) is a constant reminder of the challenge of robust numerical stability under changes in parameters.

5 Code

5.1 Newton's Method Accuracy

```
Lbar = 30;
thetabar = 6;
ggamma = 2.25;
c = 5;
H = 35;
subfun0 = @(theta) exp(-H/theta);
subfun1 = @(theta) 1-exp(-H/theta);
subfun2 = @(p,theta) 1-p*(theta^2)*(1-exp(-H/theta));
subfun3 = @(p,theta) p*H*exp(-H/theta) - 2*p*theta*(1-exp(-H/theta));
const1 = Lbar/thetabar;
Pi = @(p,theta) const1*(ggamma*theta + (1/(1-p))*log(subfun2(p,theta))  ) -
c*((1-p)/p);
dPidp = @(p,theta)c*((1-p)/(p^2)) + c/p + const1 *(-( (
(theta^2)*subfun1(theta)  )/( (1-p)*(subfun2(p,theta) )  )  )   + (
(1/(1-p)^2) * log(subfun2(p,theta))   ));
dPidtheta = @(p,theta) const1 * (ggamma +     ( (p*H*subfun0(theta)
-2*p*theta*subfun1(theta))/((1-p)*(subfun2(p,theta)))     ));
PiGrad = @(p,theta) [dPidp(p,theta);dPidtheta(p,theta)];
dPi2dp2 =@(p,theta) -2*c*(1-p)/(p^3) - 2*c/(p^2) + const1*( (-theta^4 *
subfun1(theta))/((1-p)*subfun2(p,theta)^2) -(2*theta^2*subfun1(theta))/((1-p)^2
* subfun2(p,theta)) + 2*log(subfun2(p,theta))/((1-p)^3)          );
dPi2dtheta2 =@(p,theta) const1*(-subfun3(p,theta)^2/((1-p)*subfun2(p,theta)^2)
+ (-2*p*subfun1(theta) + (p*(H^2)/(theta^2) + 2*p*H/theta)*subfun0(theta) )
/((1-p)*subfun2(p,theta)) );
dPi2dpdtheta =@(p,theta) const1*(((theta^2) *
subfun1(theta)*subfun3(p,theta))/((1-p)*subfun2(p,theta)^2) +
(H*subfun0(theta))/((1-p)*subfun2(p,theta)) -
(2*theta*subfun1(theta))/((1-p)*subfun2(p,theta)) + subfun3(p,theta)/((1-p)^2 *
subfun2(p,theta)) );
piHess = @(p,theta) [dPi2dp2(p,theta),dPi2dpdtheta(p,theta);dPi2dpdtheta(p,theta),dPi2dtheta2(p,theta)];
f = @(x) -Pi(x(1),x(2));
fgrad = @(x) -PiGrad(x(1),x(2));
fHess = @(x) -piHess(x(1),x(2));
final = @(x) -f(x);
newtonstep = @(x) - inv(fHess(x)) * fgrad(x);
lambdasquared = @(x) -fgrad(x)' * newtonstep(x);
```

```
pmax = @(theta) 1/((theta^2).*subfun1(theta));
w = @(p,theta) (-1/(theta*(1-p)))*log(subfun2(p,theta));
% constraints = [pConstraint,thetaConstraint]
theta0 = 0.5;
p0 = (1/2)*min(pmax(theta0),1);
x0 = [p0;theta0];
A = [1,0;-1,0;0,1;0,-1];
b=[1;0;thetabar;0];
x_builtin = fmincon(f,x0,A,b);
p_star_builtin = x_builtin(1);
theta_star_builtin = x_builtin(2);
f_star_builtin = f(x_builtin);
pi_star_builtin = -f(x_builtin);
max_inner_iterations = 15;
max_outer_iterations = 100;
epsilon = .00000001;
t0 = 1;
x = x0;
p0_vals = 0.01:.01:1;
theta0_vals = (thetabar/5):.01:(thetabar/2);
np0 = length(p0_vals);
ntheta0 = length(theta0_vals);
fvals = zeros(np0,ntheta0);
p_star_vals = zeros(np0,ntheta0);
theta_star_vals = zeros(np0,ntheta0);
for i = 1:np0
  for j = 1:ntheta0
    p0 = p0_vals(i);
    theta0 = theta0_vals(j);
    x0 = [p0;theta0];
    x = x0;
    if p0 < pmax(theta0)
      fprintf("Starting Newton: p0=%.2f,theta0=%.2f\n",p0,theta0);
