Job Seekers and Social Networks: Agent Based Modelling Using a Database of Job Seekers in the London Borough of Camden

Amy Heineike and Paul Ormerod

Volterra Consulting, London

December 2008

aheineike@volterra.co.uk

Context and Summary

HEFCE and the DTI via the Office of Science and Innovation made an award to University College London (UCL) under the HEIF 3 competition for a project entitled 'Building Sustainable Communities'.

This work was carried out as part of a wider project of which Meganexus was the principal partner, partly financed by UCL under this scheme. The project is called 'Reducing barriers to opportunities for people socially excluded –REBOPSE'.

The overall aim of REBOPSE (which became known colloquially as 'Green Maniac') was to address sustainability simultaneously through:

- *widening access to employment opportunities (particularly for the disadvantaged) and*
- *improving the matching of skills and enterprise needed for the emerging green economy*

It is well established in the literature that social networks are very important in the effectiveness of job seeking. Social networks serve both as a way of obtaining information about job opportunities (most jobs are obtained through personal contact) and of creating a set of attitudes in which individuals are expected to work rather than not work.

Meganexus has a large on-line database of workless individuals seeking jobs, principally though not exclusively in the boroughs of Greater London. An important part of the overall project was to interlink job-seeker, employer and Green networks, to increase the search space and diversity of choice of potential employment available for members of the workless database.

The specific purpose of the part of the project described in this report was to build an agent based model of social networks amongst workless individuals, which could be used as a means of assessing the effectiveness of projects such as REBOPSE.

Initially, we were provided with a sample of 7841 'nodes' and the links between them. The vast majority of the 'nodes' are individual jobseekers living in the London Borough of Camden, and just 68 have multiple roles – for example job broker, employer and advisor.

In total, 7943 links between 'nodes' were identified within the data set. Almost all of the links go between job seekers and the 68 nodes with multiple roles. The connections between the workless individuals were very sparse. For example out of the total of 7943 links, 6742 links connect to just three nodes. In other words, the workless population generally have tiny social networks to aid them find a route into employment.

We developed an initial agent based model calibrated on the network structure described above.

We then used the model to simulate the potential effect of interaction with Green networks in a) increasing the connections between the workless individuals in the on-line database and so helping to increase the flow of information about job opportunities and b) estimating the resulting potential increase in employment under a range of assumptions.

The sparseness of the initial social network connecting the workless individuals means that the addition of a relatively small number of connections makes little difference to the potential employment outcome. However, beyond the creation of around a further 1000 links, the potential impact increases rapidly.

We developed a further version of the model, in particular enabling a distinction to be made as to whether or not an individual who obtains a job passes on information he or she obtains about other jobs across his/her social network.

This was calibrated using a small survey carried out by Meganexus amongst the individuals on the initial database.

In particular, this enabled us to distinguish between the 22 per cent of job seekers who increased their links during the period in which they were involved with Green Maniac, and the 78 per cent who did not The model reveals a potentially dramatic difference between the two groups in the impact on potential employment levels of additional connections being generated. It is much stronger for the minority who have already developed more links in their social network

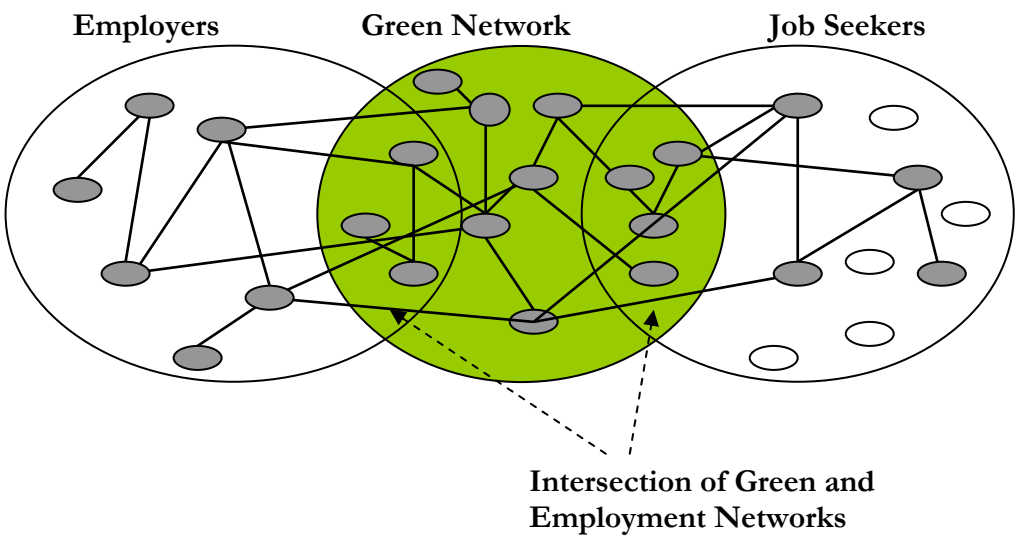
The amount of data available to us was limited in terms of further calibration. But the model can be used to benchmark the potential impact on employment of increasing the number of connections within the social network of the workless. This can be done under a range of assumptions about the propensity to pass job information on to others in the network, and the rate at which employment offers are made to people in the network. It therefore offers a way of assessing the future effectiveness of programmes such as Green maniac (REBOPSE).

1 Introduction

Sustainability Principle number 5 of the REBOPSE project was to use sound science in evaluation. This is to try to ensure that policy is developed and implemented on the basis of strong scientific evidence, whilst taking into account scientific uncertainty.

A key aim of the project was to enhance links between job seekers, both in terms of the network which connects them and their connections with the wider social and economic world.

The overall project connected individuals in Green networks with job seekers identified on the Meganexus database. Green networks have strong social connections, in contrast to the weak ones found in most groups of workless individuals. Linking the Green and job seekers networks should increase the available connections for job seekers in the job seekers network leading to increased employment opportunity. Prospects for job seekers that become active members of the Green Network should be raised significantly as shown in the following diagram:



Section 2 of the report summarises the importance of social networks in getting people into employment, and provides a list of references in the literature, from diverse sources, which supports this conclusion. Section 3 describes the initial data with which we were provided, and section 4 puts this in the context of the calibration of the initial version of the agent based model we developed.

