Search and Matching - Ph.D. Training Course

Lecture 3: Basics of Search Frictions

Philipp Kircher[1]

[1]University of Edinburgh

December 6th 2013

Search Frictions: The Basics

Why search frictions (also called matching frictons):

- competitive model: no voluntary unemployment, no co-existence of unfilled vacancies and unemployed individuals
- search frictions: one way to introduce both.
- idea: it takes time to find a parter (the right partner).

Search Frictions: The Basics

The basic model elements:

- number v of vacancies (possibly entry cost K)
- number u of workers searching for these vacancies
- matching function $M(v, u)$ gives the number of matches
  (often calibrated to be Beveridge curve; discuss recent events)
  str incr and str concave in each argument, overall constant returns

Search Frictions: The Basics

The basic model elements:

- number v of vacancies (possibly entry cost K)
- number u of workers searching for these vacancies
- matching function $M(v, u)$ gives the number of matches
  (often calibrated to be Beveridge curve; discuss recent events)
  str incr and str concave in each argument, overall constant returns
- queue length or market tightness: $\lambda = u/v$
- vacancy filling prob: $m(\lambda) = M(1, \lambda) = M(v, u)/v$
- worker matching prob: $n(\lambda) = m(\lambda)/\lambda = M(1, \lambda)/\lambda = M(u, v)/u$

Search Frictions: The Basics

The basic model elements:

- number v of vacancies (possibly entry cost K)
- number u of workers searching for these vacancies
- matching function $M(v, u)$ gives the number of matches (often calibrated to be Beveridge curve; discuss recent events) str incr and str concave in each argument, overall constant returns
- queue length or market tightness: $\lambda = u/v$
- vacancy filling prob: $m(\lambda) = M(1, \lambda) = M(v, u)/v$
- worker matching prob: $n(\lambda) = m(\lambda)/\lambda = M(1, \lambda)/\lambda = M(u, v)/u$
- wage w is determined. (explanation follows)

Search Frictions: The Basics

The basic model elements:

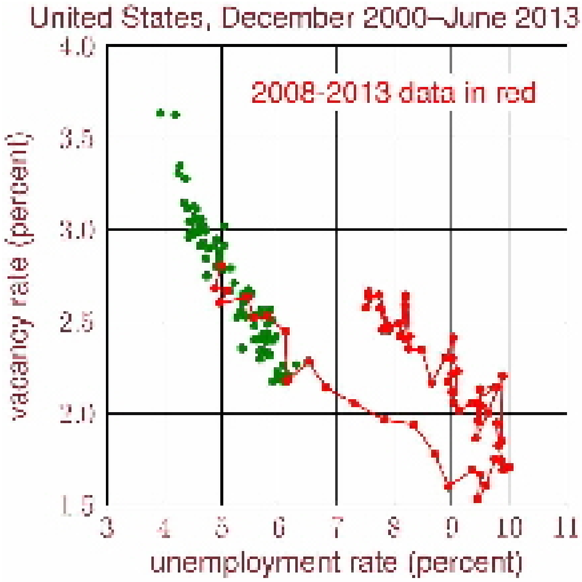
- number v of vacancies (possibly entry cost K)
- number u of workers searching for these vacancies
- matching function $M(v, u)$ gives the number of matches (often calibrated to be Beveridge curve; discuss recent events) str incr and str concave in each argument, overall constant returns
- queue length or market tightness: $\lambda = u/v$
- vacancy filling prob: $m(\lambda) = M(1, \lambda) = M(v, u)/v$
- worker matching prob: $n(\lambda) = m(\lambda)/\lambda = M(1, \lambda)/\lambda = M(u, v)/u$
- wage w is determined. (explanation follows)
- succesful: firm gets $y - w$ and worker gets w (risk-neutrality).
- Some prob $s \geq 0$ that job survives to next period.