      counter = 0;
      while lambdasquared(x)/2 > epsilon && counter < max_outer_iterations
        counter = counter + 1;
        % fprintf("Newton iteration %d\n",counter);
        dx = newtonstep(x);
        t = t0;
        xnew = x + t* dx;
        inner_counter = 0;
        while f(xnew)> f(x) && inner_counter < max_inner_iterations
          % fprintf("Using t=%.8f\n",t);
          t = t/2;
          xnew = t* dx;
          inner_counter = inner_counter + 1;
        end
        x = xnew;
      end
      fvals(i,j) = f(x);
      p_star_vals(i,j) = x(1);
      theta_star_vals(i,j) = x(2);
    else
      disp("INFEASIBLE (p0,theta0)!!");
      fvals(i,j) = inf;
      p_star_vals(i,j) = inf;
      theta_star_vals(i,j) = inf;
    end
  end
end
pivals = -fvals;
suboptimalities = pi_star_builtin - pivals;
X = repmat(p0_vals,ntheta0,1);
Y = repmat(theta0_vals(end:-1:1)',1,np0);
Z = fvals(:,end:-1:1)';
h = pcolor(X,Y,Z);
set(h, 'EdgeColor', 'none');

xlabel("p_0");
ylabel("$\tilde{\theta}_0$","interpreter", "latex");
title("Accurate Convergence Over Initial
($p_0,\tilde{\theta}_0$)","interpreter", "latex");
plot(arrayfun(pmax,theta0_vals),theta0_vals);
xlabel("Maximum Feasible $p$","interpreter","latex");
ylabel("$\tilde{\theta}$","interpreter", "latex");
title("Boundary of Implicit Domain ($p,\tilde{\theta}$)","interpreter",
"latex");
```

5.2 Newton's Method with Logarithmic Barrier Functions Accuracy

```
clear all; close all;
Lbar = 30;
thetabar = 6;
ggamma = 2.25;
c = 5;
H = 35;
ttt = 100;
subfun0 = @(theta) exp(-H/theta);
subfun1 = @(theta) 1-exp(-H/theta);
subfun2 = @(p,theta) 1-p*(theta^2)*(1-exp(-H/theta));
subfun3 = @(p,theta) p*H*exp(-H/theta) - 2*p*theta*(1-exp(-H/theta));
const1 = Lbar/thetabar;
Pi0 = @(p,theta) ttt*(const1*(ggamma*theta + (1/(1-p))*log(subfun2(p,theta))  )
- c*((1-p)/p)) + log(theta) + log(thetabar-theta);
Pi = @(p,theta) ttt*(const1*(ggamma*theta + (1/(1-p))*log(subfun2(p,theta))  )
- c*((1-p)/p)) + log(theta) + log(thetabar-theta) + log(p) + log(1-p);
dPidp = @(p,theta) ttt*(c*((1-p)/(p^2)) + c/p + const1 *(-( (
(theta^2)*subfun1(theta)  )/( (1-p)*(subfun2(p,theta) )  )  )   + (
(1/(1-p)^2) * log(subfun2(p,theta))   ))) +(1/p) - 1/(1-p);
dPidtheta = @(p,theta) ttt*(const1 * (ggamma +     ( (p*H*subfun0(theta)
-2*p*theta*subfun1(theta))/((1-p)*(subfun2(p,theta))     ))) + (1/theta) -
1/(thetabar-theta);
PiGrad = @(p,theta) [dPidp(p,theta);dPidtheta(p,theta)];
dPi2dp2 =@(p,theta) ttt*(-2*c*(1-p)/(p^3) - 2*c/(p^2) + const1*( (-theta^4 *
subfun1(theta))/((1-p)*subfun2(p,theta)^2) -(2*theta^2*subfun1(theta))/((1-p)^2
* subfun2(p,theta)) + 2*log(subfun2(p,theta))/((1-p)^3)          )) - (1/p^2) -
(1/(1-p)^2);
dPi2dtheta2 =@(p,theta)
ttt*(const1*(-subfun3(p,theta)^2/((1-p)*subfun2(p,theta)^2) +