Section 5 describes the subsequent development of the theoretical model. Section 6 plots results from the survey data we received from Meganexus, and section 7 uses this to calibrate the versions of the model set out in section 8. Finally, section 8 provides a brief conclusion. The Matlab code with which to operate the model is available in the Appendix.

2 Social networks, employment and worklessness

There is good evidence in the literature that social networks are important in explaining both why people get jobs, and why individuals with only weak social connections can form clusters with high levels of worklessness. Social networks serve both as a way of obtaining information about job opportunities (most jobs are obtained through personal contact) and of creating a set of attitudes in which individuals are expected to work rather than not work.

Examples of the relevant literature are:

- Granovetter, M. (1973), 'The Strength of Weak Ties', *American Journal of Sociology*, vol. 78, pp.1360-80
- Granovetter, M. (1974), *Getting a Job: A Study of Contacts and Careers*, Harvard University Press
- Holzer, H. (1988), 'Search method used by unemployed youth", *Journal of Labor Economics*, 6, 1-20.
- Montgomery, J. (1992), 'Job Search and Network Composition: Implications of the Strength-of-Weak Ties Hypothesis', *American Sociological Review*, vol.57, pp586-96
- Fletcher DR, Woodhill D, Herrington A (1998), *Building Bridges into Employment and Training for Ex-offenders*, York Publishing Services
- Hannan, C. (1999), 'Beyond Networks: 'Social Cohesion' and Unemployment Exit Rates', University of Essex Institute for Labour Research, Discussion Paper 99/28
- Cartmel F and Furlong A (2000) *Youth Unemployment in Rural Areas*, York Publishing Services
- Meadows PC (2000), *Young Men on the Margins of Work*, Joseph Rowntree Foundation, York
- Topa, G. (2001), 'Social interactions, local spillovers and unemployment,' *Review of Economic Studies* 68, 261-295
- Calvó-Armengol, A. and Jackson, M.O. (2004), 'The Effects of Social Networks on Employment and Inequality,' *American Economic Review*, 94, 426-454.
- Calvó-Armengol, A. and Y. Zenou, Y (2005), 'Job matching, social network and word-of-mouth communication', *Journal of Urban Economics*, 57, 500-522.
- Calvó-Armengol, A. and Ioannides, Y.M. (2008). 'Social networks in labour markets'. *The New Palgrave Dictionary of Economics*. Second Edition, eds. Steven N. Durlauf and Lawrence E. Blume. Palgrave Macmillan.

A theoretical model compatible with this is:

- P Ormerod and L Smith (2000) 'Job Search, Unemployment and the Topology of Social Networks', *Proceedings of the 10th World Congress on Social Economics*, Cambridge

3 Benchmark data

Volterra initially developed a theoretical model of the potential influence of social networks on similar lines, and did initial analysis using an extract of data from the Meganexus Camden Working dataset in September 2007.

The data covered 7841 nodes of which the majority are jobseekers. 68 were identified as having multiple roles – for example job broker, employer and advisor. Of the 7943 links within the data set the majority of the links go between job seekers and the 68 nodes with multiple roles. Of these 6,742 links connect to just three nodes.

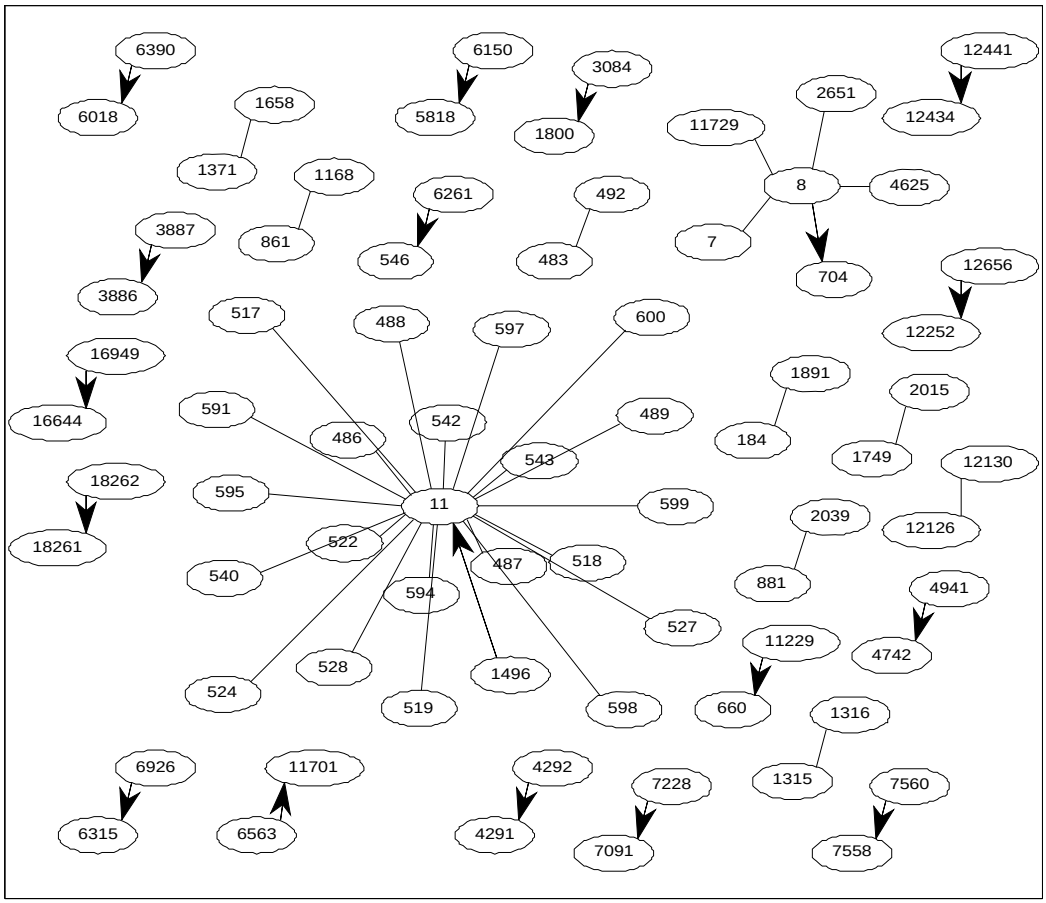
Table 1 summarises the connections.

