International Journal of Geographical Information Science
Vol. 24, No. 1, January 2010, 47–67

Research Article

Integration of GP and GA for mapping population distribution

YILAN LIAO†, JINFENG WANG*†, BIN MENG‡ and XINHU LI§

†Institute of Geographical Sciences and Natural Resources Research, Chinese Academy of Sciences, 11 Datun Road, Beijing 100101, China

‡College of Arts and Science, Beijing Union University, 197 Beitucheng Road (West), Beijing 100083, China

§Institute of Urban Environment, Chinese Academy of Sciences, 2 Huyuan Road, Xiamen 361003, China

(*Received 17 December 2007; in final form 1 May 2008*)

Mapping population distribution is an important field of geographical and related research because of the frequent need to combine spatial data representing socio-demographic information across various incompatible spatial units. However, the research may become very complex and difficult when a population in multiple places is estimated by various factors. Previous efforts in the field have contributed to the selection of appropriate independent variables and the creation of different population models. However, the level of accuracy obtainable with these studies is limited by the spatial heterogeneity of population distribution within the individual census districts, particularly in large rural areas. A high-accuracy modelling method for population estimation based on integration of Genetic Programming (GP) and Genetic Algorithms (GA) with Geographic Information Systems (GIS) is presented in this paper. GIS was applied to identify and quantify a set of natural and socioeconomic factors which contributed to population distribution, and then GP and GA were used to build and optimise the population model to automatically transform census population data to regular grids. The study indicated that the proposed method performed much better than the stepwise regression analysis and adapted gravity model methods in estimating the population of both urban and rural areas. More importantly, this proposed method could provide a single, unified approach to mapping population distribution in various areas because the paradigms of these algorithms are general.

Keywords: Mapping population distribution; Surface modelling; GIS; GP; GA

1. Introduction

The word's growing population presents humanity with increasingly difficult challenges with respect to global resources, the environment and sustainable development. Timely and accurate population estimation, its spatial distribution and its dynamics are important for understanding the effects of population increase on these social, economic and environmental problems. Moreover, population information at different levels, such as national, regional and local, is very significant for many purposes such as resource allocation, disaster relief and

*Corresponding author. Email: Wangjf@lreis.ac.cn

International Journal of Geographical Information Science
ISSN 1365-8816 print/ISSN 1362-3087 online
http://www.tandf.co.uk/journals
DOI: 10.1080/13658810802186874

infection control, etc. In general, population data are routinely collected by censuses and surveys and compiled according to political or administrative units. This form of census data, while essential for certain types of analyses, limits cross-disciplinary study. It is also difficult to maintain the data quality at current levels in view of the growth and migration of human populations. There is therefore a need to develop suitable techniques for estimating population in a manner in different spatial scales.

A number of methods have been proposed to map population distribution (Lo 2001; Nelson and Deichmann 2004). However, most of these only focus on the selection and quantification of independent variables and rarely take into account the correlation between selected variables. Much expertise is needed in modelling processes to formulate the relationships between independent variables and population data successfully. It usually not only produces information redundancy but also increases the complexity of the problem. In addition, it is very difficult for these methods to use a uniform model structure to estimate population in different grid cells because the form of interaction between independent variables and population distribution often varies from area to area.

Genetic Programming (GP) is an evolutionary technique and is gaining attention for its ability to determine the underlying data relationships and express them in a mathematical manner (Kishore *et al.* 2001). In GP, finding the functional form of the model can be viewed as being equivalent to searching the space of possible computer programs for the particular individual computer program which produces the desired output for given inputs (Koza 1990b). Genetic Algorithms (GA) is a derivative-free stochastic optimisation method based loosely on the concepts of natural selection, which was formally introduced in the 1970s by Holland (1975). Differing from conventional optimisation methods and search procedures, GA works by coding of the solution set and searching from a population of solutions based on probabilistic transition rules.

This paper proposes an approach for estimating population based on integration of GP and GA techniques with GIS. The ultimate purpose was to create a general and automatic modelling mechanism for mapping population distribution in different places. We first assumed that a set of factors (slope, land-cover type at grid, spatial distribution of rivers and transport infrastructure, population values of neighbouring villages and their spatial distribution) were likely to influence population distribution. After GP was applied to eliminate non-functional factors and create a model structure closest to the truth, GA was used to optimise parameters in the GP model. According to the GP&GA model, a gridded population map of the study area was finally generated. The result demonstrated that the GP&GA-based method did not require any a priori knowledge about how the factors influenced the population distribution and had much better performance than stepwise regression analysis and adapted gravity model approaches.

2. Interpolation of population data

Practically, mapping population distribution is a spatial interpolation problem that can be stated as follows: given a set of population data either in the form of discrete points or for subareas, how can one find the function that will best represent the whole surface and predict values at other points (Lam 1983). Since the publication of the first population density isopleths map in 1857 (Robinson and Sale 1971), interpolation methods of population data have developed quickly (Tobler 1979; Goodchild and Lam 1980; Flowerdew and Green 1989; Dodson *et al.* 2000). These

methods usually combine regular population censuses with vital registration data to simulate population distribution and manipulate the whole process within a framework of administrative units. In giving a systematic review of the approaches for estimating population distribution within administrative units, Deichmann (1996) classified them into areal interpolation and surface modelling.

However, a significant problem in these methods is how to represent 'real world' population distribution as accurately as possible. Areal interpolation is the process of transforming population data between various areal units (Goodchild and Lam 1980). This zone-based method is convenient to apply, but it implies that the phenomenon under consideration is equally distributed over the zone, which tends to obscure local specificity, diversity and intra-unit variation. In contrast to viewing data as discrete zones or bounded units, surface modelling of population distribution is aimed at formulating the population in a regular grid system, in which each grid cell contains an estimate of total population that is representative for that particular location (Yue *et al.* 2005). This method allows for greater spatial detail than is available from the zonal data alone (Bracken and Martin 1995). It also takes account of selection of a unit of analysis based on theoretical considerations without the limitation of the census geographical hierarchy (the researcher could draw boundaries anywhere). Furthermore, the development of novel types of analysis not available with the zone-based data becomes possible (measures of distance, spatial differentiation) (Zola and Frank 2001).

The key to build population surface models is how to disaggregate population data into grid cells. Bracken and Martin (1989) used the inverse distance weighted (IDW) method to develop population surfaces for census enumeration districts in the UK. They first assigned population counts to district centroids and then interpolated empty cells by moving windows in which the population of an empty cell was the weighted sum of all the centroids included in the window with closer centroids having greater 'weight' value and *vice versa*. This approach, however, is over-simplified, and its accuracy needs to be improved.

Meanwhile, use of satellite imagery as additional geographical information provides a different approach. In this approach, linear analysis is usually applied to develop models for estimating population quantities. For instance, Lo (1995) developed four linear regression models based on multispectral SPOT imagery to estimate the population and dwelling unit numbers in 44 tertiary planning units (TPUs) in Kowloon, Hong Kong. Harvey (2002) introduced a variety of standard spectral transformations of Landsat TM Imagery into regression models for population estimation in Ballarat, Sydney, Australia.

Stepwise regression is a common algorithm of linear analysis. Li and Weng (2005) used stepwise regression analysis to develop models for estimating population quantities in Indianapolis, IN, USA. In the process of modelling, remote sensing variables were sequentially added to and removed from the model using the list of candidate variables (original ETM+ bands, principal components, vegetation indices, fraction images, temperature and textures) until the model could not predict the population data any more either by adding or by removing a single variable. The number of exponents then became much greater than that using other possible means. The study demonstrated that this algorithm was good for identifying suitable variables for developing a population estimation model. However, it required the model structure to be specified and model coefficients to be determined in advance, which was commonly difficult to do, especially for low and high density regions.

