A Simulation of Bilateral Migration Using Markov Chain Monte Carlo

Charley Ferrari, Paul Garaud, Dieudonne Ouedraogo

December 13, 2015

In this paper, we build on Pan and Nagurney's Markov Chain migration model and create a Markov Chain Monte Carlo model for migration. We continue to view migration as a chained phenomenon, represented by a random walk through possible population distributions rather than a deterministic markov chain. These simulations can show us the dynamics of reaching equilibrium, and the sensitivity of equilibrium and convergence to initial populations, costs, and utility.

Introduction

Markov Chains have a long history of being used as theoretical models for studying human mobility. Pan and Nagurney (Pan, Nagurney 1994) provide a useful survey of past attempts at modeling migration, and advance the thinking by allowing for the possibility of non-homogenous markov chains. In their model and review of the literature, geographical locations are seen as possible states of the chain. Previous attempts at modeling population involved a single transition matrix that could be calculated. Pan and Nagurney, by introducing costs, allow for the possibility of a nonhomogenous markov chain that still tends towards equilibrium.

Pan and Nagurney's models can give insight into the dynamics of population mobility, but are limited by their deterministic nature. Costs can influence the time it takes to reach equilibrium and represent imperfect information or time lags associated with moving decisions in the real world, but once equilibrium is reached population is assumed to remain static. The authors point out that equilibrium does not necessarily mean there is no more migration, but simply that the probabilities of moving between locations have equalized, and net migration remains at zero.

In this paper we provide an alternative way of looking at a population model using markov chain monte carlo. By introducing randomness into the model, we hope to gain some insight into how exactly our population will converge to equilibrium, and what sort of variation will occur once equilibrium is acheived. In this way, we hope to model both migration induced by comparative utilities across regions, and movement due to random factors.

The choice of markov chain monte carlo led to some necessary departures from Pan and Nagurney's model. Each iteration of Pan and Nagurney's markov chain represents an equilibrium between comparative utilities and the cost of moving, and this equilibrium is reached by considering a flow of migrants. The iterations in our model do not represent equilibrium, but are rather defined steps towards an equilibrium population distribution. Each step represents the possibility of one person moving. Because of this, cost is not dependent on a flow of people, but is rather assumed to be a static, dependent only on the source and destination.

Pan and Nagurney look at network equilibrium models as one-step models. They arrive at equilibrium, but do not allow one to view chain migration nor the dynamics of migration. By using markov chain monte carlo, introducing randomness allows us another way to view these features of a migration model.

Literature Review

Our model is based directly off of the Pan and Nagurney model, repurposed for use with markov chain monte carlo.

The authors define utility functions, u_i for each location i, which are dependent on the populations of each region. Costs are a function of the flow between two locations, so each bilateral pair i and j has its own cost c_{ij} .

Starting from an initial population p_0 , the flow between locations i and j can be discovered by solving the below equation:

$$u_i(p_i^1) + c_{ij}(f_{ij}^1) = u_j(p_j^1)$$

For each i and j.

The two region example shown in Pan and Nagurney's paper assumed utility functions of $u_1(p) = -p_1 + 8$ and $u_2(p) = -p_2 + 14$. Costs were simply based on flow, so $c_{12}(f_{12}) = 2f_{12}$ and $c_{21}(f_{21}) = 2f_{21}$.

The solution to this toy model is explained and diagrammed below:

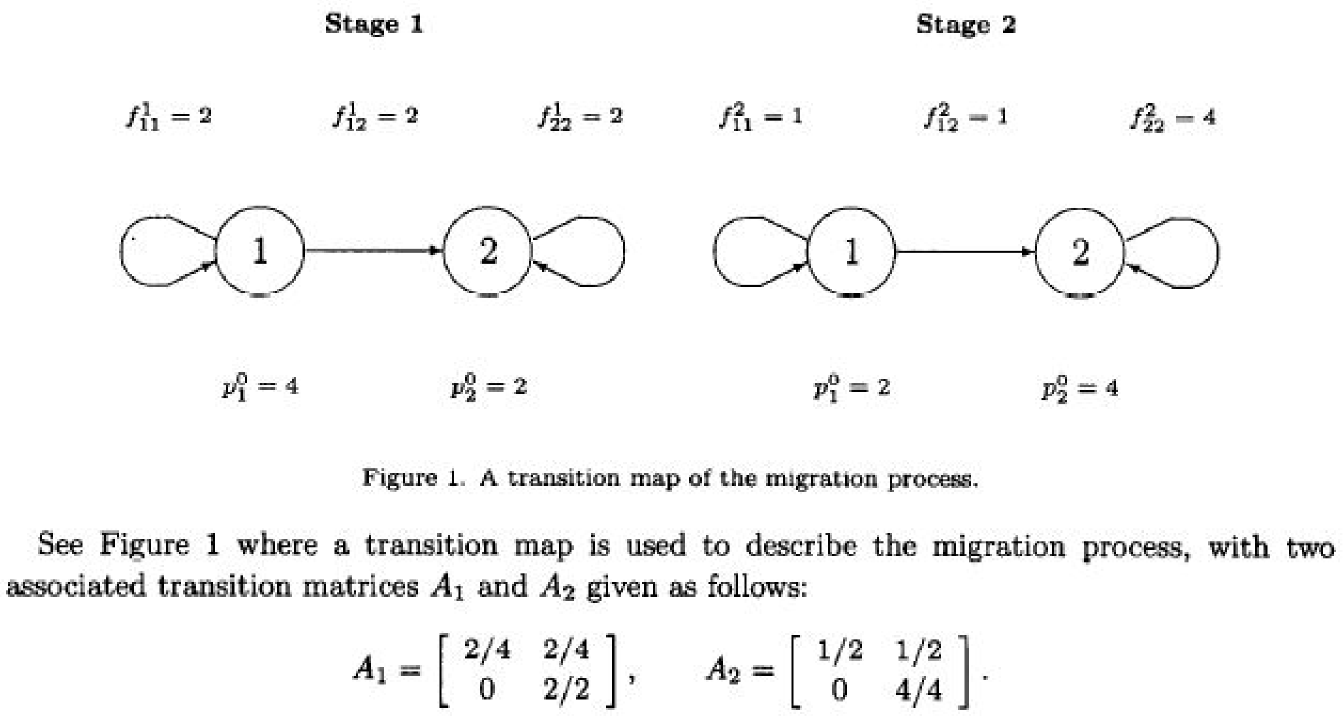


Figure 1. A transition map of the migration process.

See Figure 1 where a transition map is used to describe the migration process, with two associated transition matrices A_1 and A_2 given as follows:

$$A_1 = \begin{bmatrix} 2/4 & 2/4 \\ 0 & 2/2 \end{bmatrix}, \qquad A_2 = \begin{bmatrix} 1/2 & 1/2 \\ 0 & 4/4 \end{bmatrix}.$$

In this case, the transition matrix is the same (and the derivation shows in what way they will differ.)