Table 1 *Links in the data*

		To	
		Job Seeker	**Providers**
From	**Job Seeker**	100	7,673
	Providers	12	157

Figure 1 plots the links between job seekers in the following figure. One way links are shown with an arrow, while two way links are just a line.

Figure 1 Links between Job Seekers



It is readily apparent that the network which connects the jobseekers in this extract from the Meganexus database is very sparse. There is little opportunity for information about job opportunities to spread across this network.

4 Initial model calibration

The key dynamic of the model is the passage of information through a group. Knowledge about a job is first given to one job seeker who may get the job, but if not, has the opportunity to pass the information on to all their contacts. The model therefore explores the role of information spread in a community.

Because this model focuses on information spread it does not forecast the complex circumstances that lead an individual to apply for or be successful in attaining a particular job, for example their skills or confidence. Instead an overall probability that a person who hears about a job attains it is used. This probability encompasses all of the elements of the process:- the chance they will want or be suitable for the job,

the chance they will write the application, and the chance their application will be successful.

An abstracted mechanism for information spread is also used. A probability of passing on job information is applied for each contact that the job seeker has. The information is therefore not spread in a directed way to particular job seekers, but it spread widely to a share of an individual's contacts.

By abstracting in this way, the model draws out the role of information spread in job uptake. It also allows us to understand the importance of this type of dynamic in a community of job seekers.

We designed a model which simulates the take up of jobs in the jobseeker community, and shows the importance of increasing the connectivity between individuals, as the literature suggests.

The basic model followed these rules:

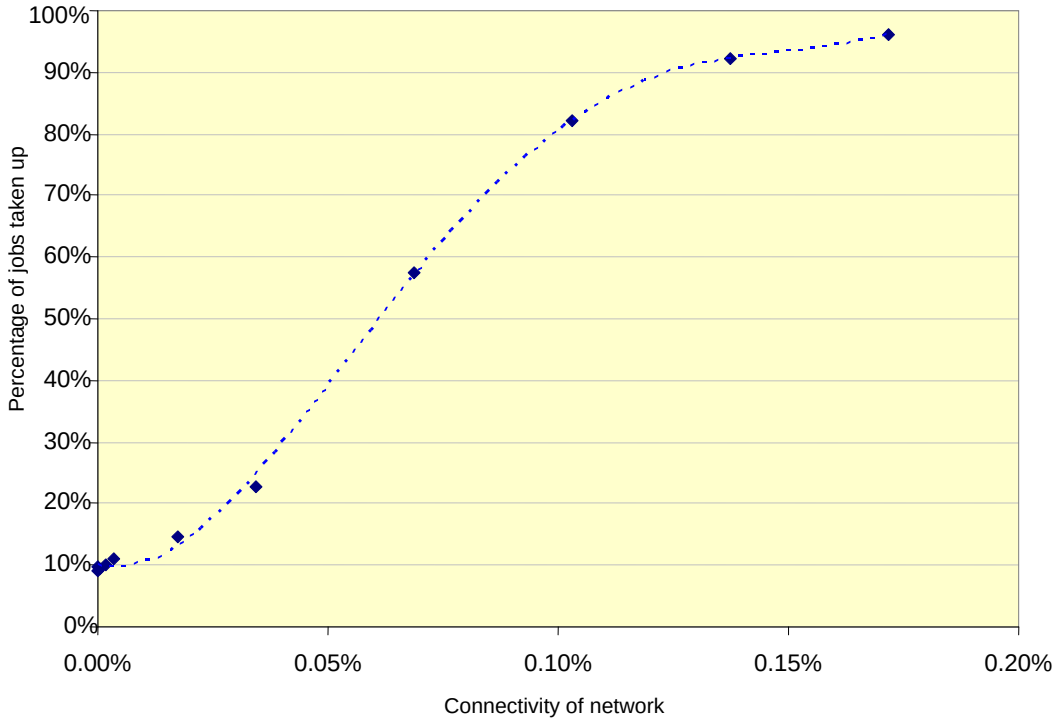
- Start off with the number of workless people given in the data (7624) and the network between them (84 links)
- Each period offer a job to one person at random. They get the job with probability=0.1. If they don't get the job they have the option to pass on the job with probability 0.5 to each of their contacts
- Each person who is offered a job also gets the jobs with probability 0.1 and passes it on to each of their contacts (who hasn't already seen it) with probability 0.5.
- Note down if someone is employed
- Repeat this process for 100 jobs, and then repeat the whole process 50 times.

The model can then be run with increased numbers of links between the workless people.

- Choose how many additional links to add, add them at random between any pair of workless who are not already connected.
- For each of the 50 repeats create a new network, adding the links at random to the base observed network.

Figure 2 below shows the key result from this analysis – the percentage of jobs which are eventually taken by someone in the job seeker network, for different levels of connectivity in the network. Connectivity is the per cent of possible links which are actually formed. If connectivity = 100% every possible link exists, for connectivity = 0% no links exist.

Figure 2 Initial model results: Percentage of jobs taken up by jobseekers in the network for different numbers of additional links



For low numbers of connections – the starting point for the analysis – we see that jobs are only found out about by one jobseeker, if they don't take it up then knowledge of the job is missed by all other agents. As the number of connections increases, jobseekers are able to pass on information to other members of the network who might also be interested in the job, or who could also pass on the job to someone else. By the time the network becomes very highly connected almost all jobs are taken up by someone in the network.