Search Frictions: The Basics

The basic model elements:

- number v of vacancies (possibly entry cost K)
- number u of workers searching for these vacancies
- matching function $M(v, u)$ gives the number of matches (often calibrated to be Beveridge curve; discuss recent events) str incr and str concave in each argument, overall constant returns
- queue length or market tightness: $\lambda = u/v$
- vacancy filling prob: $m(\lambda) = M(1, \lambda) = M(v, u)/v$
- worker matching prob: $n(\lambda) = m(\lambda)/\lambda = M(1, \lambda)/\lambda = M(u, v)/u$
- wage w is determined. (explanation follows)
- succesful: firm gets $y - w$ and worker gets w (risk-neutrality).
- Some prob $s \geq 0$ that job survives to next period.
- unsuccessful: workers unemployment payoff $b \geq 0$, firms get 0.
- Potentially try next period again (discount $\delta \in [0, 1)$).
- For now mostly static ($\delta = 0$ or $s = 0$) and rep agent.

Beveridge Curve



Wage Determination Mechanisms

Random search and bilateral meetings: surplus to be split

- firms make take-it-or-leave-it wage offer
  - Diamond Paradox
  - Wage $=$ reservation value $= b$
  - Workers have no incentive to search

Wage Determination Mechanisms

Random search and bilateral meetings: surplus to be split

- firms make take-it-or-leave-it wage offer
  - Diamond Paradox
  - Wage = reservation value = b
  - Workers have no incentive to search
- "Fix" against Diamond Paradox
- Bargaining where workers have power β
  - Mortensen, Pissarides
  - Wage in the interior unless $\beta = 0$ or $\beta = 1$

Wage Determination Mechanisms

Random search and bilateral meetings: surplus to be split

- firms make take-it-or-leave-it wage offer
  - Diamond Paradox
  - Wage = reservation value = b
  - Workers have no incentive to search
- "Fix" against Diamond Paradox
- Bargaining where workers have power β
  - Mortensen, Pissarides
  - Wage in the interior unless $\beta = 0$ or $\beta = 1$
- (Worker have unobserved outside option, Albrecht-Axel))

Non-random search:

- Some worker see more than one wage offer
  - some randomly see one offer and others more (Butters, Burdett-Judd, Varian,...)
  - on-the-job search (Burdett-Mortensen)
  - directed search: see every offer but can apply only to one (Peters, Burdett-Shi-Wright, Moen, Shimer, Kircher...)

  Here: discuss bargaining and directed search ($\delta = 0$).

The efficient benchmark

The efficient benchmark (Hosios 1990):

$$
\begin{aligned}
& \max_v M(u,v)y + (u - M(u,v))b - vK \\
\Leftrightarrow \quad & \max_v vm(u/v)(y-b) + ub - vK
\end{aligned}
$$

$$
\begin{aligned}
\Rightarrow \quad & [m(\lambda^*) - \lambda^* m'(\lambda^*)](y-b) = K; \quad \lambda = u/v \\
\Leftrightarrow \quad & m(\lambda^*)\left[1 - \lambda^* m'(\lambda^*)/m(\lambda^*)\right](y-b) = K.
\end{aligned}
$$

- Conditional on matching, workers have to get a fraction of output equal to the elasticity of the matching function.

The efficient benchmark

The efficient benchmark (Hosios 1990):

$$\begin{aligned} & \max_v M(u,v)y + (u - M(u,v))b - vK \\ \Leftrightarrow \quad & \max_v vm(u/v)(y-b) + ub - vK \end{aligned}$$

$$\begin{aligned} \Rightarrow \quad & [m(\lambda^*) - \lambda^* m'(\lambda^*)](y-b) = K; \quad \lambda = u/v \\ \Leftrightarrow \quad & m(\lambda^*)\left[1 - \lambda^* m'(\lambda^*)/m(\lambda^*)\right](y-b) = K. \end{aligned}$$

- Conditional on matching, workers have to get a fraction of output equal to the elasticity of the matching function.
- Insight applies even when $\delta > 1$, i.e., when the future matters.

The efficient benchmark

The efficient benchmark (Hosios 1990):

$$
\begin{aligned}
& \max_{v} M(u,v)y + (u - M(u,v))b - vK \\
\Leftrightarrow \quad & \max_{v} vm(u/v)(y-b) + ub - vK \\
\\
\Rightarrow \quad & [m(\lambda^*) - \lambda^* m'(\lambda^*)](y-b) = K; \quad \lambda = u/v \\
\Leftrightarrow \quad & m(\lambda^*)\left[1 - \lambda^* m'(\lambda^*)/m(\lambda^*)\right](y-b) = K.
\end{aligned}
$$

- Conditional on matching, workers have to get a fraction of output equal to the elasticity of the matching function.
- Insight applies even when $\delta > 1$, i.e., when the future matters.
- Comment: Urnball $m(\lambda) = 1 - e^{-\lambda}$ generates condition $(1 - e^{-\lambda} - \lambda e^{-\lambda})(y-b) = K$, so firm gets paid everything if two or more workers are there, while worker gets everything when alone (which is exactly when he brings surplus).