In addition to extracting ancillary information or pixel characters from satellite imagery, researchers apply plenty of natural and socioeconomic data to mapping population distribution. The global demography project of National Centre for Geographic Information and Analysis (NCGIA) is a typical application of these approaches. The gravity model has been used by the project to create continental-scale population databases for Africa, Asia and Latin America, with support from the United Nations Environment Program, the International Center for Tropical Agriculture (CIAT) and others. The implementation of the gravity model is commonly based on the assumption that people tend to live in or close to cities and tend to move toward areas that are well connected with urban centres. Even in rural areas, it is expected that densely populated areas are closer to transport infrastructures than more isolated areas, and higher densities are nearer cities than the hinterland. The stylised facts concerning the distribution of people across space were implemented using the concept of accessibility, a measure of the ease by which destinations such as markets or service centres could be reached from a given location. The applications represented the sum of indicators of size or mass at the destinations (such as population of surrounding cities) inversely weighted for some function of distance (Balk *et al.* 2006). In this measure, deciding what to select into the models was also a difficulty. Bias was easily introduced if the focus of the analysis was on one of the factors used in the model.

Although previous research has indicated that populations with high density were often underestimated and those with low density were often overestimated (Harvey, 2000), no suitable solution has been proposed to correct these errors. There is still a need to invent computer-based tools which can automatically derive population models which are closest to the real population distribution in various places.

3. Methodology

As is well known, generating a gridded population map often consists of three basic steps: (1) creating a surface of weighting factors in a regular grid system for the study areas; (2) adjusting the basic weights derived in the first step using auxiliary data sources; (3) distributing the total population in the study areas to the corresponding grids in proportion to the weights constructed in the previous steps (Yue *et al.* 2003). In these steps, the most important issue is the size and shape of the model of population distribution. That is, one should first find the functional form of the model that best fits the observed empirical data, and only then go on to find any constants and coefficients that happen to be needed. A general framework for this study was designed as shown in Figure 1. The system comprised three parts, namely, data input, model structure selection and model parameter optimisation from left to right.

In the beginning, a number of natural and socioeconomic factors modified from those recommended by Dobson *et al.* (2000), Yue *et al.* (2003) and Nelson and Deichmann (2004) were selected and assumed to have some relation with the population distribution. Corresponding data layers were gathered and input into the GIS database. GIS software was used to calculate original values of these factors. Then, the relationship between population distribution and the input variables was formulated by GP. Mathematically, the relationship can be expressed as

$$popu(x) \rightarrow (x_1, x_2, \ldots, x_n) \tag{1}$$

where $popu(x)$ was the estimated population value in a grid and $x_1, x_2, \ldots, x_n$ were the

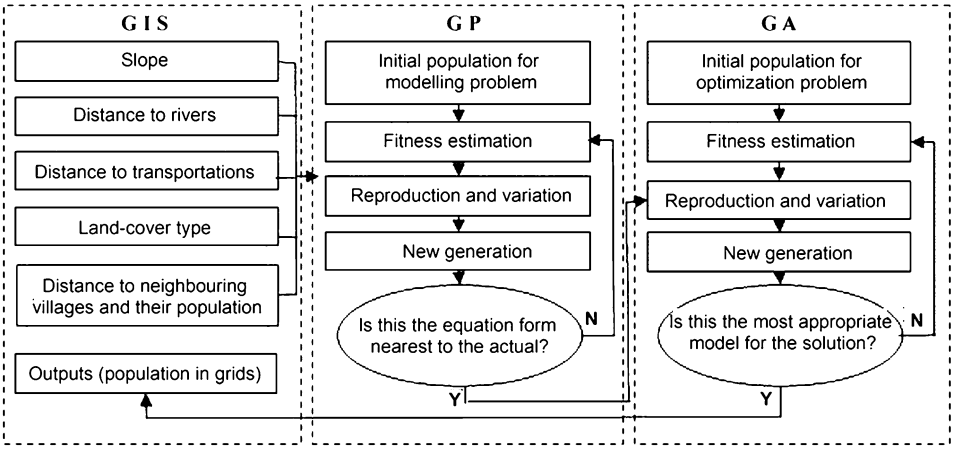

Figure 1. Integration of GP, GA and GIS for mapping population distribution.

normalised values of various factors in the grid. During the process of modelling, GP produced a computer program which took the factor variables as input and created the structure for the model of population distribution as output. The resulting model structure (computer program) evolved by GP was the one that best fitted the observed empirical data. Because the search space of GP is too large to optimise a specified node of computer programs, it is often difficult to evolve appropriate constants as part of the solution. So a number of techniques to supplement the tuning of constants in an evolved equation have been used to augment the evolutionary process (Whigham and Keukelaar, 2001). In the system, GA was applied to find the values of certain coefficients and constants required by the GP model so as to achieve the best fit between the observed data and the model. Individuals in GA were directly composed of original variables of the GP model and their evaluation criterion was how close the estimated results were to the actual. In the end, populations in census districts were allotted to grids in the GP&GA-based model.

3.1 *Preparing spatial data under GIS environment*

GIS plays a significant role in data processing and has insuperable advantages over traditional methods (Huang *et al.* 2004). It allows the addition of relevant layers which can be used for analysing the spatial relationships among selected factors. Also, it offers database capabilities that can handle attributes data effectively. Attribute calculations are simple and relatively accurate. We used ARCGIS 9.0i and Geoda 095i as the GIS platform to quantify the selected factors.

3.2 *Creating the population model using GP*

The key step in building population models is to create a model structure. A rapidly developing method for solving modelling problems is based on Evolutionary Algorithms (EA). GP which is a member of the EA family can produce the inherently hierarchical results to solve the problem in a relatively economical way. One important feature of GP is the absence of preprocessing of inputs and the fact that the solution is expressed directly in terms of the functions and arguments from

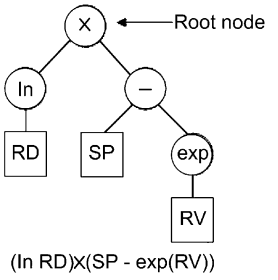

Figure 2. An example of GP parse tree representation.

the problem domain. This makes the results immediately comprehensible and intelligible in terms of the problem domain (Koza 1990a). Specifically, GP has been successfully applied in geographical and relative studies, such as predicting the density of an Australian marsupial (Whigham 2000), modelling land change (Manson 2005), downscaling daily extreme temperatures (Coulibaly 2004) and mapping landslide-hazard zones (Listchert 2004).

3.2.1 **Initial group.** A key step of GP is to create an initial group, which consists of a number of potential solutions (computer programs). In this study, every computer program had a tree-based structure and each internal node of this tree was a function node taking one of the values from the set $\{+, -, \times, /, \exp, \ln\}$. Moreover, the leaves below these nodes were placed within the terminal set which includes influencing factor variables and random constants. An example of such a parse tree with the expression that it represents is given in Figure 2. The function set nodes are represented by circles and the terminal set nodes by rectangles. This expression was evolved for a population in grids, i.e. for estimating *popu*(x), SP, RD and RV representing slope, distance to transport infrastructure and rivers, respectively. The 'tree depth' of this expression was 4, where 'tree depth' was the length of the longest path from the 'root node' of the tree (Figure 2) to the selected node.

The closure property was maintained by ensuring that all possible arrangements of the expressions would lead to a computer program which could be evaluated without error. The method of creating the initial group was ramped half and half, which permitted half the group to be created with a ramped variable (where the computer program could be of a size or structure up to the maximum depth specified for its creation) and the other half to use ramped grow (where only the creation mechanism could choose functions until the maximum depth was reached when a terminal had to be chosen). After the initial group has been created, GP goes into a loop of evaluation, selection and modification (Zhang *et al.* 2005).

3.2.2 **Fitness evaluation.** The simulation of natural selection in GP depends on the fitness function, which decides various evolutionary operations by calculating the fitness value of each computer program. Therefore, the fitness function is closely related to the convergence speed and accuracy of GP.