Within this size of network, low levels of connectivity can actually still represent a large number of links. For example, 0.2% connectivity reflects 50,000 new links in the network.

5 Extended model

We developed two separate variants of the model. The difference depends upon how people with jobs act once they have take a job up. In version A, once an individual has a job they are no longer involved in spreading information about jobs. In version B they still continue to hear about jobs and pass the information on to their contacts – they are just restricted from taking the job up.

We also improved the software code so that it was feasible to solve the model for many more individual solutions than the 50 used in the initial analysis. In the results

below, we use 1000 separate solutions which, given the stochastic nature of the model, is more reliable.

Variant A: Individuals with jobs no longer discuss job opportunities

Variant A follows these rules:

- Start with 1000 individuals
- Each period offer a job to one workless person at random. They get the job with probability=0.1. If they don't get the job they have the option to pass on the job with probability 0.5 to each of their workless contacts
- Each person who is offered a job also gets the jobs with probability 0.1 and passes it on to each of their contacts (who hasn't already seen it) with probability 0.5.
- If someone gets the job, the job information is spread no further.
- Repeat this process for 1000 jobs, and then repeat the whole process 1000 times.

Variant B: Individuals with jobs still active in spreading information

Variant B follows these rules:

- Start with 1000 individuals
- Each period offer a job to any one individual at random. If they don't have a job, they get the job with probability=0.1. If they don't get the job they have the option to pass on the job with probability 0.5 to each of their contacts
- Each person who is offered a job also gets the jobs with probability 0.1 and passes it on to each of their contacts (who hasn't already seen it) with probability 0.5.
- If someone gets the job, the job information is spread no further.
- Repeat this process for 1000 jobs, and then repeat the whole process 1000 times.

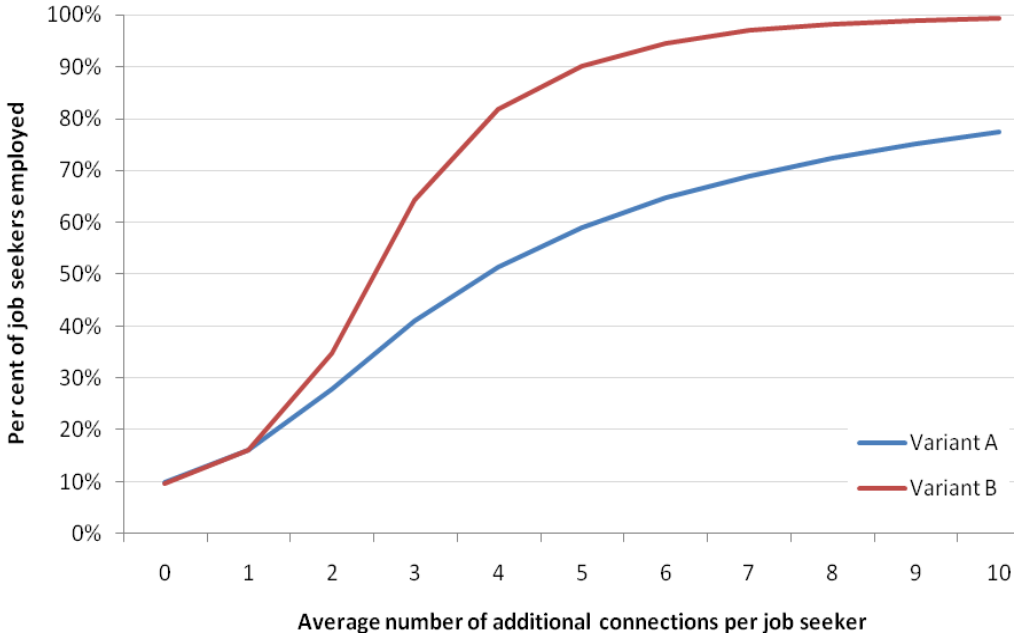
The model can then be run with increased numbers of links between the workless people.

- Choose how many additional links to add, add them at random between any pair of workless who are not already connected.
- For each of the 1000 repeats create a new network, adding the links at random to the base network.

The key result from this model is the per cent of individuals who eventually end up employed. In the simulations, 1000 jobs are offered to 1000 individuals, and the results show how many manage to get a job.

We plot this compared to the average number of new links for each job seeker.

Figure 3 Potential job take up as links improve



For low numbers of connections – the starting point for the analysis – we see that jobs are only found out about by one jobseeker, if they don't take it up then knowledge of the job is missed by all other agents. As the number of connections increases, jobseekers are able to pass on information to other members of the network who might also be interested in the job, or who could also pass on the job to someone else. By the time the network becomes very highly connected almost all jobs are taken up by someone in the network.

The important point here is that even very low numbers of links quickly improve overall job take up. If every job seeker passes on information about jobs to just 1 other, the take up rate improves by almost 50%, if everyone had 3 links take up improves 4 to 6 times.

While the same overall pattern holds for both variants we see that for variant B, where job holders still find and pass on information, there is an even higher value to increased connectivity.

6 Subsequent survey data

It must be noted that we only received one further set of data during the project, which was a survey of job seekers participating in the project and who were asked about their experiences. We were not involved at any stage either in the design of the

survey or in the scope of the questions. Nevertheless, several of the findings have proved useful in further calibration of the model.

We were sent the results of the responses to nine questions, and Figures 4.1 to 4.5 below summarise these (Figure 4.5 combines the responses to two of the questions).