(-2*p*subfun1(theta) + (p*(H^2)/(theta^2) + 2*p*H/theta)*subfun0(theta) )
/((1-p)*subfun2(p,theta)) )) - (1/theta^2) - (1/(thetabar - theta)^2);
dPi2dpdtheta =@(p,theta) ttt*(const1*(((theta^2) *
subfun1(theta)*subfun3(p,theta))/((1-p)*subfun2(p,theta)^2) +
(H*subfun0(theta))/((1-p)*subfun2(p,theta)) -
(2*theta*subfun1(theta))/((1-p)*subfun2(p,theta)) + subfun3(p,theta)/((1-p)^2 *
subfun2(p,theta)) ));
piHess = @(p,theta) [dPi2dp2(p,theta),dPi2dpdtheta(p,theta);dPi2dpdtheta(p,thet⌟
a),dPi2dtheta2(p,theta)];
```

```
f = @(x) -Pi(x(1),x(2));
f0 = @(x) -Pi0(x(1),x(2));
fgrad = @(x) -PiGrad(x(1),x(2));
fHess = @(x) -piHess(x(1),x(2));
final = @(x) -f(x);
newtonstep = @(x) - inv(fHess(x)) * fgrad(x);
lambdasquared = @(x) -fgrad(x)' * newtonstep(x);
pmax = @(theta) 1/((theta^2).*subfun1(theta));
w = @(p,theta) (-1/(theta*(1-p)))*log(subfun2(p,theta));
theta0 = 0.5;
p0 = (1/2)*min(pmax(theta0),1);
x0 = [p0;theta0];
A = [1,0;-1,0;0,1;0,-1];
b=[1;0;thetabar;0];
x_builtin = fmincon(f0,x0,A,b);
p_star_builtin = x_builtin(1);
theta_star_builtin = x_builtin(2);
f_star_builtin = f(x_builtin);
pi_star_builtin = -f(x_builtin);
max_inner_iterations = 15;
max_outer_iterations = 100;
epsilon = .00000001;
t0 = 1;
x = x0;
p0_vals = 0.01:.01:1;
theta0_vals = (thetabar/5):.01:(thetabar/2);
np0 = length(p0_vals);
ntheta0 = length(theta0_vals);
fvals = zeros(np0,ntheta0);
p_star_vals = zeros(np0,ntheta0);
theta_star_vals = zeros(np0,ntheta0);
for i = 1:np0
  for j = 1:ntheta0
    p0 = p0_vals(i);
    theta0 = theta0_vals(j);
    x0 = [p0;theta0];
    x = x0;

    if p0 < pmax(theta0)
      fprintf("Starting Newton: p0=%.2f,theta0=%.2f\n",p0,theta0);
      counter = 0;
      while lambdasquared(x)/2 > epsilon && counter < max_outer_iterations
        counter = counter + 1;
        % fprintf("Newton iteration %d\n",counter);
        dx = newtonstep(x);
        t = t0;
        xnew = x + t* dx;
        inner_counter = 0;
        while f(xnew)> f(x) && inner_counter < max_inner_iterations
          % fprintf("Using t=%.8f\n",t);
          t = t/2;
          xnew = t* dx;
          inner_counter = inner_counter + 1;
        end
        x = xnew;
      end
      fvals(i,j) = f(x);
      p_star_vals(i,j) = x(1);
      theta_star_vals(i,j) = x(2);
    else
      disp("INFEASIBLE (p0,theta0)!!");
      fvals(i,j) = inf;
      p_star_vals(i,j) = inf;
      theta_star_vals(i,j) = inf;
    end
  end
end
pivals = -fvals;
suboptimalities = pi_star_builtin - pivals;
X = repmat(p0_vals,ntheta0,1);
Y = repmat(theta0_vals(end:-1:1)',1,np0);
Z = fvals(:,end:-1:1)';
Z = real(Z);
fig = figure('visible','off');

h = pcolor(X,Y,Z);
set(h, 'EdgeColor', 'none');

xlabel("p_0");
ylabel("$\tilde{\theta}_0$","interpreter", "latex");
```

```