For the problem in this paper, the coefficient of determination (R^2) of the actual population data and population estimates from each computer program was introduced to evaluate the effectiveness of potential solutions. Because the actual population data of each grid was unavailable in the study, we had to specify the population data of each village for the fitness calculation. Suppose that the study area was composed of K villages and also could be divided into N grids. Let *Size_GP*

be the group size of GP and *Gen_GP* be the genetic generations. The fitness value *BsJi*(*i*, *t*)_*GP* of the computer program *Kpid*(*i*)_*GP* ($1 \leqslant i \leqslant Size\_GP$) in the *t*th ($1 \leqslant t \leqslant Gen\_GP$) generation could be defined as

$$BsJi(i, t)\_GP = \frac{\sum_{j=1}^{K} \left(P(j) - \overline{P}\right)\left(P'(j) - \overline{P}\right)}{\sqrt{\sum_{j=1}^{K} \left(P(j) - \overline{P}\right)^2 \sum_{j=1}^{K} \left(P'(j) - \overline{P'}\right)^2}} \tag{2}$$

where $\bar{P}$ and $\overline{P'}$ indicated average values of the actual and simulated population in all villages respectively, $P(j)$ was the actual population value of village $j(1 \leqslant j \leqslant K)$, and $P'(j)$ was the simulated population value of village j with computer program *Kpid*(*i*)_*GP*, which was calculated by the equation

$$P'(j) = \sum_{g=1}^{GN} popu(g, j) \tag{3}$$

where *GN* was the number of grids in village *j*, popu(*i*, *j*) was the simulated population value of grid $g(1 \leqslant g \leqslant n)$ in village *j* which was determined by the computer program *Kpid*(*i*)_*GP*. From the fitness equation, we could find that a higher fitness value means a better solution.

3.2.3 **Selection and reproduction.** Based on the fitness value, a computer program from the group was selected for further modification. A number of computer programs from the group were randomly selected, and then the fitness of members of this group are compared with each other. Finally, the actual best replaced the worst. Note that all the selected computer programs returned to the current generation after every selection. Therefore, some high-fit computer programs could be selected or copied many times. In this way, the computer program with the best fitness would, on average, be reproduced more often than the lower fitness computer program. This adheres to the Darwinian principle of 'survival of the fittest'.

3.2.4 **Evolutionary operators.** The selected program was then modified by evolutionary operators and encapsulated into the next generation. Crossover and mutation are the simplest, yet the most useful evolutionary operators (Zhang *et al.* 2005).

In crossover, two computer programs (the highlighted subtrees in Figure 3(a)) were selected and one point on each was taken as a swapping point. Each subtree from this point was exchanged with the other to create the offspring (bottom trees in Figure 3(a)). The assumption was that high-fit computer programs were composed of 'building blocks' which could be reshuffled with positive effect. The mutation operator was implemented in a 'shrink' way. Shrink mutation only referred to the subtree below the selected node of a computer program. Once a node of one computer program (the highlighted subtrees in Figure 3(b)) was chosen randomly, the offspring of its subtree was moved into the position of the parent. Although it may reduce the diversity, shrink mutation was avoided to make the computer program grow from generation to generation. It was particularly useful to consider that how long some computer programs got as the evolutionary process continued.

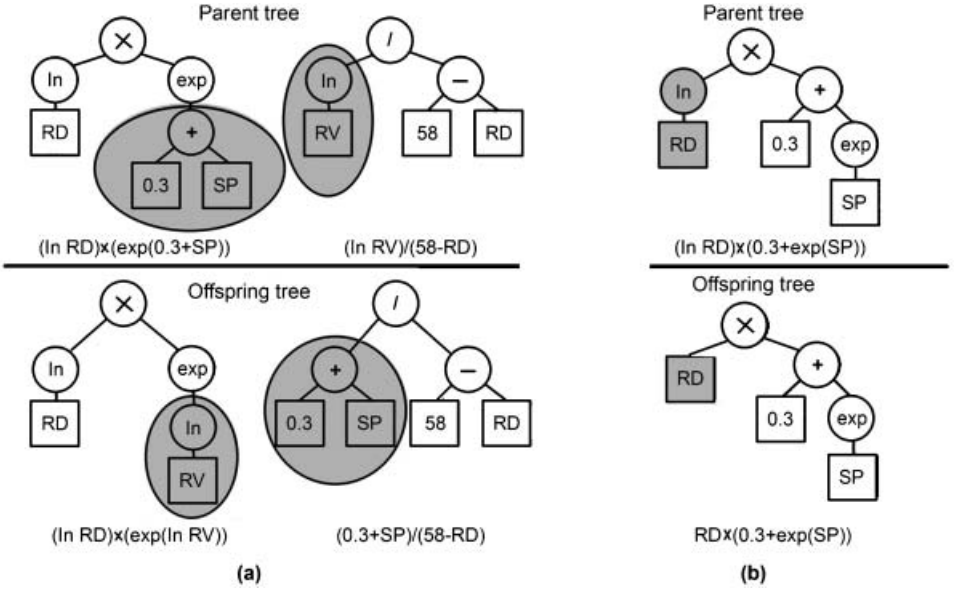

Figure 3. Examples of GP crossover (a) and mutation (b).

3.3 *Parameters optimisation using real-valued GA*

To improve GP model performance, optimisation of the model parameters was necessary. GA is a robust, domain-independent mechanism for optimisation and has a high probability of searching for optimal solutions in large and complex non-linear spaces. Although GP and GA use the same principles, representation is the biggest difference between these two algorithms. GA directly manipulates the coded representation of the problem, because the representation scheme can severely limit the window by which the system observes its word. However, a string-based representation scheme does not provide the hierarchical structure central to the organisation of computer programs (into programs and subroutines) and the organisation of behaviour (into task and subtasks). Moreover, the representation scheme does not provide any convenient way of representing arbitrary computational procedures or of incorporating iteration or recursion when these capabilities are inherently necessary to solve the problem (Koza 1990a). GP is a symbolic approach to program induction that overcomes the fixed-length limitations of standard GA approach.

In particular, GA is a mathematically near optimal approach to adaptation in the sense that it maximizes overall expected payoff when the adaptive process is viewed as a set of multi-armed slot machine problems for allocating future trials in the search space given currently available information (Koza 1990a). It has been used to solve optimisation problems in many geographic studies, such as path configurations design (Brookes 2001), the HAZMAT route plan (Huang *et al.* 2004) and optimal location search (Li and Yeh 2005).

3.3.1 **Real-valued coding.** Conventional GA generally represents trial solutions in the form of a discrete or binary string called a chromosome. For large problems, binary encoding results in very large strings which can slow down the evolution process. Moreover, if the length of the string is not long enough, it is only possible

for GA to get near the region of the global optimum rather than arrive at it. Therefore, conventional GA seems to have difficulties in fine tuning the parameters (Su and Chang 2000). In most physical areas, because the parameters involved in the optimisation problem are all real-valued, it is better to operate them directly in the original real-valued space instead of the discrete space. Thus, in our real-valued GA, each individual represented an *n*-dimensional vector which consisted of the parameters of the optimisation problem to be solved and the evolutionary operators merely acted on genes of individuals. This coding method was quite convenient for implementing the operators.

3.3.2 **Fitness evaluation.** The fitness function of GA was a little more complex. Similarly, let N be the total number of grids under study, K be village numbers, $Size\_GA$ be group size of GA, and Gen_GA be genetic generations. The raw fitness function of the individual $Kpid(i)\_GA$ in the tth ($1 \leqslant t \leqslant Gen\_GA$) generation was expressed as

$$BsJi(i, t)\_GA = \frac{1}{SERR(i, t) + \varpi} \tag{4}$$

where SERR(i, t) was the sum of square error of population estimation of all villages with individual $Kpid(i)\_GA$ and was taken as the performance index of the individual; while *?* was a constant, which was assigned 10^{-10} for the study in this paper. The simulated population in each village was likewise calculated according to equation (3).

To avoid the premature convergence problem in calculating fitness, we calculated the performance index sum of all individuals in the current generation $Total\_BsJ(t)$ by using the equation $Total\_BsJ(t) = \sum_{i=1}^{Size\_GA} SERR(i, t)$. Then with group size $Size\_GA$, GA diverged from and converged with the performance index of each individual in accordance with

$$SERR'(i, t) = \frac{(Size\_GA \times SERR(i, t))}{Total\_BsJ(t)} \tag{5}$$

Finally, the algorithm gave the eventual fitness function

$$BsJi(i, t)\_GA = \frac{1}{SERR'(i, t) + \varpi} \tag{6}$$

3.3.3 **Selection and reproduction.** Reproduction is usually the first operator applied to a group. It selects good individuals in a group and forms a mating pool so as to improve the chances of converging towards an optimal region. Because multiple copies of the good individuals are carried out, bad individuals are eliminated from the group for further consideration. Thus the reproduction operator is an exploitative operation for the good individual in the group. In our real-valued GA, we took the commonly-used reproduction operator, the proportional selection operator, where an individual in the current group was selected with a probability proportional to the individual's fitness.