Figure 4.1 'Through Green Maniac I developed positive relationships with businesses and other job seekers'

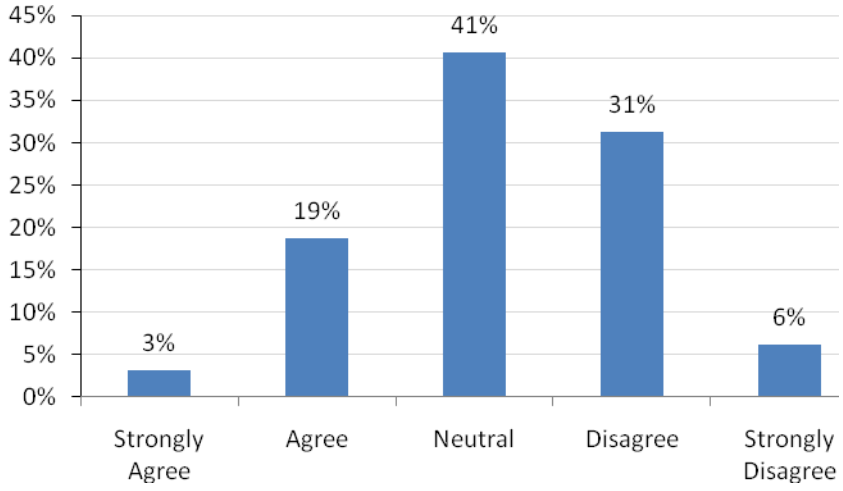


Figure 4.2 'I was helped towards employment opportunities'

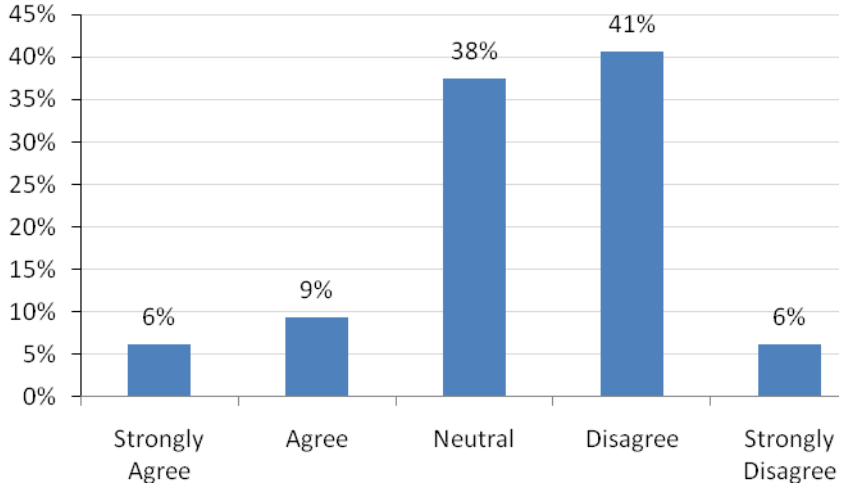


Figure 4.3 'I will continue to use the Green Maniac program'

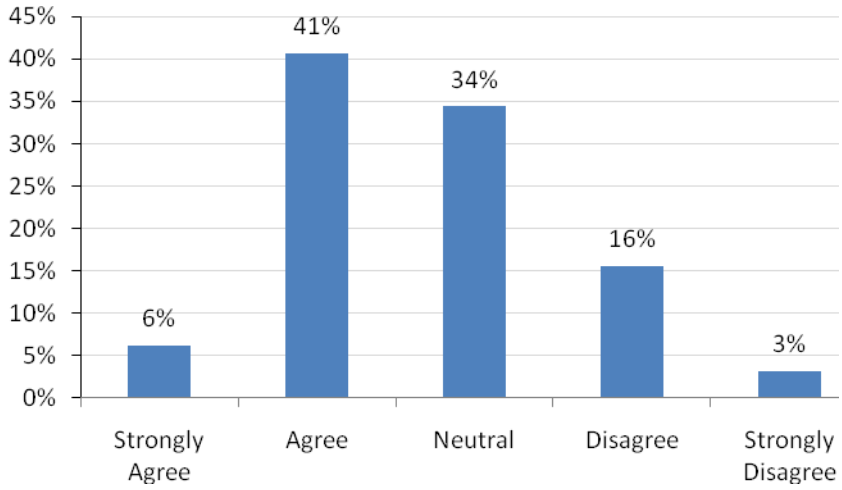


Figure 4.4 'I have become employed since joining'

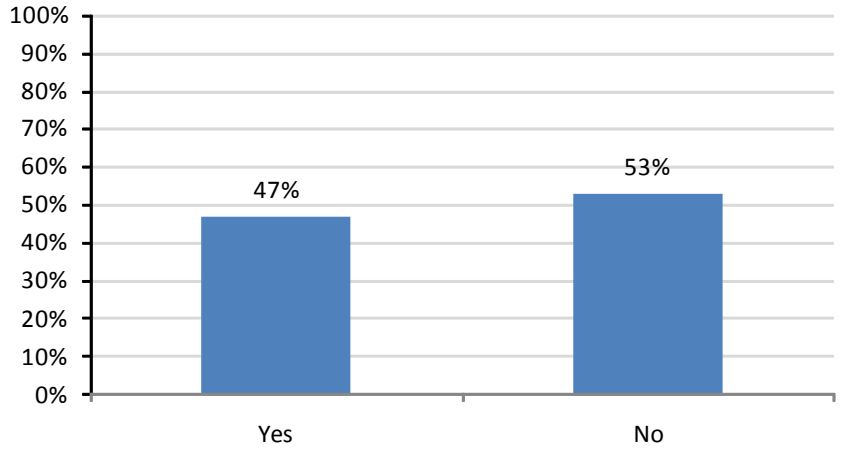
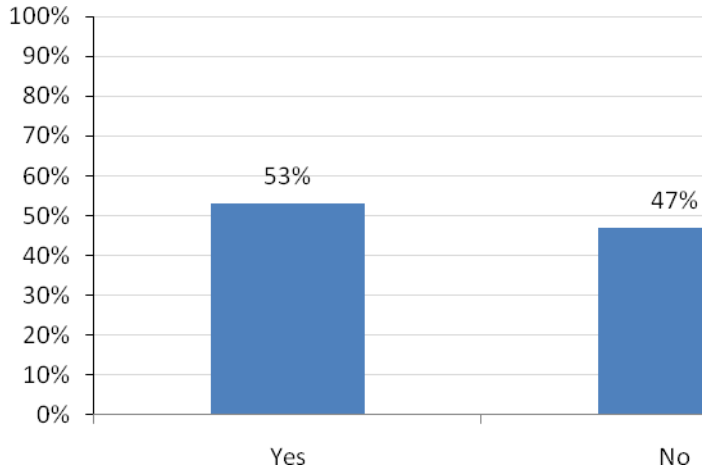


Figure 4.5 (Using Q4 and Q5 in the survey) 'I got a job or job offer since joining'



The key findings that we are able to use in the model are:

- 22% increased the links they had with businesses and job seekers
- 16% were helped towards job opportunities
- 47% will continue to use Green Maniac
- And 53% got a job or job offer in the period they were involved in the project.