Pan and Nagurney reference an earlier paper written by Nagurney: A Network Model of Migration Equilibrium with Movement Cost (Nagurney 1990). In this paper, Nagurney provides a very similar model to population, but does not express it in terms of markov chains. Costs are assumed to be fixed, rather than based on flow. Utility functions are similar to the ones represented in the markov chain model, and are formally defined as $u_i = -a_i p_i + b_i$.

Nagurney shows that the question of optimal flows and equilibrium populations is solveable using quadratic programming. Given utility defined above, and costs defined as c_{ij}, the problem becomes:

Minimize

$$\sum_i \frac{1}{2} a_i (\sum_{k \neq 1} f_{ki} - \sum_{k \neq 1} f_{ik} + \bar{p_i})^2 - \sum_i b_i (\sum_{k \neq 1} f_{ki} - \sum_{k \neq 1} fik + \bar{p_i}) + \sum_{i,k \neq i} c_{ik} f_{ik}$$

Subject to

$$\sum_k f_{ik} \leq \bar{p_i}, \forall i$$

and

$$f_{ik} \geq 0, \forall i$$

After defining the problem, Nagurney suggests an algorithm to solve it. Problems like this were solved step by step and numerically using markov chains when they were introduced to the model.

Other attempts at explaining migration using markov chains include A. Constant's and K. Zimmerman's paper "The Dynamics of Repeat Migration: A Markov Chain Analysis." (Constant, Zimmerman 2012.) Using a Markov Chain approach to model repeat migration, this paper aims to describe the dynamics of repeated exit and entry between two countries. The paper estimates the transition probabilities using the formula $P(E_{t+1} = i|E_t = j) = \frac{e^{\beta} ij^x mt}{\sum e^{\beta} ik^x mt}$ using empirical data from Germany. These irepresent the coefficients on the personal characteristics variables for transition state i (note that only two transition state coefficient vectors are calculated as the other two are complements and calculated by taking 1 - P) and are estimated using logistic regression.

The paper then converts micro level longitudinal migration data at the individual level to person-years, which are treated as separate observations, to estimate these conditional probabilities. The end result are empirically-derived formulae that describe the personal characteristics that influence the decision to (repeat) migrate.

This approach is an interesting twist on the traditional Markov Chain analysis by detailing a particular case and integrating real world information about migrants. This could be useful for future considerations of our model as a guide to link real world data to our currently theoretical utility functions.

Methodology

Our model takes utilities and costs and models an individual's decision to move as a step in a markov chain monte carlo algorithm.

Considering n locations, in its broadest terms the algorithm works as follows:

- Pick a source location, i, sampling from all locations with non-zero populations with equal weights.
- Pick a destination location, j, sampling from all locations (except for i) with equal weights (an individual may move to a location with a current population of zero.)
- Compare $u_i(p) + c_{ij}$ to $u_j(p)$
- If $u_i(p) + c_{ij} \geq u_j(p)$, accept the move with probability 1.
- If $u_i(p) + c_{ij} < u_j(p)$, accept the move with some probability.

The first major departure needed to adopt Pan and Nagurney's model to MCMC is a subtle one. Given n locations, Pan and Nagurney's markov chain has n states. In our model, each possible population distribution is a state of our markov chain, and we're randomly walking around this distribution space. This state can be described as the total number of population vectors $p = [p_1, p_2, p_3, \ldots, p_n]$ with a total population t such that $\sum_{i=1}^{n} p = t$.

The general form of the probability a of accepting a state change to proposal state x' drawn from probability distribution $Q(p)$ from state x_t is:

$$a = \frac{P(x')}{P(x_t)} \frac{Q(x_t|x')}{Q(x'|x_t)}$$

$\frac{P(x')}{P(x_t)}$ is calculated by comparing the utility $u_i(p)$ of staying in location i with the utility $u_j(p) - c_{ij}$ of moving. So:

$$\frac{P(x')}{P(x_t)} = \frac{u_j(p) - c_{ij}}{u_i(p)}$$

The second ratio is a test to see if the proposal distribution is symmetric. The proposal distribution is based on the distribution space defined above. By picking a random source and random destination, we are limiting ourselves to the number of states acheivable by moving one person from one location to another. Given n locations, and $n_{p\neq0}$ locations with non-zero populations, and the current popuation distribution state x_t, the proposal distribution $Q(x'|x_t)$ includes these $n_{p\neq0}(n-1)$ states weighted equally, with zero probability of reaching any other state.

In most cases, our proposal distribution is symmetric. If we're considering a move between locations i and j, representing a change in state from x_t to x', and locations i and j are both non-zero, the number of states accessible from either x_t or x' is simply $n_{p\neq0}(n-1)$. The probability $Q(x_t|x') = Q(x'|x_t) = \frac{1}{n_{p\neq0}(n-1)}$.

The only state changes of x_t to x' that aren't symmetric are x_t's where the populations $p_i = 1$ or $p_j = 0$. In these cases, the number of locations with non-zero populations change, for obvious reasons, so $Q(x_t|x') \neq Q(x'|x_t)$. Interestingly, if $p_i = 1$ and $p_j = 0$, the situation remains symmetric.

This means our final probability a of accepting a move from location i to j, representing a state change from state x_t to state x' is:

$$a = \frac{u_i(p)}{u_j(p) - c_{ij}} \frac{1/(n_{p\neq 0,x_t}(n-1))}{1/(n_{p\neq 0,x'}(n-1))}$$

Our code can be found in Appendix A. Combining our general algorithm above with the theory, the probability with which we accept a move is calculated as a. If a is greater than one, we accept the move. If a is less than one, then a represents the probability of accepting the move.

Model dynamics

In this section, model dynamics will be examined. The first subsection will build a base model to use as a benchmark and examine what steady state populations these initial parameters generate. Subsequent subsections will examine various parameter perturbations against this baseline model in order to explore the dynamics of the model.

Base model

The base model relies upon arbitrarily chosen parameter values.