Bargaining

Bargaining: With $\delta = 0$ workers get $b + \beta(y - b)$ and firms get $(1 - \beta)y$.

Entry:

$$m(\lambda)[1 - \beta](y - b) = K$$

Efficient v (i.e., efficient $\lambda = u/v$) iff $\beta =$ elasticity of m. Generically inefficient.

Directed Search (Competitive Search)

Each firm simultaneously publicly posts wage w, then workers decide where to search. (Foundations: Peters, Burdett-Shi-Wright, Galenianos-Kircher)

Firm maximizes: $\max_w m(\lambda(w))(y-w) \qquad (= K$ due to free entry)

Workers must be indifferent: $n(\lambda(w))w + (1 - n(\lambda(w))b = U$

Since $w = \lambda(U-b)/m(\lambda) + b$, firm maximizes:

$$\begin{aligned} & \max_\lambda m(\lambda)(y-b) - \lambda(U-b) \qquad (= K) \\ \Rightarrow \quad & m'(\lambda)(y-b) = U - b \\ \Rightarrow \quad & m(\lambda)[1 - \lambda m'(\lambda)/m(\lambda)](y-b) = K \end{aligned}$$

Entry is efficient. Bargaining Power as in the Hosios condition.

Is Directed Search Always Efficient?

Is directed search always (constrained) efficient:

- The good that is being priced is the matching probability
- If there are enough "prices" then it is efficient
- It is as efficient as the planner if the planner does not have more instruments.
- Discuss:
  - risk-averse workers (utility $u(w)$), no savings
  - risk-neutral firms, can only pay employed workers
  - planner with and without unemployment benefits

Particular Usefulness of Directed Search: Simplicity

"block-recursivity" (Menzio-Shi,... Kaas-Kircher):

- Heterogeneous workers: Let $G(b)$ be number of workers with type below b
- Future matters: discounting ($\delta \in (0,1)$), and separations ($s \in (0,1)$)
- Aggregate shocks: $y \in \{y_l, y_h\}$ with transitions $\pi_{ll}, \pi_{lh}, \pi_{hl}, \pi_{hh}$

Particular Usefulness of Directed Search: Simplicity

"block-recursivity" (Menzio-Shi,... Kaas-Kircher):

- Heterogeneous workers: Let $G(b)$ be number of workers with type below b
- Future matters: discounting ($\delta \in (0,1)$), and separations ($s \in (0,1)$)
- Aggregate shocks: $y \in \{y_l, y_h\}$ with transitions $\pi_{ll}, \pi_{lh}, \pi_{hl}, \pi_{hh}$
- Let $\Upsilon_t(b)$ be number of unemployed with types below b:

$$\Upsilon_{t+1}(b) = \int_0^b [1 - n(\lambda_t(w(b)))] d\Upsilon_t(b) + (1-s)[G(b) - \Upsilon_t(b)]$$

- This object moves around over the cycle. Infinite-dimensional state space. Hard to find $\lambda_t(w)$.

Particular Usefulness of Directed Search: Simplicity

"block-recursivity" (Menzio-Shi,... Kaas-Kircher):

- Heterogeneous workers: Let $G(b)$ be number of workers with type below b
- Future matters: discounting ($\delta \in (0,1)$), and separations ($s \in (0,1)$)
- Aggregate shocks: $y \in \{y_l, y_h\}$ with transitions $\pi_{ll}, \pi_{lh}, \pi_{hl}, \pi_{hh}$
- Let $\Upsilon_t(b)$ be number of unemployed with types below b:

$$\Upsilon_{t+1}(b) = \int_0^b [1 - n(\lambda_t(w(b)))] d\Upsilon_t(b) + (1-s)[G(b) - \Upsilon_t(b)]$$

- This object moves around over the cycle. Infinite-dimensional state space. Hard to find $\lambda_t(w)$.
- Relevant with on-the-job search (where b is value of current job).
- Solution: Free entry: $\lambda_t(w)$ only depends on y_t! :

$$m(\lambda_t(w))[V(y_t) - W(w)] = K$$

  where $W(w(b)) = w(b)/(1 - s\delta)$, $V(y) = y + s\delta E(V(y'))$.
- So $\lambda_t(w)$ determined without knowing Υ!!! Only current y_t matters.