7 Calibration of Extended Model

Of those job seekers involved with Green Maniac, 53% got a job or job offer over the period of their involvement. Since 16% were helped towards a job, we should expect roughly 47% to get a job in the absence of the programme (since a 16% improvement on 47% gives 53%).

In order to decide how many jobs to offer across the network we need to consider both the number of people who we would expect to become employed in the absence of Green Maniac (47%) and probability of someone getting a job they hear about (assumed to be 10%). Combining these suggest that the model should include 4.7 times as many job offers as individuals in the model.

22% of job seekers increased the number of links they have – we will call this subset the 'active' job seekers. It can be assumed that there is no growth in the number of connections for the 'non-active' job seekers.

The results are used in the following way:

- Keep considering 1000 workless individuals
- Increase the number of jobs offered to 4700
- Define active and non-active jobseekers. Only add links between active jobseekers. 22% are active.
- Use the same probabilities as before: probability of getting a job you have heard about is 10%, while the probability of passing it on to a contact is 50%.

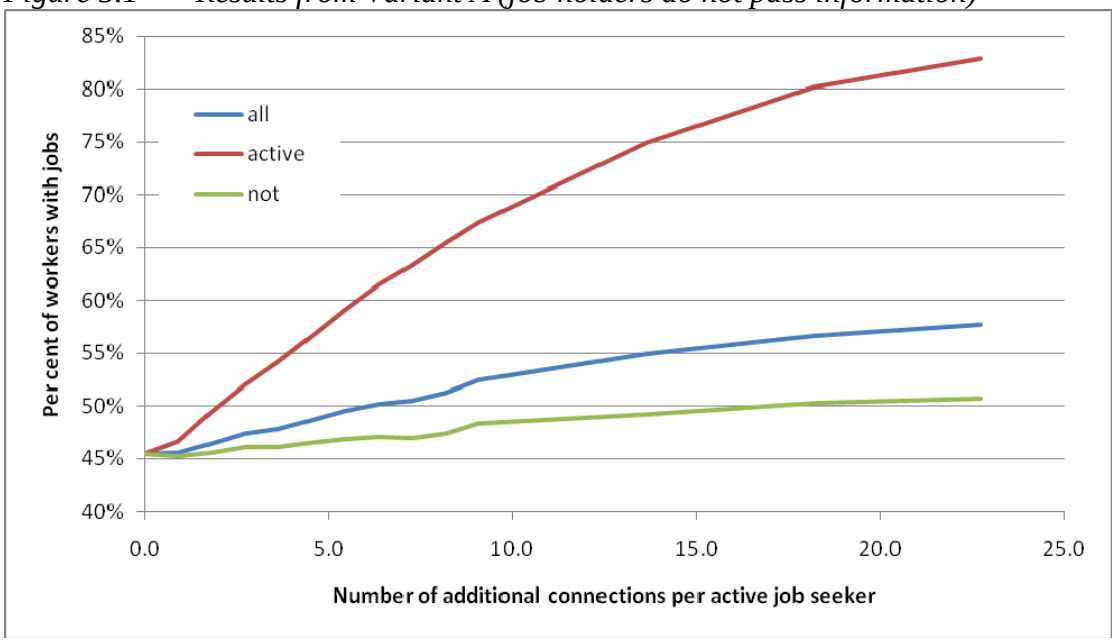
The results from the two variants of the model are shown below.

For variant A, with job holders playing no part in passing on job information, lifting the level of employment to 53% for active jobseekers takes between an additional 0.5 and 1 links each on average. This corresponds to an additional 110 to 220 new links between the active individuals, or a connectivity of 0.01% and 0.02% across the network as a whole.

Increasing the *overall* average, in other words including now those we define as non-active, to 53% takes a larger number of links – around 2 each, or around 440 links between the active workers, or 0.04% across the network as a whole.