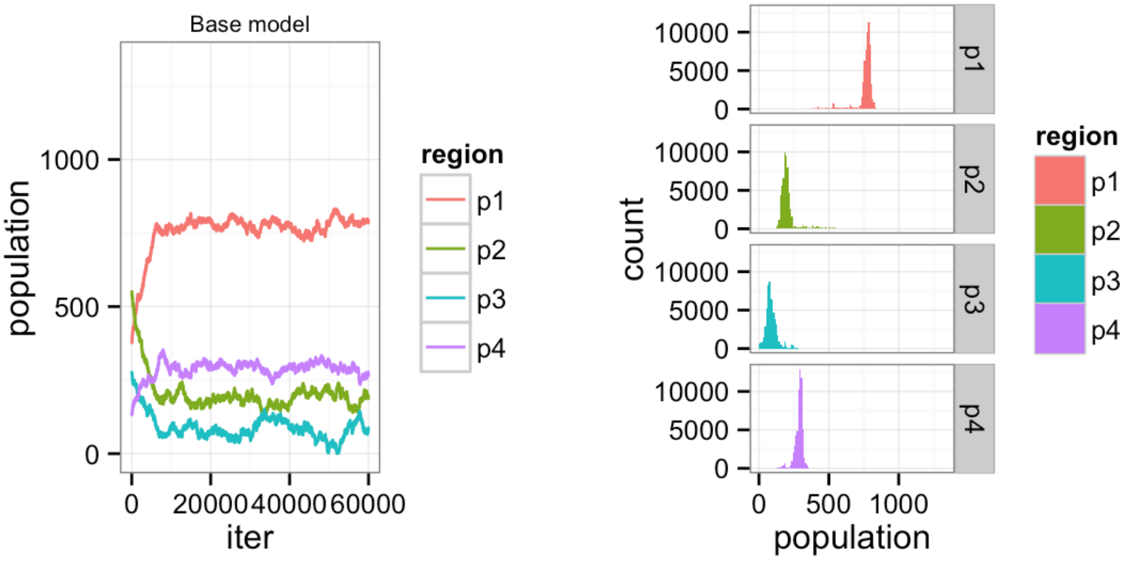
```
source('sensitivity_analysis.R')

params <- reset.params()
params$c  # matrix of costs for moving from region i (row) to j (col)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    0  200  400  600
## [2,]  200    0  200  400
## [3,]  400  200    0  200
## [4,]  600  400  200    0
```

Initial utility is:

```
params$k - params$p
```

```
## [1] 2021 1253 1426 1765
```

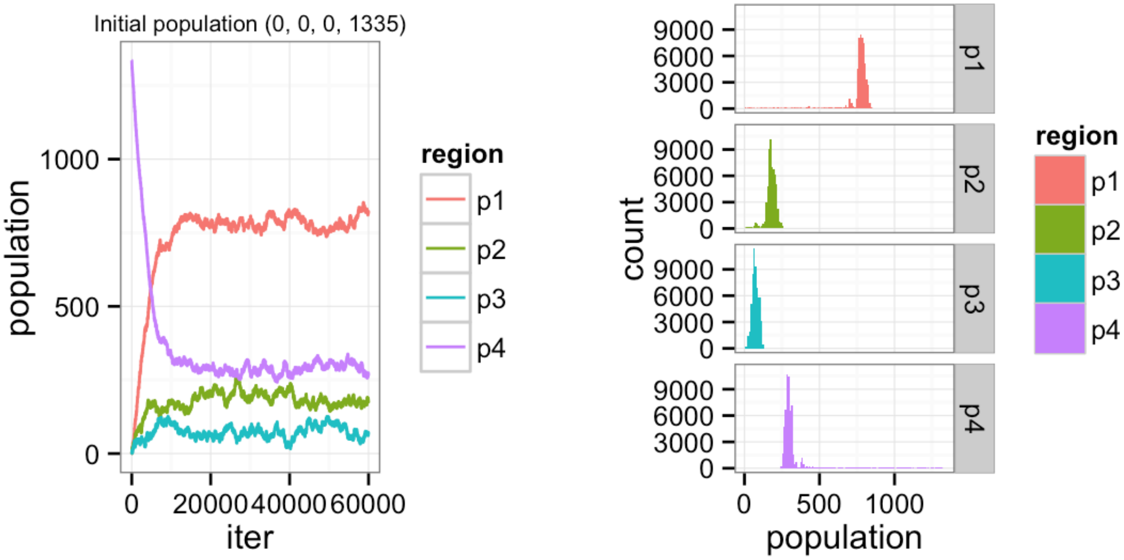


```
##   region  pop.mean ss.utility change.in.utility
##       p1 777.53998   1622.460         -398.5400
##       p2 186.92806   1613.072          360.0719
##       p3  79.30948   1620.691          194.6905
##       p4 291.22248   1608.778         -156.2225
```

We see that the populations converge after roughly 10,000 iterations, with relative stability across the different regions beyond that point. In the case of region 1, the population appears to converge to a steady state population of roughly 800. Region 2, 3, and 4 converge to equilibrium populations of roughly 200, 100, and 300, respectively. Steady state utilities across regions do not vary by much.

Initial conditions

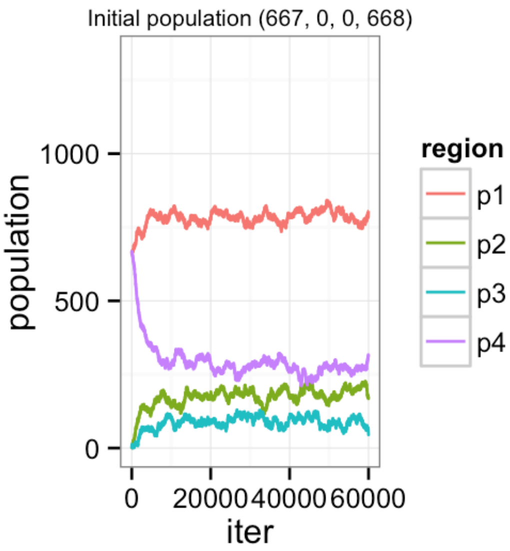
Now that a base model has been run and examined, it is possible to explore whether its results are robust to changes in initial conditions. In the base model, the initial populations are 379, 547, 274, 135 for regions 1-4, respectively. Suppose the initial populations are instead, 0, 0, 0, 1335.

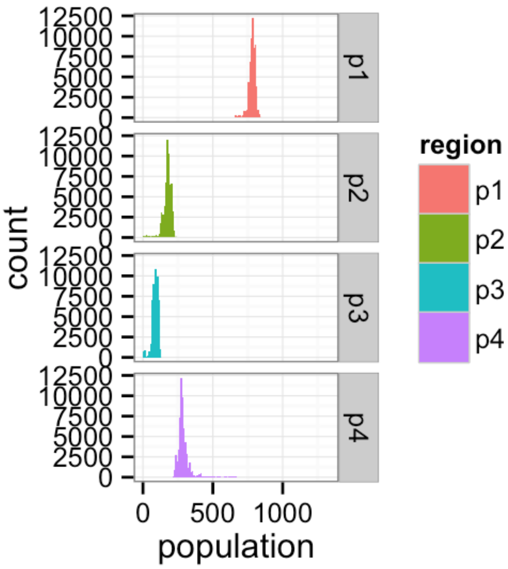


```
##   region  pop.mean ss.utility change.in.utility
##       p1 784.23856   1615.761         -405.2386
##       p2 186.81578   1613.184          360.1842
##       p3  73.26002   1626.740          200.7400
##       p4 290.68564   1609.314         -155.6856
```

The steady state populations and utilities are remarkably similar to the base model.