3.3.4 **Evolutionary operators.** To exploit the potential of the current gene pool, we used crossover operators to generate new individuals in the hope of retaining

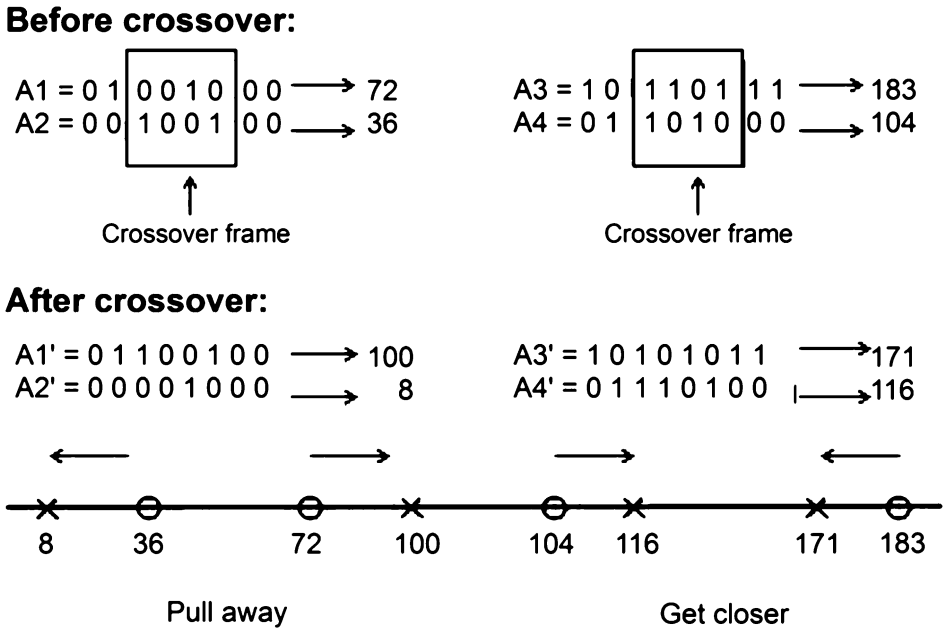

Figure 4. An example of a conventional two-point crossover mechanism.

positive features from the previous generation. There are two types of crossover operators: (1) one-point crossover where a crossover point on the gene code is selected at random and two parent individuals are interchanged at this point; (2) two-point crossover where two crossover points are selected and the part of the individual between these two points is then swapped to generate two offspring (Su and Chang 2000). Figure 4 shows the conventional two-point crossover mechanism undergoing crossing over. As can be seen, the crossover operator may bring two new individuals more close to each other. Thus, the crossover mechanism used in the study could be defined as follows

$$TempC(i)\_GA = \alpha \times Kpid(i+1)\_GA + (1-\alpha) \times Kpid(i)\_GA \tag{7}$$

$$TempC(i+1)\_GA = \alpha \times Kpid(i)\_GA + (1-\alpha) \times Kpid(i+1)\_GA \tag{8}$$

where $Kpid(i)\_GA$ and $Kpid(i+1)\_GA$ $(1 \leqslant i \leqslant Size\_GA-1)$ were selected individuals at the tth generation which were going to proceed to crossover. The two vectors $TempC(i)\_GA$ and $TempC(i+1)\_GA$ were the two offspring produced after crossing over $Kpid(i)\_GA$ and $Kpid(i+1)\_GA$. The parameter α was a random value in [0, 1].

By spontaneously generating new individuals, mutation then provided a mechanism for maintaining the group diversity. The conventional way of implementing mutation is to flip a bit with a probability equal to a very low given mutation rate. In our GA, since the individuals were represented as n-dimensional vectors, the mutation mechanism proceeded by adding a small or large amount of noise to the selected individual. It should be emphasized that the mutated individual would still lie in its domain space.

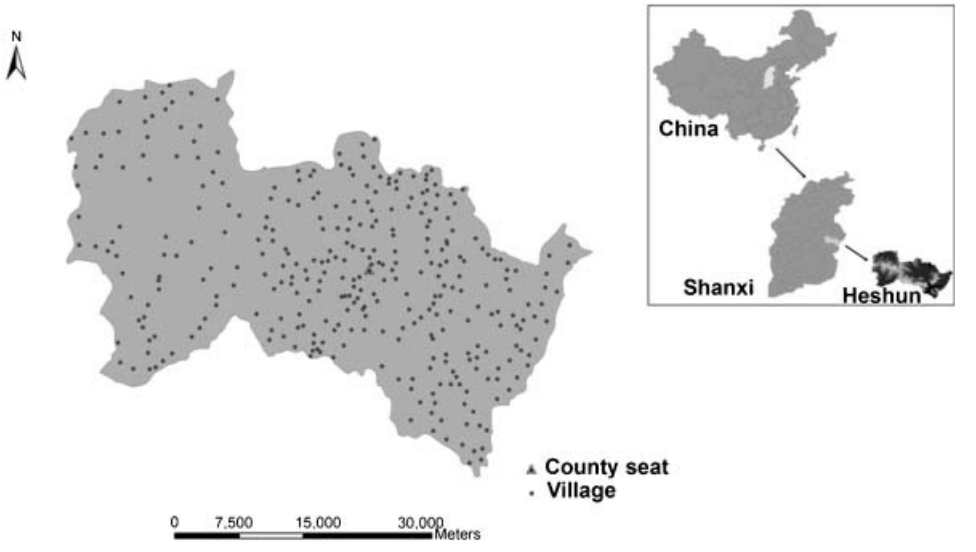

Figure 5. Location of Heshun.

4. Implementation and analysis results

4.1 *Study area and input variables*

The study area is located at Heshun, a county in Shanxi Province, China. The proposed method was tested by mapping the population distribution in Heshun County in 2001 based on some natural and socioeconomic factors. This study compared the effectiveness of applying the proposed GP&GA-based method, stepwise regression analysis, and adapted gravity model for mapping population distribution.

Heshun lies in the east of Shanxi, at 37°03′ E and 113°05′ N (Figure 5). It is composed of 326 administrative villages and the area of the region is 2250 km^2. In 2001, the total population was 134,766 and the average population density was 59.896 persons/km^2. The region is sparsely populated, which is similar to most counties in the north of China. This work may be of great benefit to further research in mapping population in rural areas of China, and even the world.

The village census data (from the Heshun Statistics Department) was allocated to grids through a 'smart' interpolation based on the relative likelihood of population occurrence in the grids. Because Heshun is a sparsely populated area, the grid layer used in the mapping process had a cell size of 1 km^2 and a dimension of 75×30 cells (2250 data points). In our study, the probability coefficients of the grids related to the following factors (Figure 6):

- slope, weighted by favourability of slope categories;
- rivers, weighted by distance from grids to the nearest river;
- transport infrastructure, weighted by distances from grids to the nearest road and railway;
- land-cover, weighted by the population density in certain types;
- neighbouring villages, weighted by the distances from grids to neighbouring villages and the population of these neighbours, also including distances to the county seat and its population data.