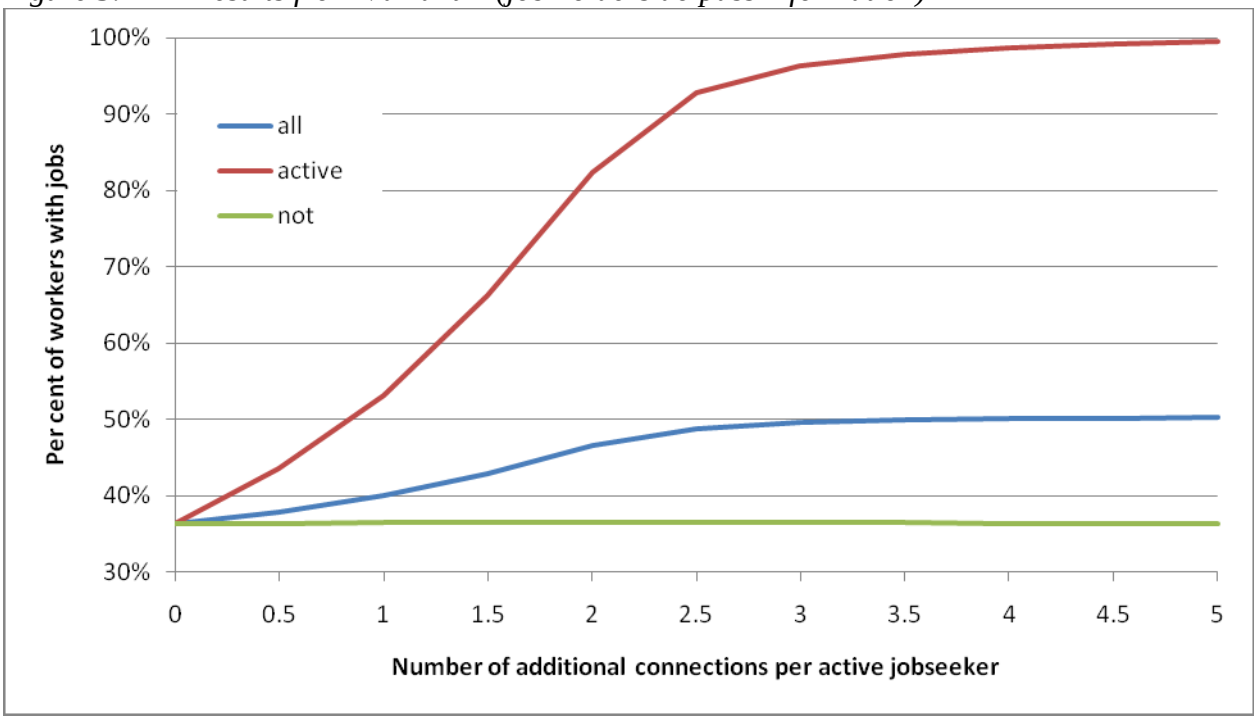
The implication of this is that a modest improvement in connectivity of the social network of the 'active' seekers can help to raise the eventual level of employment.

Figure 5.1 Results from Variant A (job holders do not pass information)



With Variant B of the model, the starting point is lower, since someone with a job can hear about a new opportunity but not pass it to any jobseeker. Again around 1 connection is enough to improve take up to 53% for the 'active' job seekers.

Figure 5.2 Results from Variant B (job holders do pass information)



If the number of connections were to continue rising through Green Maniac's involvement we could see benefits rising sharply. In particular, if the number of

connections rose as high as 4 links per active individual and all individuals became active, we might expect to see employment rates over 80%.

8 Conclusions

The aim of this report has been to use sound science to evaluate the effectiveness of the REBOPSE (Green Maniac) project. A key aim of the wider project was to enhance links between job seekers and the wider social and economic world.

The report started by presenting evidence from the wider literature which has established the importance of social networks in job seeking. Examination of an existing Meganexus database, showed the sparseness of existing links between job seekers.

This network data was used as the basis for the initial agent based model. It simulated how increasing job take up could be generated by increased connectivity in the social network. As more links between job seekers were created, more job seekers found employment. The benefits rose quickly after the number of links passed an initial threshold.

Following the implementation of the project, a further model was developed which differed in the way it simulated the behaviour of job seekers once they find employment. The model was calibrated using a small survey carried out by Meganexus amongst participants.

The model enabled us to distinguish between the 22 per cent of job seekers who increased their links during the period in which they were involved with Green Maniac and the 78 per cent who did not. The model reveals a potentially dramatic difference between the two groups in their potential employment as those who'd increased their links increased them even further.

The amount of data available to us was limited in terms of further calibration. But the model can be used to benchmark the potential impact on employment of increasing the number of connections within the social network of the workless. This can be done under a range of assumptions about the propensity to pass job information on to others in the network, and the rate at which employment offers are made to people in the network. It therefore offers a way of assessing the future effectiveness of programmes such as Green maniac (REBOPSE).

Matlab Code

VARIANT A

```
function results = greenA(rep,w, w_net, N_ww, j, p_get, p_pass, share,
twoway)
% Green Maniac Model, December 2008, VARIANT A - people with jobs do not
% pass job information on
%
% Inputs:   w       number workless,
%           w_net   list of beginning links between workless,
%           N_ww    vector of number of new links to add
%           j       number of jobs offered,
%           p_get   probability a person will get a job they hear about,
%           p_pass  probability they will pass the info on
%           share   % of workers between whom extra links are added
%           twoway  0 if all added links are one way
%                   1 if they are two way

% start by using a random network for building networks
tic
results.version='01 Dec 08';

% set up
ind=1:w;
lw_net=length(w_net);
L=10000;
while L<w
    L=L*10;
end

n_active=zeros(length(N_ww),rep);
offer_active=zeros(length(N_ww),rep);
jobs_all=zeros(length(N_ww),rep);
jobs_active=zeros(length(N_ww),rep);
jobs_notactive=zeros(length(N_ww),rep);

% Start loop
for n=1:length(N_ww)
    n_ww=N_ww(n)
    jobs=zeros(w,rep);
    for r=1:rep
        %% Increase density of workless links
        isactive=(rand(1,w)<share);
        active=ind(isactive);
        inactive=ind(isactive==0);
        a=length(active);
        net=w_net;
        new=0;
        while new<n_ww
            tos = active(ceil(a.*rand(n_ww-new,1)));
            froms = active(ceil(a.*rand(n_ww-new,1)));
            net=[net;tos' froms'];
            if twoway==1
                net=[net;froms' tos'];
            end
            % remove self links
            net=net(net(:,1)~=net(:,2),:);
            % remove duplicates (quickly)
            codes=L*net(:,1)+net(:,2);
            lc=numel(codes); tmp=sort(codes);
            tmp2 = diff(tmp)~=0;
            tmp2(lc,1) = true;
            codes = tmp(tmp2);
```

```
            if numel(codes)~=lc
                %need to remake net
                parta=floor(codes/L);
                partb=codes-L*parta;
                net=[parta partb];
            end
            new=size(net,1)-lw_net;

        end

        %% Examine how jobs pass through

        for job=1:j  %for each job

            % Initial job offer
            pick=ind(jobs(:,r)==0);
            to=pick(ceil(length(pick)*rand(1))); %offer the job
             %count if active
            offer_active(n,r)=offer_active(n,r)+isactive(to);
            %decide whether they get the job
            jobs(to,r)=job*(rand(1)<p_get);
            flag=jobs(to,r)>0;
            use=jobs(:,r)==0; % don’t include people with jobs
            while flag==0;
                use(to)=0;%don't re-offer the job
                neigh=[];
                for from=1:length(to)
                    froms=to(from);
                    neigh=[neigh; net(net(:,1)==froms,2)];
                end
                send=use(neigh).*(rand(length(neigh),1)<p_pass);
                if sum(send)>0
                    to=neigh(send>0); %make offers to valid people
                    want=to(rand(length(to),1)<p_get);
               else
                    to=[];
                    want=[];
                end
                if length(want)>0
                    jobs(want(1),r)=job;
                    flag=1;
                end
                flag=flag+(isempty(to));
            end
        end % job ends
        n_active(n,r)=length(active);
        jobs_all(n,r)=sum(jobs(:,r)>0)/w;
        jobs_active(n,r)=sum(jobs(active,r)>0)/length(active);
        if isempty(inactive)==0
            jobs_notactive(n,r)=sum(jobs(inactive,r)>0)/length(inactive);
        end

    end %rep ends

end
results.N_ww=N_ww;
results.wall=sum(jobs_all,2)'/rep;
results.jall=sum(w*jobs_all/j,2)'/rep;
results.wactive=sum(jobs_active,2)'/rep;
results.jactive=sum(n_active.*jobs_active./offer_active,2)'/rep;
results.wnot=sum(jobs_notactive,2)'/rep;
results.jnot=sum((w-n_active).*jobs_notactive./(j-offer_active),2)'/rep;
toc
```