Suppose the population is split evenly between regions 1 and 4.





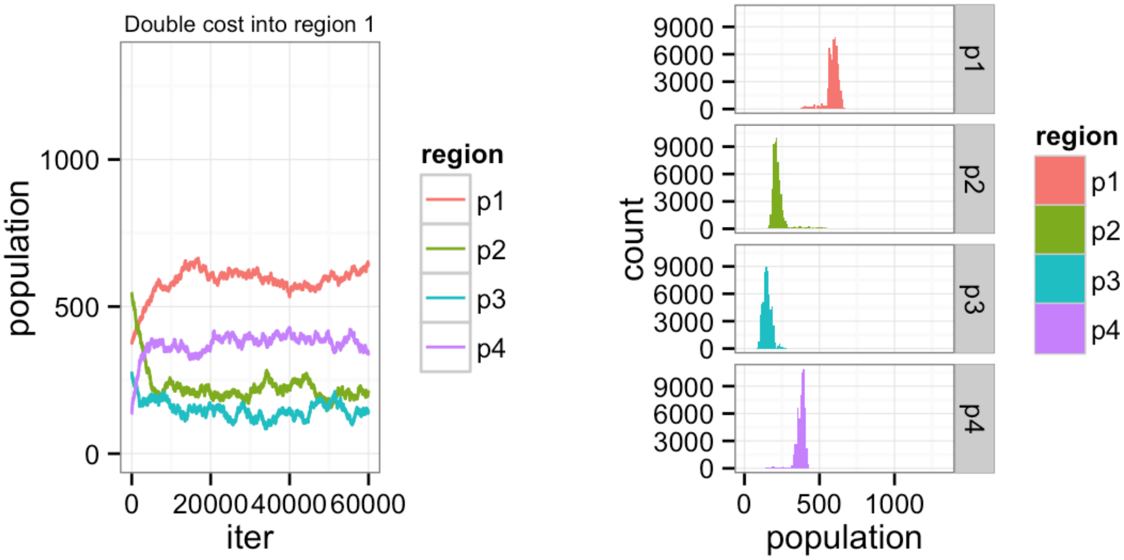
```
##   region pop.mean ss.utility change.in.utility
##       p1 785.3548   1614.645         -118.3548
##       p2 181.1429   1618.857         -181.1429
##       p3  91.3342   1608.666          -91.3342
##       p4 277.1681   1622.832          390.8319
```

Once again, the population in each region quickly converges to the same steady state values as in the base model. We conclude that the steady state values are stable.

Cost of moving

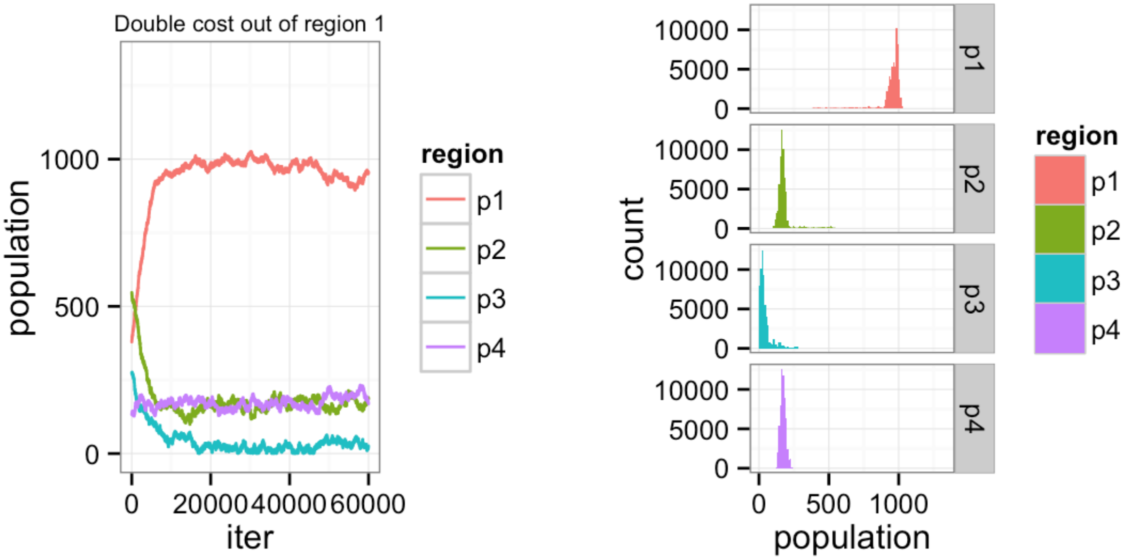
Since the decision to move from one region to the other is a function of the utilities at both the source and destination regions, and cost, changing the cost of moving should directly impact the steady state populations. Specifically, the steady state population of a region should increase with cost out of that region and decrease with cost into that region.

If costs are increased for all moves into a region, fewer people will move into that region in steady state since the increased cost will effectively eat into the utility differential between the two regions. Simulation confirms this intuition.



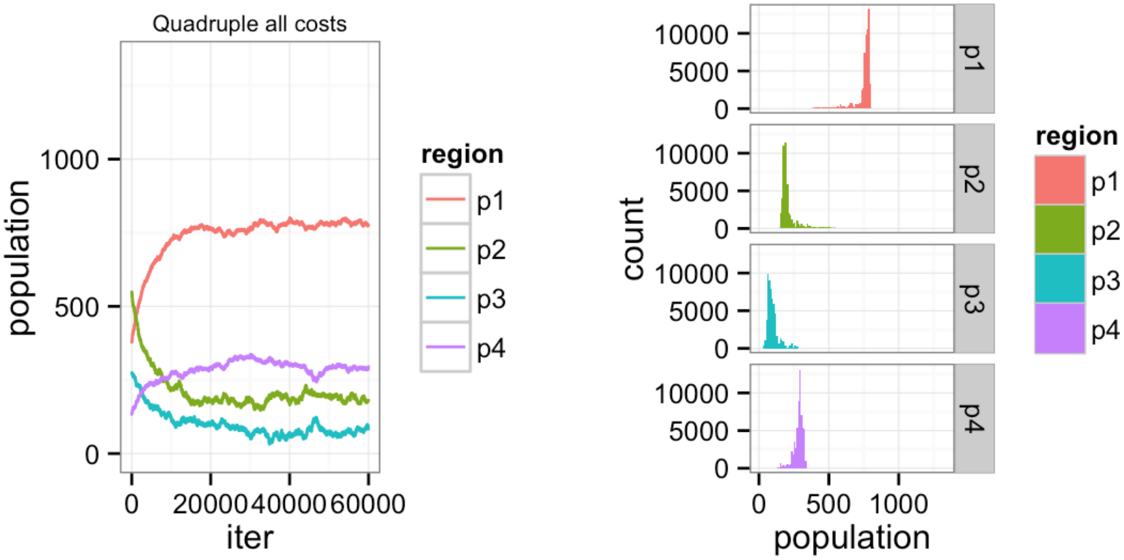
```
##   region pop.mean ss.utility change.in.utility
##       p1 600.5234   1799.477         -221.5234
##       p2 213.3615   1586.639          333.6385
##       p3 141.0889   1558.911          132.9111
##       p4 380.0262   1519.974         -245.0262
```

A similar reasoning applies to costs for moving out a region, implying that fewer people will move out in equilibrium. Once again, simulation is in line with expectations.