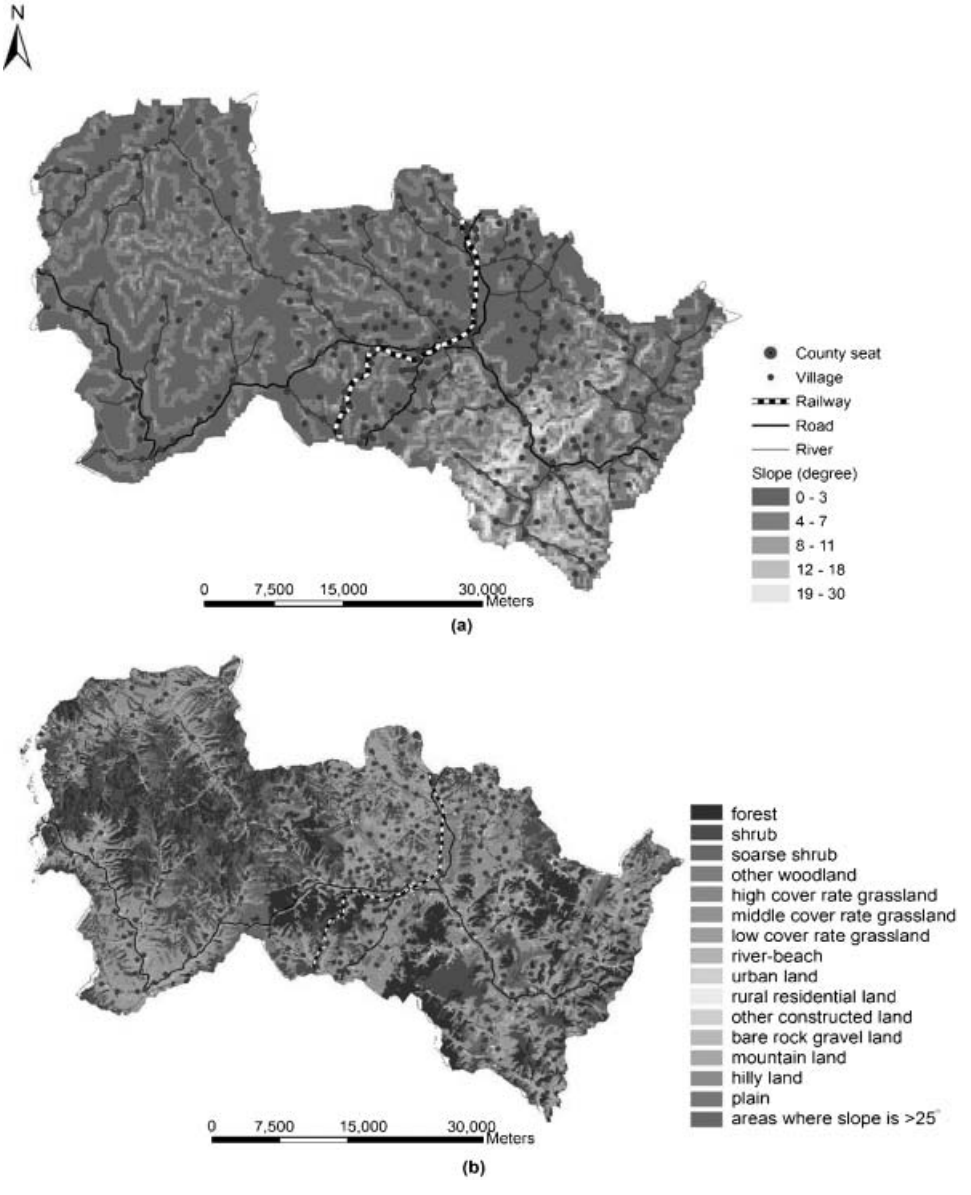

Figure 6. Distribution of transport infrastructure, rivers, slope (a) and land-cover types (b) in Heshun.

To accelerate the convergence of the proposed method, these resulting weighted values of different units were normalised and assigned to each value of each input variable, and a composite weighted value was calculated for each grid.

4.2 *The GP and GA-based method*

To ensure the accuracy of the algorithm, grids data of 261 villages (80% of the total number of villages) were randomly selected as training set to build the population model. The GP software tool used in this study was Gpc++ 0.40,

Table 1. Genetic programming parameters.

Parameter	Value
Group size	500
Generations	2000
Max creation depth	40
Max crossover depth	17
Crossover rate	0.98
Mutation rate	0.05
termination criterions	Maximum generation: 2000 $R^2 \geqslant 0.9500$

which was developed by Adam P. Fraser, Cybernetics Research Institute, University of Salford, Salford, UK. Gpc++ is a program package for finding functions on data. All used parameters were set as listed in Table 1. The parameters 'Max creation depth' and 'Max crossover depth' indicated the maximum size of the tree of the initial group and of the group from crossover, respectively. It was observed that a larger initial tree size often gave better results. This may be for the reason that the larger initial tree will lead to good initial exploration of the search space. The values of 'Max creation depth' and 'Max crossover depth' were thus constrained to 40 and 17. This restriction is necessary since GP has the tendency to produce uncontrollably large trees, if the tree size is not limited (Muttil and Lee 2005). Therefore, a maximum tree size of 17 evolved simple expressions that were easy to interpret. It was noted that there were two termination criteria for computing processes in both GP and GA: one was that the best fitness could achieve a certain value and the other was that the sum of generation exceeded the pre-specified number.

In the study, the GP program was run 100 times and the result is listed in Table 2. Because GP was capable of selecting input variables that contributed to the model, a measure of the significance of a variable in GP was the number of times the variable was selected. In this way, population densities in various land-cover types and distance to transport infrastructure were the most significant variables for this study.

Then, we chose the best GP model with the highest fitness in runs as the optimised object of GA

$$popu(i) = 22 - 3.24 \times \frac{\ln\left(\frac{road(i)}{205.5 \times lan\_cov(i) \times slope(i)}\right)}{\exp(0.01 \times nei\_vil(i))} \tag{9}$$

$$nei\_vil(i) = \sum_{j=1}^{EN} \frac{popu\_nei(j)}{Dis(i,j)} \tag{10}$$

Table 2. Number of input variable selections in 100 GP runs for population distribution in Heshun.

Input variables	Slope	Rivers	Land-cover	Transport infrastructure	Neighbouring villages	Total
No. of selection	18	6	71	58	25	178

where *popu*(*i*) was the simulated population of grid *i*; *slope*(*i*) was slope at grid *i*; *lan_cow*(*i*) was the population density of the land-cover type to which grid *i* belonged; *road*(*i*) was the distance from grid *i* to the nearest road; *nei_vil*(*i*) was the influence which the neighbouring villages exerted to grid *i*. According to equation (10), the value of variable *nei_vil*(*i*) was calculated by two factors: *Dis*(*i*, *j*), the distance from grid *i* to neighbouring village *j* and *popu_nei*(*j*), the population of the neighbouring village *j*. *EN* was the total number of neighbouring villages of grid *i*. Hence, a GA individual in the study comprised the above four factors, namely, *popu*(*i*), *slope*(*i*), *lan_cow*(*i*) and *nei_vil*(*i*).

In the evolutionary approach, the initial group of candidate solutions is generated randomly across the search space (Sastry *et al.* 2005). Each individual in the group is evaluated to determine its 'fitness', which decides how likely the individual is to survive and breed into the next generation (Li and Yeh 2005). Then, reproduction selects the good ones in a group according to their fitness and forms a mating pool. In this study, the reproduction method was fitness proportionate. Because the fitness proportions were rounded in computing, the individual number of the new group sometimes was not consistent with that of the previous one. The algorithm sorted the differences between the individual numbers before and after being rounded and added 1 in turn to those individuals whose losses were relatively greater until the differences became zero. New individuals were created by the operations of crossover and mutation. The values of crossover and mutation rates had a large influence on the performance of GA. If they were too big, then the optimising process had difficulty in convergence; if they were too small, premature convergence might occur, leading to erroneous conclusions. In conventional GA, crossover and mutation rates are often set to some fixed values. Much experience is required to set the values of crossover and mutation rates. To avoid this problem, genes of two individuals were randomly exchanged with probability P_{cr} (0.8–1) in the study. Besides, there was a criterion to decide whether to mutate the selected individual or not rather than giving a very low mutation rate. First, a finite value P_{mi} was assigned to a specified individual *Kpid*(*i*)*_GA* as

$$P_{mi} = \theta - Code\_po \times 0.01 / Size\_GA \tag{11}$$

where *Code_po* was the serial number of the individual *Kpid*(*i*)*_GA* in the individual fitness vector, which increased by its own fitness; and θ was a value in [0, 1]. According to equation (11), the greater the fitness, the smaller was P_{mi}. A random value (typically 0 or 1) was then assigned to every individual in the generation to compare with the correlating P_{mi}. If it was less than P_{mi}, the selected individual *Kpid*(*i*)*_GA* directly added noise which was a uniformly distributed random value in [−0.5, 0.5]. In this way, individuals with greater fitness seemed to have little probability of being mutated.

