

Simulated Annealing

Jared Yu

April 20, 2018

```

dist_mat = matrix(c(0, 1, 2, 4, 9, 8, 3, 2, 1, 5, 7, 1, 2, 9, 3,
                    1, 0, 5, 3, 7, 2, 5, 1, 3, 4, 6, 6, 6, 1, 9,
                    2, 5, 0, 6, 1, 4, 7, 7, 1, 6, 5, 9, 1, 3, 4,
                    4, 3, 6, 0, 5, 2, 1, 6, 5, 4, 2, 1, 2, 1, 3,
                    9, 7, 1, 5, 0, 9, 1, 1, 2, 1, 3, 6, 8, 2, 5,
                    8, 2, 4, 2, 9, 0, 3, 5, 4, 7, 8, 3, 1, 2, 5,
                    3, 5, 7, 1, 1, 3, 0, 2, 6, 1, 7, 9, 5, 1, 4,
                    2, 1, 7, 6, 1, 5, 2, 0, 9, 4, 2, 1, 1, 7, 8,
                    1, 3, 1, 5, 2, 4, 6, 9, 0, 3, 3, 5, 1, 6, 4,
                    5, 4, 6, 4, 1, 7, 1, 4, 3, 0, 9, 1, 8, 5, 2,
                    7, 6, 5, 2, 3, 8, 7, 2, 3, 9, 0, 2, 1, 8, 1,
                    1, 6, 9, 1, 6, 3, 9, 1, 5, 1, 2, 0, 5, 4, 3,
                    2, 6, 1, 2, 8, 1, 5, 1, 1, 8, 1, 5, 0, 9, 6,
                    9, 1, 3, 1, 2, 2, 1, 7, 6, 5, 8, 4, 9, 0, 7,
                    3, 9, 4, 3, 5, 5, 4, 8, 4, 2, 1, 3, 6, 7, 0),
                    ncol = 15, byrow=TRUE) # distance matrix

alphafx = function(tau, p){ # alpha function
  return(p*tau)
}

betafx = function(m, j){ # beta function
  #return(m*j)
  return(100)
}

f = function(theta, dist_mat){ # distance function
  n = length(theta)-1 # number of cities
  d = diag(dist_mat[theta[-(n+1)],theta[-1]])
  return(sum(d))
}

sa.improved = function(theta0, dist_mat, tau0, p, m, max.stage){ # simulated annealing
  n = length(theta0)-1 # number of cities
  theta = theta0
  tau = tau0
  hist.theta = array(dim = max.stage*m)
  best.candidate = theta0 # keep tracking the best candidate
  for (j in 1:max.stage) {
    for (i in 1:m) {
      changePosi = sample(2:n,2)
      theta[changePosi] = theta[rev(changePosi)]
      delta = f(theta,dist_mat)-f(theta0,dist_mat)
      if (f(best.candidate,dist_mat) > f(theta,dist_mat)) best.candidate = theta
      if (delta > 0) {
        a.prob = exp(-delta/tau) # accepting probability
        theta = ifelse(rep(runif(1) < a.prob, n+1), theta, theta0)
      }
    }
  }
}

```

```

    }
    theta0 = theta
    hist.theta[i+(j-1)*m] <- f(theta,dist_mat)
  }
  tau = tau*p
}
out = list(theta, hist.theta, best.candidate)
names(out) = c('theta','hist.theta','best.candidate')
return(out)
}

set.seed(2018)

theta0 = c(1,sample((1:15)[-1]))
theta0 = c(theta0,theta0[1])

ans = sa.improved(theta0 = theta0, dist_mat = dist_mat, tau = 400, p = 0.999, m = 10
0, max.stage = 10000)
f(ans$best, dist_mat) # 18

set.seed(243)

theta0 = c(1,sample((1:15)[-1]))
theta0 = c(theta0,theta0[1])

ans = sa.improved(theta0 = theta0, dist_mat = dist_mat, tau = 400, p = 0.999, m = 10
0, max.stage = 10000)
f(ans$best, dist_mat) # 17

```