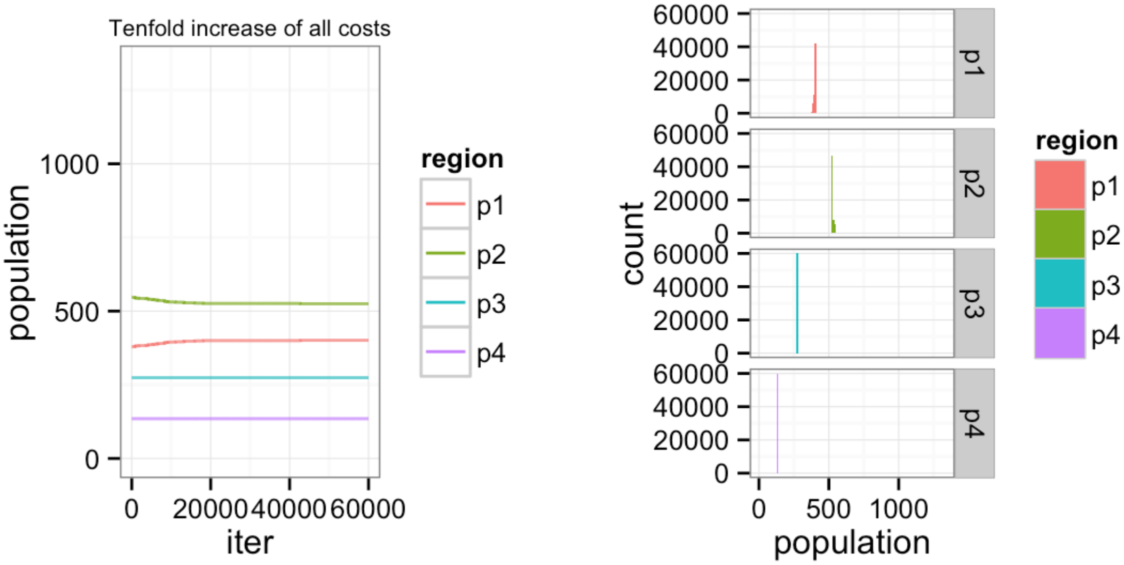


```
##   region pop.mean ss.utility change.in.utility
##       p1 970.7902   1429.210        -591.79024
##       p2 163.5344   1636.466         383.46556
##       p3  26.9619   1673.038         247.03810
##       p4 173.7134   1726.287         -38.71342
```

An interesting case is when costs are increased proportionally across all pairwise moves. We would expect that if costs are high enough, no moves will take place and the steady state will be the initial populations. For smaller increases, it is less clear: relative attractiveness among regions remains unchanged from the base case, but how steady state populations are affected is not immediately evident.



```
##   region  pop.mean ss.utility change.in.utility
##       p1 769.79312   1630.207         -390.7931
##       p2 187.42262   1612.577          359.5774
##       p3  82.27148   1617.729          191.7285
##       p4 295.51278   1604.487         -160.5128
```

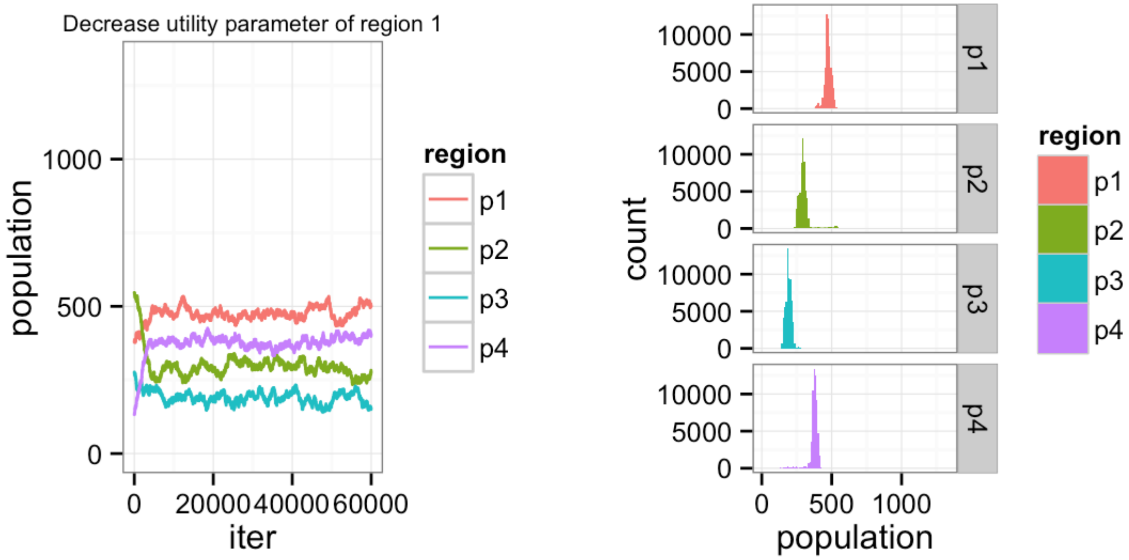


It appears that when costs quadruple, steady state populations do not change noticeably. Convergence appears to suffer, as the acceptance condition in the Metropolis-Hastings algorithm rejects more proposed moves. With a tenfold proportional increase in costs, we see hardly any movement from the starting conditions, as we would expect when costs are extremely high.

Utility

The base model computes the utility in the form of $u_i = k_i - p_i$, where $k_1 = 2400$, $k_2 = 1800$, $k_3 = 1700$, and $k_4 = 1900$.