In GA, the group size is essential to improve the efficiency of the algorithm. The calculation speed may become slow when the group size is too large. Studies were carried out to determine the proper group sizes for GA. It was found that the group size of 150 could yield the highest value of the best fitness. This means that the group size of 150 was more effective in finding the optimal solution because it can generate the highest value. The improvement in the best fitness value stabilised after 1200 generations for all group sizes.

Similarly, the GA program was run 100 times and the final GP&GA model was

$$popu(i) = 28 - 2.86 \times \frac{\ln\left(\frac{road(i)}{172.5 \times lan\_cov(i) \times slope(i)}\right)}{\exp(0.002 \times nei\_vil(i))} \quad (12)$$

We only did a single training-test for the model in the study in order to simply validate the feasibility and accuracy of the proposed method. To obtain a more precise estimate of each model, one could apply a cross-validation with repeated training/test samplings. However, to some extent, this operation greatly increases the calculation and complication of the algorithms and results in prolongation of the running time.

4.3 *Results*

Comparison was made by applying the three methods, the stepwise regression analysis, the adapted gravity model approach and the GP&GA-based method, to estimate the population of 1×1 km grids in the remaining 65 villages. The resulting grid values were then aggregated to villages and compared to actual census data at the village level.

In our stepwise regression analysis, taking the census data being used for the regression function, six variables were selected, i.e. *slope*, *river*, *road*, *rail*, *lan*_cov and *nei_vil*. The regression equation was obtained as follows using the uniform training data

$$popu(i) = 49.31 + 543.876 \times land\_cov(i) - 25.792 \times slope(i) + 243.764 \times nei\_vil(i) \quad (13)$$

The multiple correlation coefficient (R) was 0.735, while the ratio (F) of regressive standard error (S') and residual standard error (S) was 574.305. The test of significance showed that the significance level of the regression equation was more than 95%, indicating that the established model equation (13) was reliable.

The simplest gravity models for population estimation are additive linear models of the form

$$popu(i) = \sum_{j=1}^{m} \frac{s_j}{(d_{ij})^{\phi}} \quad (14)$$

where $popu(i)$ is the population of grid i, s_j is the size of the city j, d_{ij} is the distance from grid i to city j, m is the total number of cities within the given searching extent and ϕ is exponent to be simulated. Besides socioeconomic factors, population distribution is greatly influenced by natural factors (Yue *et al.* 2003). Therefore, the adapted gravity model used in the study took the effect of the same six independent variables into account and was formulated as

$$popu(i) = 59.89 \times road(i)^{0.001} \times rail(i)^{0.002} \times slope(i)^{0.15} \times lan\_cov(i)^{1.03} \times river(i)^{0.007} \times nei\_vil(i)^{1.22} \quad (15)$$

where $river(i)$ and $rail(i)$ were respectively the distance from grid i to the nearest river and railway.

Table 3 illustrates the distribution of percentage differences between simulated populations and census data, using three different interpolation methods. It is easy to find from Table 3 that the overall correspondence of the GP&GA-based method

Table 3. Statistics of percentage differences in 65 villages.

Percentage difference	No. of villages		
	GP&GA-based model	Gravity model	Stepwise regression model
<-0.3	0	0	16
From -0.3 to -0.2	0	6	7
From -0.2 to -0.1	2	9	5
-0.1–0	20	13	6
0	22	4	2
0–0.1	18	16	7
0.1–0.2	3	11	6
0.2–0.3	0	6	6
>0.3	0	0	10

is such that 34% of the simulated village population corresponds with the official census. The table also indicates a difference of less than 10% ($\pm$) between the census data and the simulated population in 77% of the total of 65 villages, and most of these villages are aggregated to the middle and the east of Heshun and contain the majority of the population. However, in the adapted gravity model estimation, only 51% of villages show differences of less than 10 ($\pm$) and the rest show differences of 10–30% ($\pm$). The result of stepwise regression analysis is much worse. There are just 60% of villages where differences are less than 30% and differences in some villages reach 55% or more. This may be viewed as preliminary evidence that the GP&GA method is better in estimating accuracy.

Table 4 shows results from simple linear regression analysis on 'real' population counts of all 326 villages (Figure 7) and population estimates from each of the three methods. Three indices from the regression analysis are used for the comparison, i.e. regression coefficient (β), coefficient of determination (R^2) and mean square error (MSE) (Cai *et al.* 2006). The regression coefficients are all smaller than that with the largest value from stepwise regression algorithm. This might indicate a trend of underestimation in the interpolation methods (0.897 from the stepwise regression algorithm, 0.983 from the gravity model and 0.997 from the GP&GA). The values of R^2 from the three regressions are 0.805 for the stepwise regression algorithm, 0.936 for the gravity model and 0.973 for the GP&GA, indicating that the GP&GA is better than the gravity model method, and the gravity model method is better than the stepwise regression algorithm. The MSEs show the same order as the R^2. Thus, it is the finest interpolation of GP&GA model, rather than any fundamental error in the values of stepwise regression algorithm and the gravity model, which results in this highly favorable comparison.

Table 4. Results from regression analysis between 'real' population in 326 villages and population estimation from the three models.

Coefficient	Regression coefficient	Coefficient of determination	Mean square error
GP&GA-based model	0.997	0.973	801.132
Gravity model	0.983	0.936	3976.289
Stepwise regression model	0.897	0.805	22,794.268

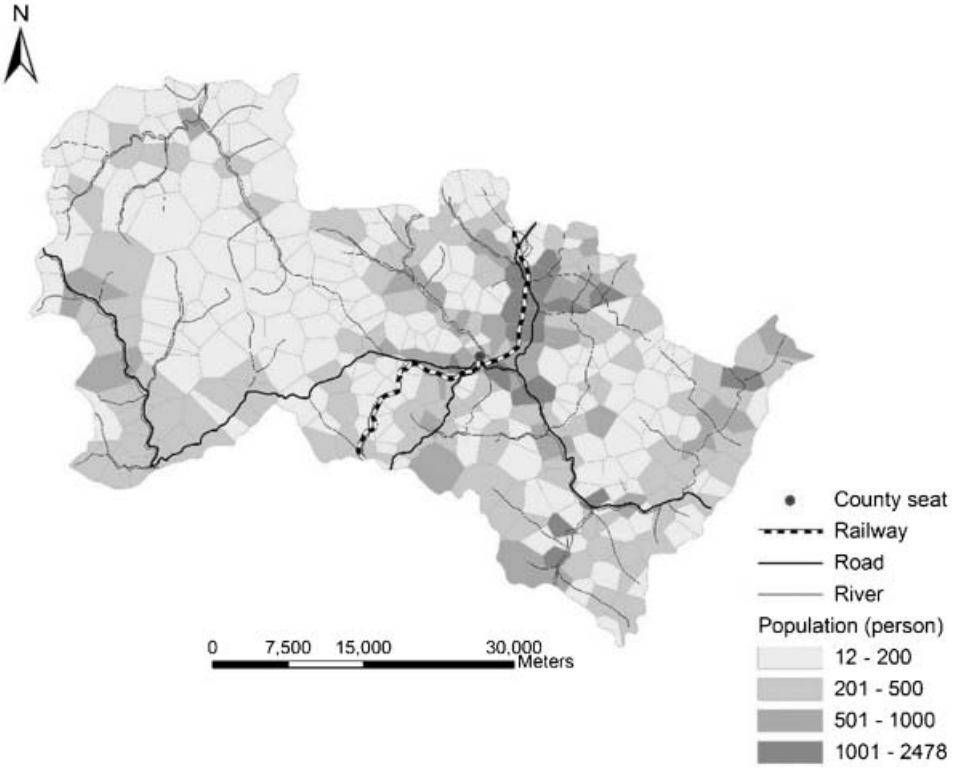

Figure 7. Population distribution of villages in Heshun (2001).

