WSG Forschungsberichte

WSG-RR 6/96

SIM User's Manual.

A Flexible Toolbox for Spatial Interaction Modelling.

Manfred M. Fischer, Adrian Trapletti and Jinfeng Wang

Institut für Wirtschafts- und Sozialgeographie

Wirtschaftsuniversität Wien

Department of Economic and Social Geography

Vienna University of Economics and Business Administration

WSG Research Reports

WSG-RR 6/96

SIM User's Manual.

A Flexible Toolbox for Spatial Interaction Modelling.

Manfred M. Fischer, Adrian Trapletti and Jinfeng Wang

Abteilung für Theoretische und Angewandte Wirtschafts- und Sozialgeographie
Institut für Wirtschafts- und Sozialgeographie
Wirtschaftsuniversität Wien

Vorstand: o.Univ.Prof. Dr. Manfred M. Fischer
A - 1090 Wien, Augasse 2-6, Tel. (0222) 313 36 - 4836

Redaktion: Mag. Petra Staufer

WSG-RR 6/96

SIM User's Manual.

A Flexible Toolbox for Spatial Interaction Modelling.

Manfred M. Fischer, Adrian Trapletti and Jinfeng Wang

WSG-Research Report 6

January 1996

SIM User's Manual

A Flexible Tool Box for Spatial Interaction Modelling

Manfred M. Fischer
Adrian Trapletti
Jinfeng Wang

Department of Economic and Social Geography
Vienna University of Economics and Business Administration

February 1996

This Manual is for SIM Version 1.0
Email address for comments, suggestions and bug reports:
adrian@wigeo1.wu-wien.ac.at

Contents

1 Copyright

SIM version 1.0

2 Introduction

SIM is a flexible tool box for spatial interaction modelling running on UNIX workstations under X Windows with the Motif graphical user interface. The program combines a graphical input/output user interface with robust and efficient algorithms. The package was implemented in C and has been tested in research situations on a Sun SPARC 10 workstation at the Department of Economic and Social Geography of the Vienna University of Economics and Business Administration.

The tool box provides three major choice dimensions on spatial interaction modelling:

Model types: doubly constrained, production constrained, attraction constrained, unconstrained.

Separation functions: power function, exponential function, Tanner function, generalized Tanner function.

Estimation procedures: least squares estimation (ordinary and weighted) with odds ratio procedure, maximum likelihood estimation by simulated annealing combined with a downhill simplex method.

The structure of the user's manual is as follows. Section 3 is dealing with the installation of SIM, whereas in section 4 the required input and the input user interface are outlined. The outcoming results, statistics and the possibilities of visualizing these results are described in section 5. Section 6 gives an overview over the implementation of SIM. Section 7 contains a list of known limits, problems and bugs of SIM. In the appendix some basics of the theory of spatial interaction models, estimation techniques and performance statistics used are briefly described.

In this documentation `typewriter` style is used for user input, syntax specification and commands, *Italic* shape for options to commands, for file and directory names, for field and button names and **Boldface** series to emphasize text. Syntax specification is given in EBNF (Extended Backus-Naur-Form).

3 Installation

3.1 Main Installation

SIM is written for UNIX workstations under X Windows/Motif GUI. To use SIM comfortably the workstation should have at least 32 MB RAM.

The first step in the installation procedure is to decompress the file *sim.tar.gz* with the `gzip` command

```
% gzip -d sim.tar.gz
```

and to extract the source and documentation files with the `tar` command

```
% tar xvf sim.tar
```

Then change to the builded directory *sim*

```
% cd sim
```

The directory should contain the following files:

AUS.dat	Austrian telecommunication data
AbsErr.gnu	GNUPLOT macro
AbsRelErr.par	ACE/gr parameter file
Acknowledgement	
Copyright	
Flow3d.gnu	GNUPLOT macro
README	
RelErr.gnu	GNUPLOT macro
US.dat	US interregional migration data (example data from Fotheringham and O'Kelly, 1989)
XMsim	X11 application defaults file
estimation.txt	Specification file for estimation-actions
forecasting.txt	Specification file for forecasting-actions
info_textfile1.txt	Read only information panel file
makefile	Makefile to build and install SIM
mle.c	Maximum likelihood module
ols.c	Ordinary least squares module
sim.c	Main module
simdef.h	Header file for *.c files
user.ps	This guide in postscript
wls.c	Weighted least squares module

To compile SIM a C compiler and the C standard libraries as well as the additional libraries for the Motif toolkit and for X Windows have to be available. Our configuration uses the Gnu C compiler `gcc` as default.