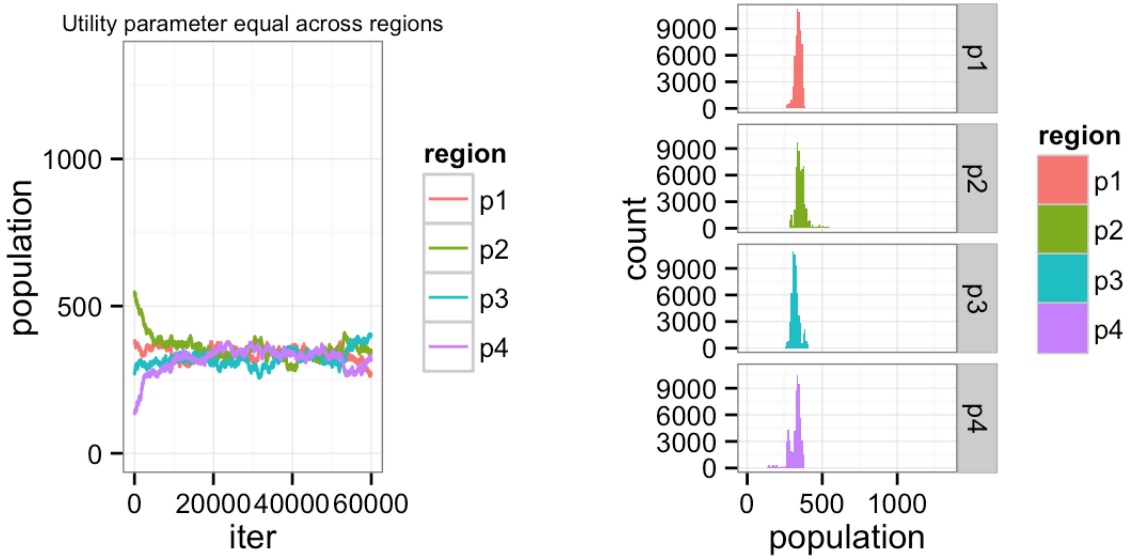
Decreasing k_1 to 2000, we would expect the steady state population of region 1 to decrease and the populations of region 2, 3, and 4 to increase.



```
##   region pop.mean ss.utility change.in.utility
##       p1 476.9453   1523.055         -97.94530
##       p2 290.8278   1509.172         256.17218
##       p3 187.1787   1512.821          86.82126
##       p4 380.0481   1519.952        -245.04814
```

As expected, the steady state population of region 1 is much smaller, at just below 500.

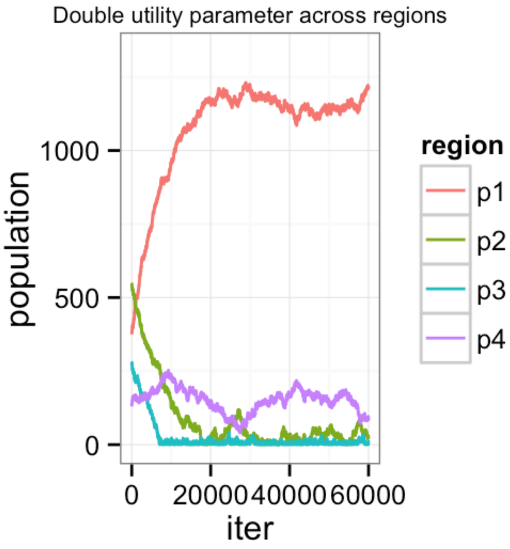
What would happen if the k parameter is equal across all regions? The steady state populations will be roughly equal to each other.

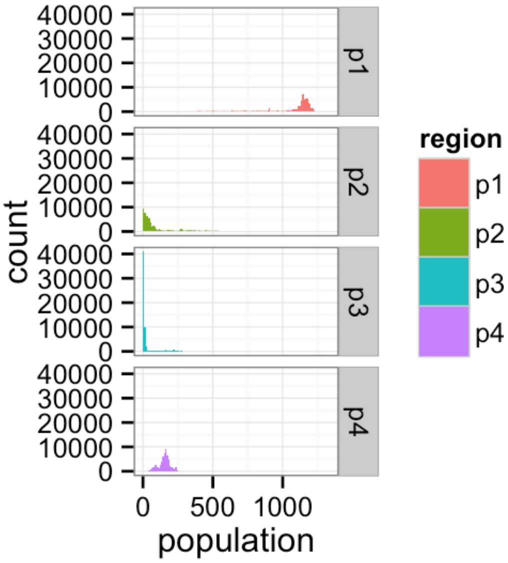


```
##   region pop.mean ss.utility change.in.utility
##       p1 333.9781   1616.022          45.02188
##       p2 344.7460   1605.254         202.25402
##       p3 324.6862   1625.314         -50.68624
##       p4 331.5897   1618.410        -196.58966
```

As the graphic above clearly demonstrates, the steady state utility levels and populations are approximately equal.

Given that a proportional increase or decrease of the cost of moving (below extreme levels) appears to leave the steady state distribution of population unaffected, we might expect a similar dynamic to hold in the case of utilities. In the below simulation, we double the utilities for all regions and expect to see no change in steady state populations.





```
##   region  pop.mean ss.utility change.in.utility
##       p1 1145.9437   3654.056        -766.94372
##       p2   37.8364   3562.164         509.16360
##       p3    6.5557   3393.444         267.44430
##       p4  144.6642   3655.336          -9.66418
```

Doubling all utility parameter values has a very different effects than doubling (or halving) all moving costs. Rather than maintain the same steady states, region 1 now accumulates a much larger share of the equilibrium population, while region 2 and 3 are essentially driven to zero.