title("Accurate Convergence Over Initial
($p_0,\tilde{\theta}_0$)","interpreter", "latex");
```

5.3 Exhaustive Search Over One Variable with Newton's Method

```
clear all; close all;
options = optimoptions('fmincon','Display','off');
Lbar = 30;
thetabar = 6;
ggamma = 2.25;
c = 5;
H = 35;

eepsilon = .01;

ttt = 100;

theta_vals = eepsilon:eepsilon:thetabar;
nprob = length(theta_vals);
fvals = zeros(1,nprob);
p_star_vals = zeros(1,nprob);
f_builtin_vals = zeros(1,nprob);
p_star_builtin_vals = zeros(1,nprob);

max_linesearch_iterations = 15;
max_outer_iterations = 100;

for i = 1:nprob
  fprintf("ITERATION %d/%d\n",i,nprob);
  theta = theta_vals(i);
  subfun0 = exp(-H/theta);
  subfun1 = 1-exp(-H/theta);
  subfun2 = @(p) 1-p*(theta^2)*(1-exp(-H/theta));
  subfun3 = @(p) p*H*exp(-H/theta) - 2*p*theta*(1-exp(-H/theta));
  const1 = Lbar/thetabar;
  Pi0 = @(p) const1*(ggamma*theta + (1/(1-p))*log(subfun2(p))  ) - c*((1-p)/p);
  Pi = @(p) ttt*(const1*(ggamma*theta + (1/(1-p))*log(subfun2(p))  ) -
  c*((1-p)/p)) + log(p) + log(1-p);
```

```
dPidp = @(p) ttt*(c*((1-p)/(p^2)) + c/p + const1 *(-( ( (theta^2)*subfun1
)/( (1-p)*(subfun2(p) )  )  )   + (  (1/(1-p)^2) * log(subfun2(p))   ))) +
(1/p) - 1/(1-p);
PiGrad = @(p) [dPidp(p)];
dPi2dp2 =@(p) ttt*(-2*c*(1-p)/(p^3) - 2*c/(p^2) + const1*( (-theta^4 *
subfun1)/((1-p)*subfun2(p)^2) -(2*theta^2*subfun1)/((1-p)^2 * subfun2(p)) +
2*log(subfun2(p))/((1-p)^3)))- (1/p^2) - (1/(1-p)^2);
piHess = @(p) [dPi2dp2(p)];



f = @(x) -Pi(x(1));
f0 = @(x) -Pi0(x(1));
fgrad = @(x) -PiGrad(x(1));
fHess = @(x) -piHess(x(1));
final = @(x) -f(x);
newtonstep = @(x) - inv(fHess(x)) * fgrad(x);
lambdasquared = @(x) -fgrad(x)' * newtonstep(x);

pmax = 1/((theta^2).*subfun1);
% w = @(p) (-1/(theta*(1-p)))*log(subfun2(p,theta));

p0 = (1/2)*min(pmax,1);
x0 = [p0];
A = [1;-1];
b=[1;0];
x_builtin = fmincon(f0,x0,A,b,[],[],[],[],[],options);
p_star_builtin = x_builtin(1);
f_star_builtin = f(x_builtin);

f_builtin_vals(i) = f_star_builtin;
p_star_builtin_vals(i) = p_star_builtin;



epsilon = .00000001;
t0 = 1;

x0 = [p0];
x = x0;

% fprintf("Starting Newton: p0=%.2f,theta=%.2f\n",p0,theta);
```

```
  counter = 0;
  while lambdasquared(x)/2 > epsilon && counter < max_outer_iterations
    counter = counter + 1;
    % fprintf("Newton iteration %d\n",counter);
    dx = newtonstep(x);
    t = t0;
    xnew = x + t* dx;
    inner_counter = 0;
    while f(xnew)> f(x) && inner_counter < max_linesearch_iterations
      % fprintf("Using t=%.8f\n",t);
      t = t/2;
      xnew = t* dx;
      inner_counter = inner_counter + 1;
    end
    x = xnew;
  end

  fvals(i) = f(x);
  p_star_vals(i) = x;
end
not_bad_inds = find((real(fvals)>-10^6).*(real(p_star_vals)<=1)); %10-20 values
of theta seem to diverge to -inf!