Figure 8 indicates that in some sparsely populated areas, in particular the western villages, the populations are significantly overestimated by all three methods. Obviously, as most of these areas are close to the transport infrastructure and/or relatively developed towns, small percentages of urban populations redistributed on rural grids result in such overestimations. Most substantial differences occur in villages whose populations are negligible at a regional scale. The most significant 'over-estimation' of simulated population is for Yixing, whose values for various factors are unusually good. As the county seat, it has a disproportionately larger number of administrative buildings compared with its resident population. Also, in transport and economic centres of the county (most are located in the middle of the region), populations may justify the higher simulated values relative to the official census counts.

To validate the efficiency of the three methods in estimating the population in high and low density areas at the same time, 326 villages in the study area were subdivided into 10 townships and 316 villages. From Figure 9(b), the percentage differences in the villages with the GP&GA model are much smaller than with the other models and the results tend to be consistent because most percentage differences of the model are in $[-0.10, 0.10]$. In this figure, the villages were coded based on their size. However, the other two methods performed better in some townships. In Weima, a southern transport centre in Heshun, the GP&GA model and the gravity model overemphasised the importance of the influence of the transport infrastructure and the percentage difference in the stepwise regression is the smallest. Although populations in 10 townships were all overestimated, the GP&GA model was the best of the three methods and its percentage differences were all below 0.15.

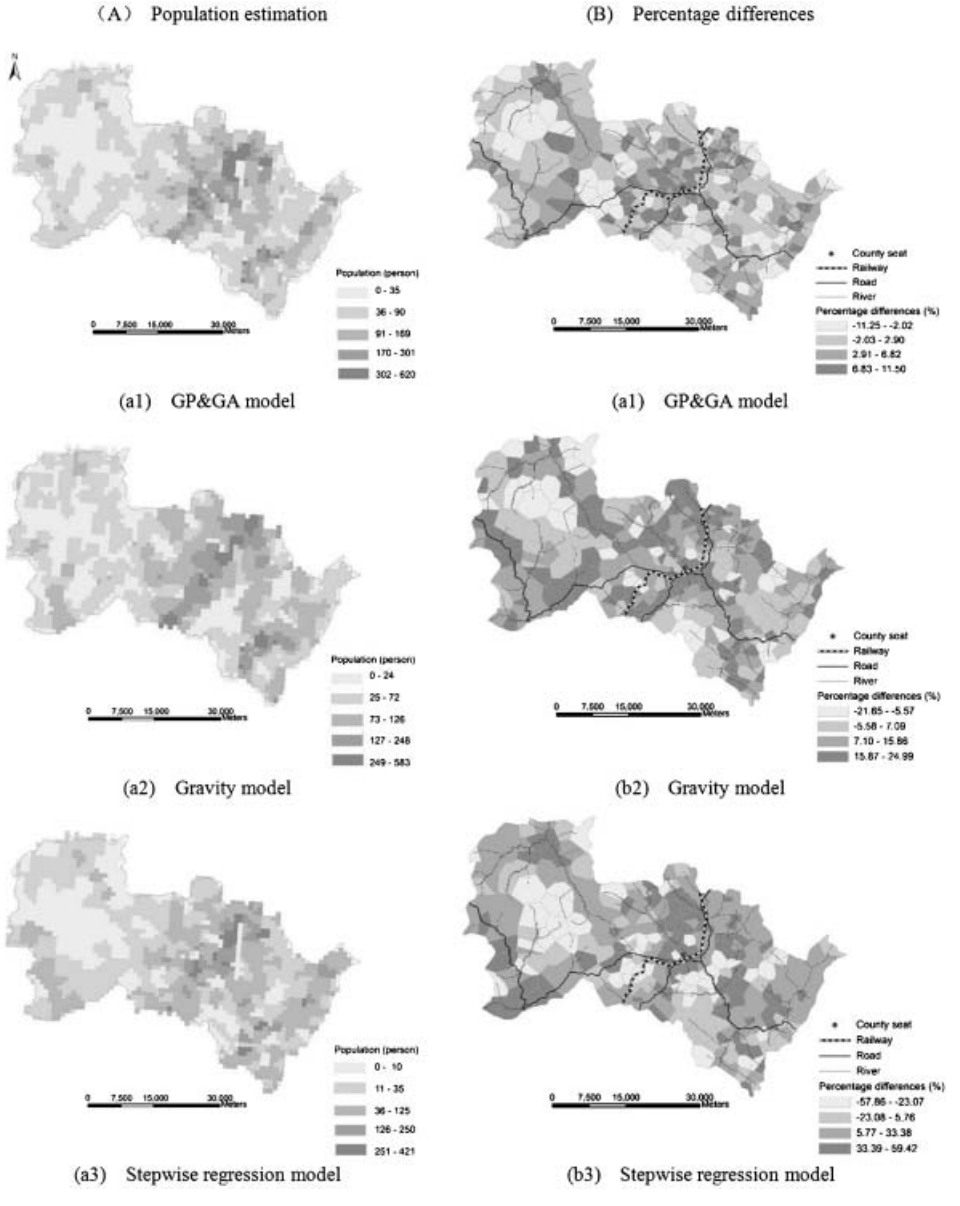

Figure 8. Population estimations and percentage differences from GP&GA model (a1 and b1), gravity model (a2 and b2) and stepwise regression model (a3 and b3).

5. Conclusions

Population estimation models based on the integration of independent variables and census data have numerous applications. These models can be used to provide knowledge of the size, behaviour and spatial distribution of the human population, which is useful for understanding many social and political processes and phenomena. However, which factors actually impact on the population distribution in a specific region? How do they affect it? None of these developed models performed very well to solve these problems.

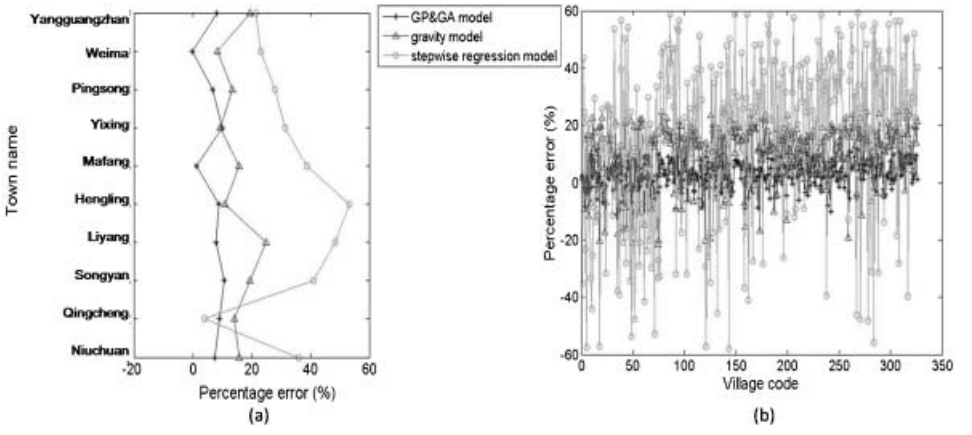

Figure 9. Percentage differences of population estimation in townships (a) and villages (b) from the GP&GA model, the gravity model and the stepwise regression model.

After reviewing issues related to the population model and population interpolation methods, we creatively introduced a GP&GA-evolved model combining various GIS derived variables to estimate population distribution. The whole modelling procedure consisted of two stages: the first stage focused on finding the most appropriate mathematical representation of the relationship between population distribution and independent variables; the second aimed to optimise the GP model by GA. In the implementation example, we selected slope, land-cover types, distance to rivers and to transport infrastructure, and influence of neighbouring villages as explanatory variables, and used three different population models to estimate population in 1×1 km grids. We then discussed the errors of these models in detail. The results indicated that, compared with the stepwise regression analysis and the adapted gravity model approaches, the GP&GA-based method not only overcomes the disadvantage of traditional mapping methods which usually require a priori knowledge about how to solve the problem, but also has better fault-tolerance and greatly enhances the calculation accuracy both in high and low density areas.

Although the proposed method has been tested only in Heshun, a rural region in the north of China, it can be used as a modelling tool to solve population interpolation problems on any scales as GP&GA is general and provides a single, unified method for addressing a variety of seemingly different problems in a variety of areas. Potential applications may include mapping the population distribution of the world, various administrative regions and different natural zones, etc. In addition, this method can simultaneously simulate population surfaces in a number of regions for automatic modelling and better accuracy.