VARIANT B

```
function results = greenB(rep,w, w_net, N_ww, j, p_get, p_pass,
share, twoway)
% Green Maniac Model, December 2008, VARIANT B - people with jobs
% still pass job information on
%
% Inputs:   w       number workless,
%           w_net   list of links between workless,
%           N_ww    vector of number of new links to add
%           j       number of jobs offered,
%           p_get   probability a workless person will get a job they
%                      hear about,
%           p_pass  probability they will pass the info on if they
%                      don't get it
%           share   % of workers between whom extra links are added
%           twoway  0 if all added links are one way
%                   1 if they are two way

tic
results.version='04 Dec 08';

% Set up
ind=1:w;
lw_net=length(w_net);
L=10000;
while L<w % Used in link code
    L=L*10;
end

%Initialise variables
n_active=zeros(length(N_ww),rep);
offer_active=zeros(length(N_ww),rep);
jobs_all=zeros(length(N_ww),rep);
jobs_active=zeros(length(N_ww),rep);
jobs_notactive=zeros(length(N_ww),rep);
%% Set up loop
for n=1:length(N_ww) % for each number of links
    n_ww=N_ww(n)
    jobs=zeros(w,rep);
    for r=1:rep % for each repeat

        %% Increase density of workless links
        isactive=(rand(1,w)<share);
        active=ind(isactive);
        inactive=ind(isactive==0);
        a=length(active);
        net=w_net;
        new=0;
        while new<n_ww
            tos = active(ceil(a.*rand(n_ww-new,1)));
            froms = active(ceil(a.*rand(n_ww-new,1)));
            net=[net;tos' froms'];
            if twoway==1
                net=[net;froms' tos'];
            end
            % remove self links
            net=net(net(:,1)~=net(:,2),:);
            % remove duplicates (quickly)
            codes=L*net(:,1)+net(:,2);
            lc=numel(codes); tmp=sort(codes);
            tmp2 = diff(tmp)~=0;
```

```
            tmp2(lc,1) = true;
            codes = tmp(tmp2);
            if numel(codes)~=lc
                %need to remake net
                parta=floor(codes/L);
                partb=codes-L*parta;
                net=[parta partb];
            end
            new=size(net,1)-lw_net;

        end

        %% Examine how jobs pass through

        for job=1:j  %for each job

            if sum(jobs(:,r)==0)==0  % if eveyone has a job stop!
                break
            end
            use=ones(w,1);
            % Initial job offer - offer to anyone
            to=ind(ceil(w*rand(1))); %offer the job
             %count if active
            offer_active(n,r)=offer_active(n,r)+isactive(to);
            flag=0;

            while flag==0
                % decide if people offered job get it
                choice=(rand(length(to),1)<p_get).*(jobs(to,r)==0);
                wants=to(choice>0);
                if isempty(wants)==0
                    %somebody wants the job - so give it to them
                    jobs(wants(1),r)=job;
                    flag=1;
                else
                    %noone wants it, pass it on
                    use(to)=0; %don't reoffer the job
                    %find the neighbours
                    neigh=[];
                    for froms=1:length(to)
                        from=to(froms);
                        neigh=[neigh; net(net(:,1)==from,2)];
                    end
                    %who to pass to?
                    send=use(neigh).*(rand(length(neigh),1)<p_pass);
                    to=neigh(send>0); %make offers to valid people
                    flag=isempty(to); % stop if no offers
                end
            end

        end % job ends

        n_active(n,r)=length(active);
        jobs_all(n,r)=sum(jobs(:,r)>0)/w;
        jobs_active(n,r)=sum(jobs(active,r)>0)/length(active);
        if isempty(inactive)==0
            jobs_notactive(n,r)=sum(jobs(inactive,r)>0)/length(inactive);
        end

    end %rep ends

end
results.N_ww=N_ww;
```

```
results.wall=    sum(jobs_all,2)'/rep;         % Share of employed
results.jall=   sum(w*jobs_all/j,2)'/rep;      % Share of jobs taken
results.wactive= sum(jobs_active,2)'/rep;      % just active
results.jactive=sum(n_active.*jobs_active./offer_active,2)'/rep;
results.wnot=    sum(jobs_notactive,2)'/rep; % Just inactive
results.jnot=   sum((w-n_active).*jobs_notactive./(j-offer_active),2)'/rep;
toc
```