fvals = fvals(not_bad_inds);
f_builtin_vals = f_builtin_vals(not_bad_inds);
p_star_vals = p_star_vals(not_bad_inds);
p_star_builtin_vals = p_star_builtin_vals(not_bad_inds);
theta_vals = theta_vals(not_bad_inds);


%%%% GLOBAL OPTIMUM BENCHMARK
subfun2_0 = @(p,theta) 1-p*(theta^2)*(1-exp(-H/theta));
Pi00 = @(p,theta) const1*(ggamma*theta + (1/(1-p))*log(subfun2_0(p,theta))  ) -
c*((1-p)/p);
f00 = @(x) -Pi00(x(1),x(2));
pmax0 = @(theta) 1/((theta^2).*subfun1);
theta0 = 0.5;
p0 = (1/2)*min(pmax0(theta0),1);
x0 = [p0;theta0];
A = [1,0;-1,0;0,1;0,-1];
b=[1;0;thetabar;0];
```

```
x_global_opt_builtin = fmincon(f00,x0,A,b);
p_star_global_builtin = x_global_opt_builtin(1);
theta_star_global_builtin = x_global_opt_builtin(2);
f_star_builtin = f0(x_global_opt_builtin);
pi_star_builtin = -f0(x_global_opt_builtin);
%%%% GLOBAL OPTIMUM BENCHMARK

fig = figure('visible','off');

plot(theta_vals,p_star_builtin_vals);
hold on
plot(theta_vals,p_star_vals);
scatter([theta_star_global_builtin],[p_star_global_builtin],100,'filled');
legend("Built-in Optimization fmincon (over p)","Newton's Method with Barrier
(over p)","Global optimum $(\theta,p)$","interpreter", "latex");
xlabel("$\tilde{\theta}$","interpreter", "latex");
ylabel("$p^*(\tilde{\theta})$","interpreter", "latex");
title("Optimal $p$ Value for Each Fixed $\tilde{\theta}$","interpreter",
"latex");

plot(theta_vals,f_builtin_vals);
hold on
plot(theta_vals,fvals);
scatter([theta_star_global_builtin],[f_star_builtin],100,'filled');

legend("Built-in Optimization fmincon (over p)","Newton's Method with Barrier
(over p)","Global optimum $(\theta,p)$","interpreter", "latex");
ylabel("$f^*(\tilde{\theta})$","interpreter", "latex");
xlabel("$\tilde{\theta}$","interpreter", "latex");
title("Optimal Objective Value for Each Fixed $\tilde{\theta}$","interpreter",
"latex");
```

References

[1] Daniel Kahneman and Amos Tversky. Prospect theory: An analysis of decision under risk. *Econometrica*, 47(2):263–291, 1979.

[2] Richard Thaler and Sherwin Rosen. The value of saving a life: Evidence from the labor market. In *Household Production and Consumption*, pages 265–302. National Bureau of Economic Research, Inc, 1976.

[3] A. Zielesny. *From Curve Fitting to Machine Learning: An Illustrative Guide to Scientific Data Analysis and Computational Intelligence.* Intelligent Systems Reference Library. Springer International Publishing, 2016.