Applications

Applications

- On-the-job search over the business cycle (Shi, Menzio-Shi)
- "Recall" of unemployed (Fernando Blanco)
- Output-enhancing unemployment insurance (Acemoglu-Shimer)
- Modelling large firms with output $f(y, L)$, where $L = vm(\lambda)$ (Kaas-Kircher, Schaal).
- Applications of large firms to international trade
- Introduction of two-sided heterogeneity: $f(x, y, L)$, where $L = vm(\lambda)$ (Eeckhout-Kircher, Grossman-Helpman-Kircher)
- ....

Multiple Job Applications: NTU matching...

Multiple Job Applications:

- Assume same $b = 0$ for all workers. One period ($\delta = s = 0$).
- Workers can apply for TWO jobs instead of only to one. (More general: N job applications, possibly determined by application cost).
- Big difference:
  - one application: if a firm offers the job to a worker, he accepts
  - two applications: if a firm offers its job to a worker, but the worker got a better other offer, he declines. What does the firm do then?
    Albrecht-Gautier-Vroman: Bid wage up for this worker (access entry)
    Galianos-Kircher: Too bad, firm gets no worker even if other initially applied (inefficient application behavior)
    Kircher: Firm offers the job to the next worker, and workers can change their mind (Gale-Shapley..., Leads to overall efficient behavior even though there is ONLY one price and two applications).
    Wolthoff: synthesis. Gautier-Holzer: TU matching.

Multiple Job Applications: NTU matching...

Kircher's set-up close to medical resident intern matching:

- Hospitals set compensation in advange
- Interns apply to a limited number of jobs (see Roth... simulations)
- Hospitals tend to shortlist only those they interviewed
- Final matching is Gale-Shapley

2 applications: worker behavior

With 2 applications, the workers takes the probability $p(w)$ of getting a job as given, and maximizes:

$$\max_{w_1,w_2} p(w_2)w_2 + (1 - p(w_2))p(w_1)w_1$$

Note that w_2 is only choosen if w_1 unsuccesful, two problems:

$$\max_{w_1} p(w_1)w_1 \equiv U_1 \quad (1)$$

$$\max_{w_2} p(w_2)w_2 + (1 - p(w_2))U_1 \equiv U_2 \quad (2)$$

- (1) looks like the standard worker problem with $b = 0$ (but some workers will walk away).
- (2) looks like the standard worker problem with $b = U_1$ (but all workers stay).‘
- Two wages arise in equilibrium (needs more time to explain, see paper and Gautier-Holzner).

2 applications: Firms behavior

High wage firms max:

$$\max_{w_2} \left(1 - e^{-\lambda(w_2)}\right)(y - w_2) \text{ s.t. } \underbrace{\frac{1 - e^{-\lambda(w_2)}}{\lambda(w_2)}}_{p(w_2)}(w_2 - U_1) = U_2 - U_1.$$

Low wage firms max:

$$\max_{w_1} \left(1 - e^{-(1-p(w_2))\lambda(w_1)}\right)(y - w_2) \text{ s.t. } \underbrace{\frac{1 - e^{-(1-p(w_2))\lambda(w_2)}}{(1 - p(w_2))\lambda(w_2)}}_{p(w_1)} w_1 = U_1.$$

Note that workers and firms care about the same "effective queue length". In problem 2, let $\mu(w_1) = (1 - p(w_2))\lambda(w_1)$. Substitute out and solve as in one-application case. Then use U_1 in the first problem. Let's you determine how many firms offer each wage. This is socially efficient.

- Application: Medical intern matching. Can anlyze wages, matching probabilities, number of applications, too low wages (finite numbers? see Hatfield... AER), but needs heterogeneity...