Edit the makefile to ensure that all path settings, the libraries and the compiler settings are correct. SIM is build by

```
% make sim
```

and the executable *sim* as well as the X11 application defaults file *XMsim* are installed with

```
% make install
```

in the proper place.

3.2 User Specific Installation

The final actions undertaken by SIM in the model estimation and the forecasting mode are defined in the two files *estimation.txt* and *forecasting.txt*, respectively. Each line in these files is assumed to be a UNIX command except when a line is starting with `//`, which means that this line is a comment. The files provided allow a graphical representation of estimation and forecasting results by using the freely available software packages GNUPLOT and ACE/gr.

4 Input

SIM may be started by typing the command `sim` on the Unix command line. Then the **main menu** (see Figure 4.1) appears on the screen. The main menu contains the specification fields for the input data file, the model type, the separation function and the estimation procedure, as well as the buttons for starting estimation, starting forecasting and quitting SIM. On the right side of the window is a read only information panel.

The user needs to specify the number of origins, destinations and the data input file name in the appropriate text fields and to select the model type, the separation function and the estimation procedure by setting the appropriate toggle buttons.

The **data input file** consists of several lines, each of them describing a flow from one origin to one destination. The syntax of such a line is defined as follows:

```
Line = OriginCode "," DestinationCode "," ObservedFlow ","
       Distance "," OriginPropulsiveness ","
       DestinationAttractiveness.
```

An example is provided in the file *US.dat*. During estimation the data input file is used by SIM to calibrate the model and to produce a range of calibration results and diagnostic statistics (see appendix C), whereas in the forecasting mode the forecasts and out-of-sample performance statistics are computed.

Model estimation is started by clicking on the *Estimation* button. If the estimation procedure options ordinary (OLS) or weighted least squares (WLS) are chosen, SIM immediately calibrates the model, and computes the calibration results and diagnostic statistics (see section 5). Then SIM executes the user defined batch job (see section 5) and returns finally to the main menu.

In the case of maximum likelihood estimation (MLE) it is necessary to make some more specifications. For this purpose, SIM prompts a param-