At the same time, there are some problems in our method for future studies. As the popularity of the method increases, this will require more extensive data collection to ensure that the GIS database is accurate and up to date. Moreover, the method needs more cases to improve its efficiency and accuracy in mapping population because it has only been tested on one set of data, and at only one scale in the study. The compute process of the method should be further simplified. In addition, population distribution in different areas and periods are affected by different factors, thus creating difficulties in the selection of corresponding input variables. However, these

can be improved in the future by developing GIS and computer techniques and acquiring more knowledge of the spatial composition of the population. Therefore, the method provides flexibility regardless of the extent of study.

Acknowledgements

This work was supported by the Project of the National Natural Science Foundation of China (70571076 and 40471111), the Hi-Tech Research and Development Program of China (2006AA12Z15), the National Basic Research Priorities Program (2001CB5103) of the Ministry of Science and Technology of China and Knowledge Innovation Program of the CAS (KZCX2-YW-3-8). In addition, Professor Brian Lees (UNSW@ADFA, Australia) and six anonymous reviewers gave the useful comments on the drafts during the submission.

References

BALK, D.L., DEICHMANN, U., YETMAN, G., POZZI, F., HAY, S.I. and NELSON, A., 2006, Determining global population distribution: methods, applications and data. *Advances in Parasitology*, **62**, pp. 120–154.

BRACKEN, I. and MARTIN, D., 1989, The generation of spatial population distributions from census centroid data. *Environment and Planning A*, **21**(4), pp. 537–543.

BRACKEN, I. and MARTIN, D., 1995, Linkage of the 1981 and 1991 UK Censuses using surface modeling concepts. *Environment and Planning A*, **27**, pp. 379–390.

BROOKES, C.J., 2001, A genetic algorithm for designing optimal patch configurations in GIS. *International Journal of Geographical Information Science*, **16**(6), pp. 571–587.

CAI, Q., RUSHTON, G., BHADURI, B., BRIGHT, E. and COLEMAN, P., 2006, Estimating small-area population by age and sex using spatial interpolation and statistical inference methods. *Transactions in GIS*, **10**(4), pp. 577–598.

COULIBALY, P., 2004, Downscaling daily extreme temperatures with genetic programming. *Geophysical Research Letters*, **31**, pp. 1–4.

DEICHMANN, U., 1996, *A Review of Spatial Population Database Design and Modeling*, Technical Report 96-3 (Santa Barbara, CA: National Center for Geographic Information and Analysis).

DOBSON, J.E., BRIGHT, E.A., COLEMAN, P.R., DURFEE, R.C. and WORLEY, B.A., 2000, Landscan: a global population database for estimating populations at risk. *Photogrammetric Engineering and Remote Sensing*, **66**, pp. 849–857.

FLOWERDEW, R. and GREEN, M., 1989, Statistical methods for inference between incompatible zonal systems. In M.F. Goodchild and S. Gopal (Eds). *Accuracy of Spatial Database*, pp. 239–248 (London: Taylor and Francis).

GOODCHILD, M.F. and LAM, N.S.N., 1980, Areal interpolation: a variant of the traditional spatial problem. *Geo-Preocessing*, **1**, pp. 297–312.

HARVEY, J.T., 2002, Estimation census district population from satellite imagery: some approaches and limitations. *International Journal of Remote Sensing*, **23**, pp. 2071–2095.

HUANG, B., CHEN, R.L. and LIEW, Y.S., 2004, GIS and genetic algorithms for HAZMAT route planning with security considerations. *International Journal of Geographical Information Science*, **18**(8), pp. 769–787.

HOLLAND, J.H., 1975, *Adaptation in Natural and Artificial Systems* (Ann Arbor, MI: University of Michigan Press).

KISHORE, J.K., PATNAIK, L.M., MANI, V. and AGRAWAL, V.K., 2001, Genetic programming based pattern classification with feature space partitioning. *Information Sciences*, **131**, pp. 65–86.

KOZA, J.R., 1990a, Genetic Programming: *A Paradigm for Genetically Breeding Populations of Computer Programs to Solve Problems*, Stanford University Report, Report

No. STAN-CS-90-1394. Available online at: http://www.genetic-programming.com/jkpubs72to93.html#anchor484765.

KOZA, J.R., 1990b, A genetic approach to econometric modeling. In *Economics and Cognitive Science*, pp. 57–75 (Oxford: Pergamon Press).

LAM, N.S., 1983, Spatial interpolation methods: a review. *The American Cartographer*, **10**(2), pp. 129–149.

LI, G.Y. and WENG, Q.H., 2005, Using Landsat ETM+ imagery to measure population density in Indianapolis, Indiana, USA. *Photogrammetric Engineering and Remote Sensing*, **71**(8), pp. 947–958.

LI, X. and YEH, A.G., 2005, Integration of genetic algorithms and GIS for optimal location search. *International Journal of Geographical Information Science*, **19**, pp. 581–601.

LITSCHERT, S., 2004, Landslide hazard zoning using genetic programming. *Physical Geography*, **25**(2), pp. 130–151.

LO, C.P., 1995, Automated population and dwelling unit estimation from high-resolution satellite images: a GIS approach. *International Journal of Remote Sensing*, **16**, pp. 17–34.

LO, C.P., 2001, Modeling the population of China using DMSP operational linescan system nighttime data. *Photogrammetric Engineering and Remote Sensing*, **67**, pp. 1037–1047.

MANSON, S.M., 2005, Agent-based modeling and genetic programming for modeling land change in the Southern Yucatan Peninsular Region of Mexico. *Agriculture, Ecosystems and Environment*, **111**, pp. 47–62.

MUTTIL, N. and LEE, J.H.W., 2005, Genetic programming for analysis and real-time prediction of coastal algal blooms. *Ecological Modeling*, **189**, pp. 363–376.

NELSON, A. and DEICHMANN, U., 2004, *The African Population Database*, Version 4 (New York: United Nations Environment Program and the Center for International Earth Science Information Network, Columbia University) Available online at: http://www.na.unep.net/datasets/datalist.php3.

ROBINSON, A. and SALE, R., 1971, The genealogy of the isopleths. *Cartographic Journal*, **8**, pp. 49–53.

SASTRY, K., GOLDBERG, D. and KENDALL, G., 2005, Genetic algorithms. In E.K. Burke and G. Kendall (Eds). *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*, Chapter 4, pp. 97–125 (New York: Springer).

SU, M.C. and CHANG, H.T., 2000, Application of neural networks incorporated with real-valued genetic algorithms in knowledge acquisition. *Fuzzy Sets and Systems*, **112**, pp. 85–97.

TOBLER, W.R., 1979, Smooth pychophylactic interpolation for geographical regions. *Journal of the American Statistical Association*, **367**(74), pp. 519–530.

WHIGHAM, P.A., 2000, Induction of a marsupial density model using genetic programming and spatial relationships. *Ecological Modelling*, **131**, pp. 299–317.

WHIGHAM, P.A. and KEUKELAAR, J.K., 2001, Evolving structure, optimizing content. In C. Fonseca, J.H. Kim, A. Smith and X. Yao (Eds). *Proceedings of the 2001 Congress on Evolutionary Computation*, pp. 1228–1235 (Piscataway, NJ: IEEE Press).

YUE, T.X., WANG, Y.A., CHEN, S.P., LIU, J.Y. and QIU, D.S., 2003, Numerical simulation of population distribution in China. *Population and Environment*, **25**, pp. 141–163.

YUE, T.X., WANG, Y.A., LIU, J.Y., CHEN, S.P., QIU, D.S., DENG, X.Z., LIU, M.L., TIAN, Y.Z. and SU, B.P., 2005, Surface modeling of human population distribution in China. *Ecological Modeling*, **181**, pp. 461–478.

ZHANG, L., LINDSAY, B.J. and ASOKE, K.N., 2005, Fault detection using genetic programming. *Mechanical Systems and Signal Processing*, **19**, pp. 271–289.

ZOLA, K.M. and FRANK, L.F., 2001, Population density surface: a new approach to an old problem. *Society and Natural Resources*, **14**, pp. 39–49.