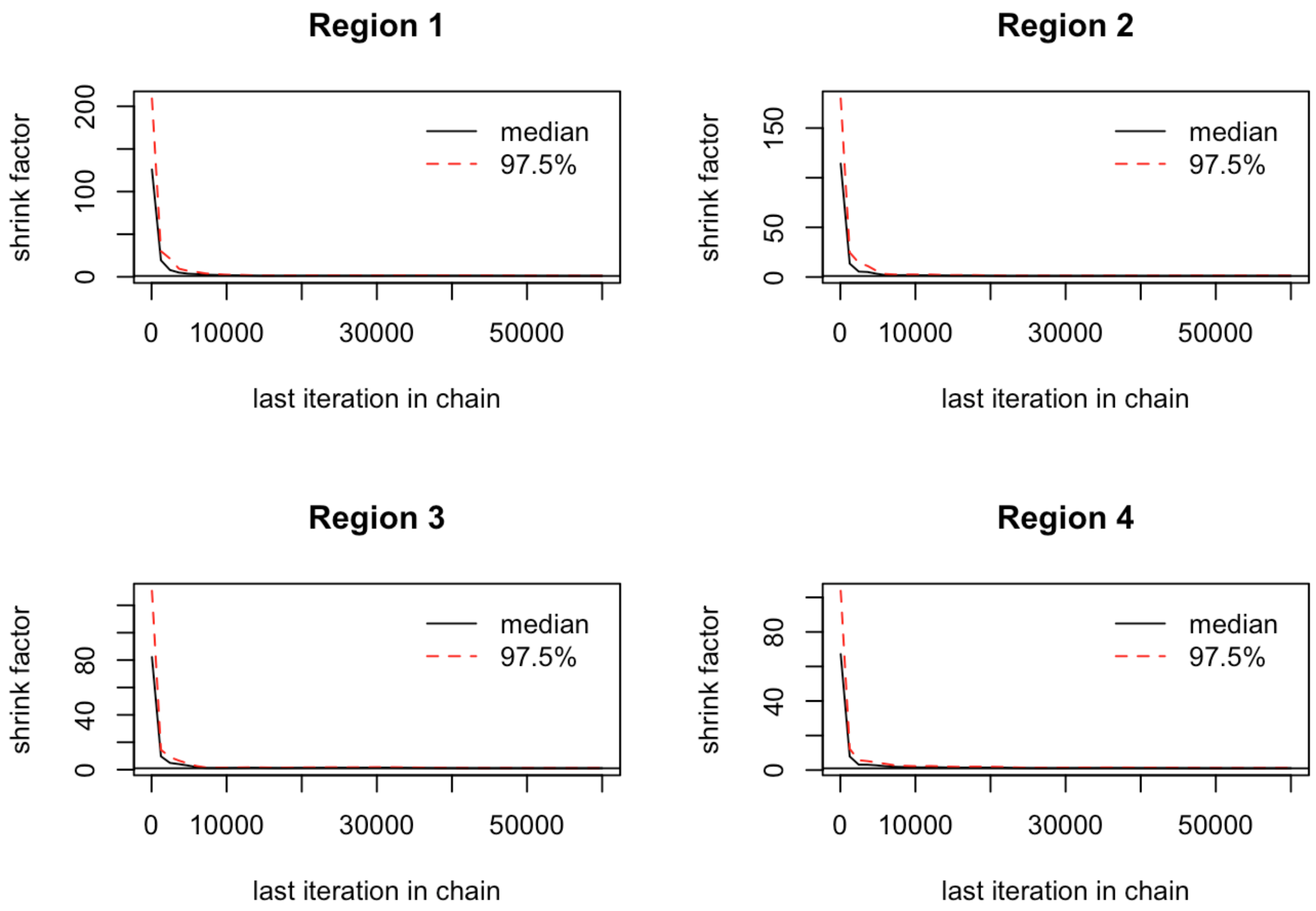
Convergence

The previous section examined model dynamics by varying parameter and recording steady state results. These results, however, depend upon the MCMC population estimates converging. While visual inspection suggests that in each case, the populations converged, this section will take a more formal approach to confirming convergence for each of the above cases.

The testing in this section will utilize the `coda` package, specifically the implementation of the Gelman-Rubin convergence diagnostics. The idea behind this approach centers on the variance within and between multiple runs of the Markov Chains with very different starting conditions. When the chains have mixed sufficiently, these variances should be roughly equal. The Gelman-Rubin diagnostic indicates how much improvement may be possible by iterating the chains infinitely.

Base model

For the base model, both the Gelman-Rubin shrinkage ratio and graphical aids will help us determine whether convergence has taken place.



Other models

For the sake of conciseness, the convergence statistics are displayed for the other exploratory cases in the table below:

```
##                                    Region 1 Region 2 Region 3 Region 4
## base case                          1.115613 1.177377 1.094621 1.112055
## init. cond.: all pop in region 4   1.147573 1.084725 1.061201 1.195277
## init. cond.: pop split b/w 1 and 4 1.054361 1.136892 1.163168 1.088447
## cost: increase into region 1       1.038894 1.063715 1.060324 1.063520
## cost: increase out of region 1     1.127866 1.085356 1.091157 1.164446
## cost: quadruple all costs          1.167496 1.107500 1.129100 1.254636
## utility: decrease k1 to 2000       1.090262 1.036825 1.074738 1.075151
## utility: set all k = 1950          1.158903 1.176094 1.199171 1.232043
## utility: double all k              1.405251 1.101483 1.036908 1.412885
```

As a rule of thumb, the diagnostic should be below 1.2 to be considered converged. Most satisfy this condition; notably, the quadruple cost and double utility paramter scenarios seem to require further iterations before convergence.

Summary and Future Works

This model represents an alternative way to view chains of migration. One can use these models to analyze not just what an equilibrium population is, but how an equilibrium is reached. Nagurney’s previous network equilibrium model only commented on the former.

By allowing population to be simulated using monte carlo methods, our model introduces randomness into the population model that Pan and Nagurney’s model does not include. While migration is definitely influenced by external macroeconomic factors, moves that lower an individuals’ economic utility can be expected to occur. This randomness can represent personal choices: the desire to be close to family for example. Our model accepts this randomness, but also allows utility to enter into the equation. Calculating our acceptance probabilities based on relative utility can be thought of as modeling an individuals decision to accept a move for personal reasons. An individual may be willing to accept a small drop in opportunities to be close to their family, but wouldn’t be willing to move to a severely economically depressed area for example.

Both of our models allow the study of chained migration, which set them apart from Nagurney’s original Network Model of Migration Equilibrium. Time in Pan and Nagurney’s model can be represented by the speed at which the markov chains converge. In our model, time is also represented by a speed of convergence, but this speed is influenced by the dynamics of how often moves are accepted by individuals.

In both of these cases, this concept of time can be used in future research. Utility functions may change with time, and more importantly utility functions may change before previous equilibriums are reached. For example, assuming a situation where it will take 10 years to arrive at a population equilibrium, our models can be used to show what would happen if an economic collapse occurred in one of the regions 5 years before equilibrium was reached. Viewing migration as a chained phenomenon allows us to see how exogenous changes in utility affect migration already influenced by differences in utility.

Currently, utility and costs are defined abstractly. Future research can also be done to relate these abstract concepts to concrete economic measures. We interpreted the constant term in our utility functions to be a “capacity” of the region. This concept of capacity can be used to relate economic measures such as employment to population.

A person entering an economy has two opposing effects on employment. By entering the economy, this person will increase the labor supply, which would be expected to have a negative impact on the employment rate. However, this person is also a consumer, and their consumption will lead to economic growth, which can be expected to increase employment. Capacity can be seen as a measure of the relative effects of these two countervailing forces. An economy at below capacity for example may be able to more efficiently absorb a person. Costs could be modeled in a similar way. Home prices, for example, could be factored into a more practical model.

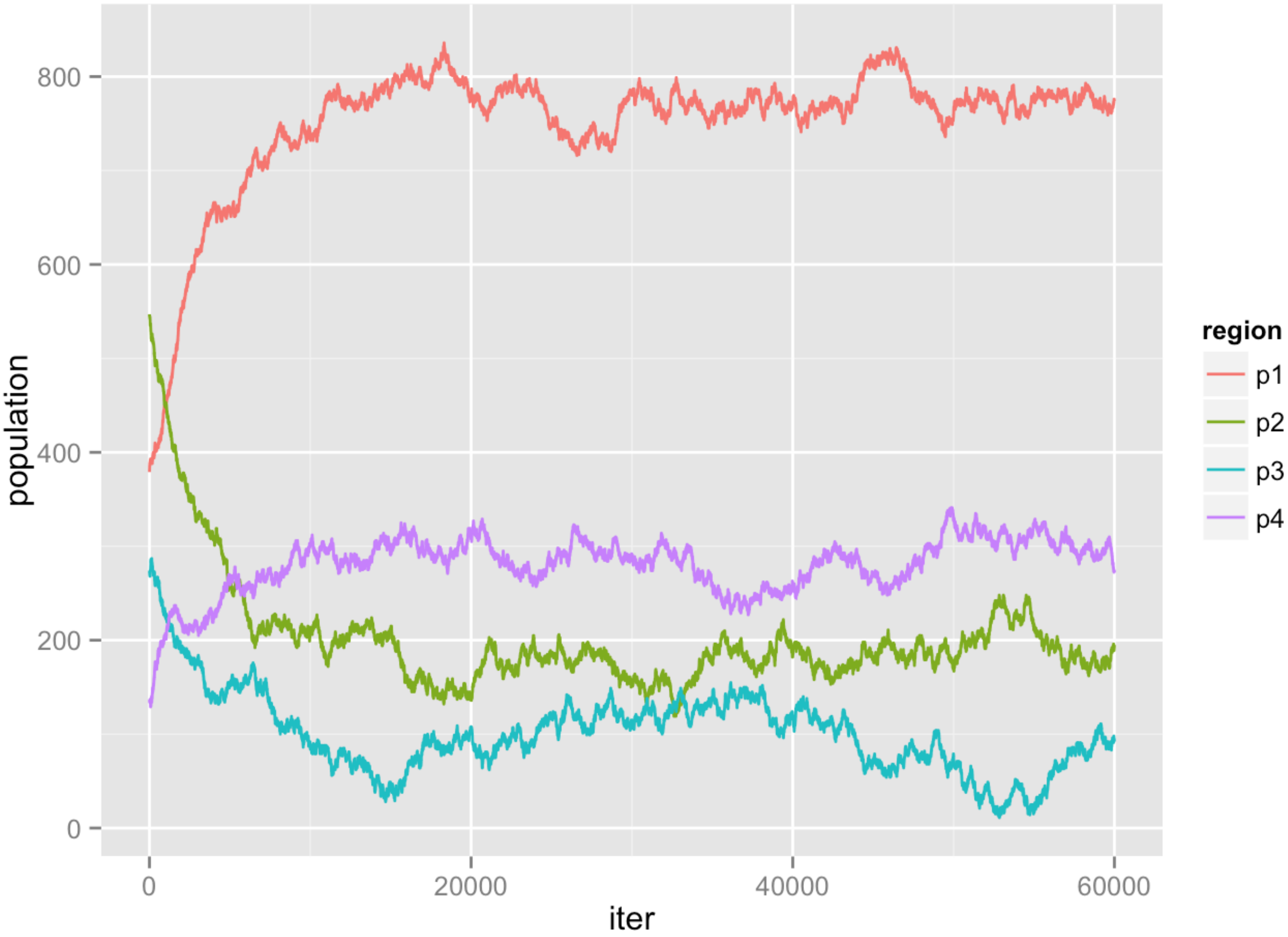
The goal of relating economic concepts to utilities and costs would be to forecast migration given different economic scenarios. This would allow us to answer questions such as how an exogenously developing regional economy would be expected to affect migration in the near term, or how quickly would an economic shock induce migration to neighboring regions.

step.MCMC is a generalized function that performs a step in our MCMC algorithm. Taking a population vector of length n, an n by n square cost matrix, and a capacity vector of length n, it will pick a random source, a random location, and generates a move with calculated probability a. It returns the updated population vector.

```
step.MCMC <- function(pop, capacity, move.cost) {
  # Params:

  # pop: vector of populations
  # capacity: list of capacities for utility functions
  #   Assumes all utility functions are of the form u = k - p
  # move.cost: matrix of costs for 1 person to move

  # returns population counts for each region

  # pan-nagurney utility-cost model helper function
  pn.prob <- function(src.utility, dest.utility, cost) {
    (dest.utility - cost) / src.utility
  }

  # randomly select source and destination regions
  num.regions <- length(pop)
  rand.ints <- sample(1:num.regions, 2, FALSE)
  region.src <- rand.ints[1]
  region.dest <- rand.ints[2]

  # create pop, utility, move cost lists for src and dest regions
  p <- list(src=pop[region.src], dest=pop[region.dest])
  u <- list(src=capacity[region.src] - p$src, dest=capacity[region.dest] - p$dest
+ 1)

  # calculate whether to accept this step using M-H
  cost <- move.cost[region.src,region.dest]

  # calculate a1

  a1 <- (u$dest - cost)/u$src

  # calculate a2

  lpop <- length(pop)
  lpop0 <- length(pop[pop != 0])
  lpop0m1 <- lpop0-1
  lpop0p1 <- lpop0+1

  if(p[1] == 1 && p[2] == 0){
    a2 <- 1
  } else if(p[1] == 1){
    qxxnew <- 1/(lpop0*(lpop-1))
    qxnewx <- 1/(lpop0m1*(lpop-1))
    a2 <- qxxnew/qxnewx
  } else if(p[2] == 0){
    qxxnew <- 1/(lpop0*(lpop-1))
    qxnewx <- 1/(lpop0p1*(lpop-1))
    a2 <- qxxnew/qxnewx
  } else{
    a2 <- 1
  }

  a <- min(1, a1*a2)

  if(runif(1) < a && p$src > 1){
    pop[region.src] <- p$src - 1
    pop[region.dest] <- p$dest + 1
  }

  return(pop)
}
```

Here’s an example of this function being used to create a plot:

```
library(ggplot2)
library(reshape2)
library(ggthemes)

pmat <- matrix(NA, 60000, 4)

p <- c(379, 547, 274, 135)
k <- c(2400, 1800, 1700, 1900)
c <- matrix(c(0,200,400,600,200,0,200,400,400,200,0,200,600,400,200,0),nrow=4)

pmat[1,] <- p

for(i in 2:60000){
  p <- step.MCMC(p, k, c)
  pmat[i,] <- p
}

pdata <- data.frame(pmat)
colnames(pdata) <- c("p1", "p2", "p3", "p4")
pdata$iter <- 1:length(pdata$p1)

pdatamelt <- melt(pdata, id.vars = c("iter"), measure.vars = c("p1", "p2", "p3",
"p4"),
                variable.name = "region", value.name = "population")

ggplot(pdatamelt, aes(x=iter, y=population, color=region)) + geom_line()
```



Bibliography

Nagurney, A. “A Network Model of Migration Equilibrium with Movement Costs.” *Mathematical and Computer Modelling* 13.5 (1990): 79-88. Web.

Pan, J., and A. Nagurney. “Using Markov Chains to Model Human Migration in a Network Equilibrium Framework.” *Mathematical and Computer Modelling* 19.11 (1994): 31-39. Web.

Constant, Amelie F., and Klaus F. Zimmermann. “The Dynamics of Repeat Migration: A Markov Chain Analysis1.” International Migration Review 46.2 (2012): 362-88. Web.