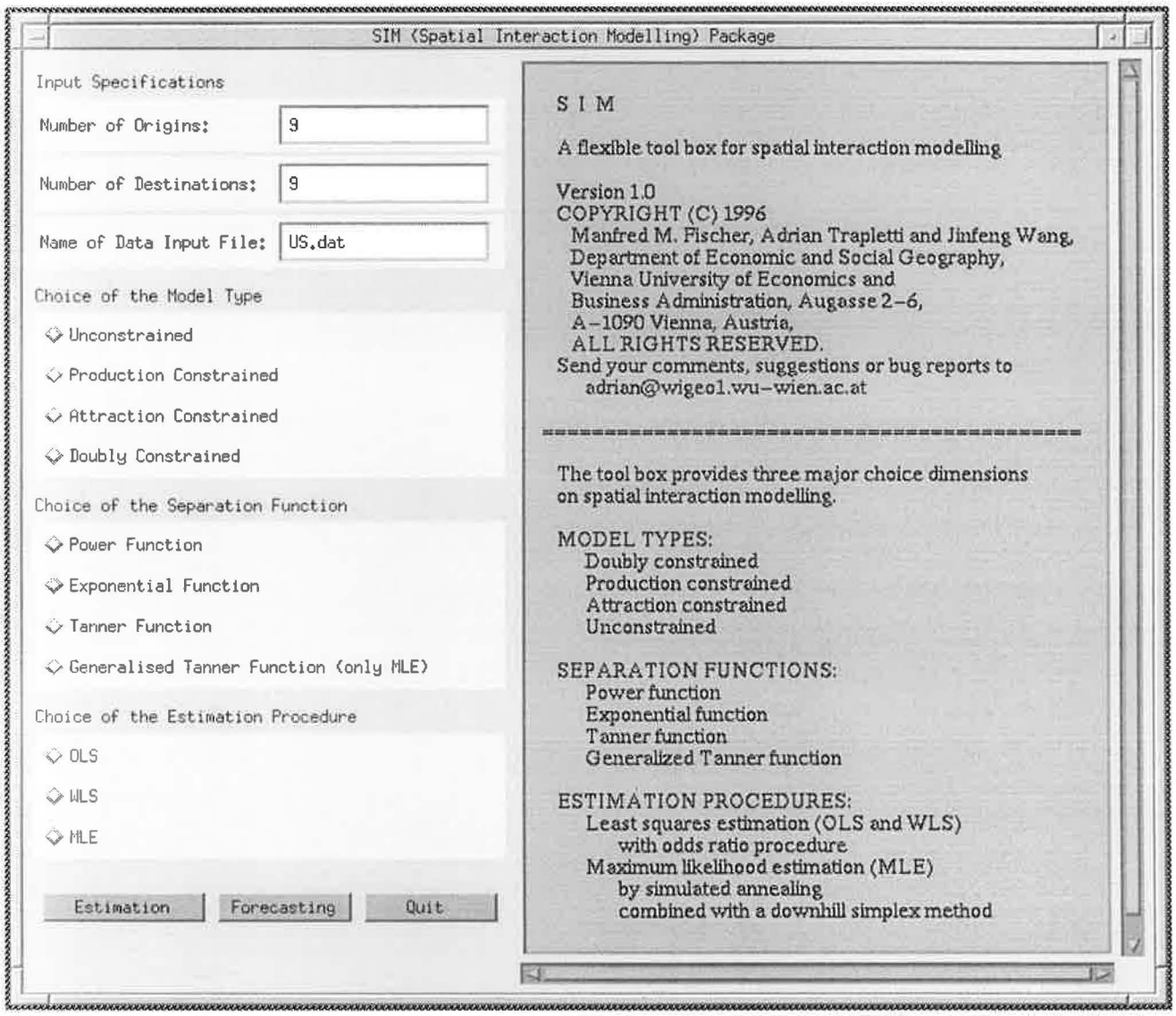

Figure 4.1: The main menu

eter setting window on the screen, where the annealing schedule for the optimization algorithm of simulated annealing has to be specified (see appendix B.2.2). An example of such a window is shown in figure 4.2. The annealing schedule specifies the starting temperature, the number of iterations computed at a certain temperature, the amount by which the temperature is decreased each time and the initial system state or in other words the initial values of the simplex matrix. The *Start Temperature*, *Temperature Decrease* and *Number of Iterations* fields are numerical fields, whereas the *Initial Values of Simplex Matrix* field specifies a matrix, defined by the

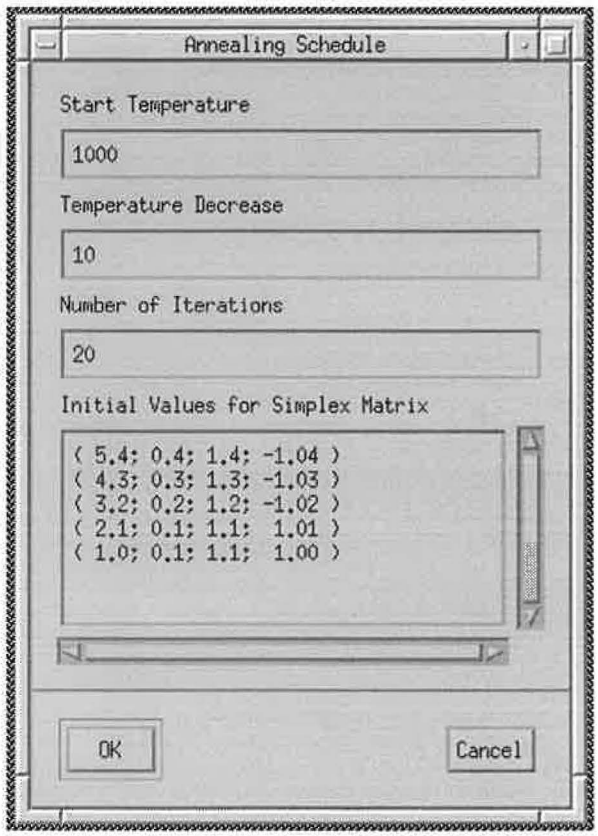

Figure 4.2: The parameter setting window

following syntax rules:

```
Matrix = Line { Line }.
Line   = "(" Float { ";" Float } ")".
```

In the case of a

1. constrained model combined with
   (a) the power or exponential separation function a (2,1) matrix,
   (b) the Tanner separation function a (3,2) matrix and
   (c) the generalized Tanner separation function a (4,3) matrix

   has to be specified by the user.
2. For the unconstrained model combined with
   (a) the power or exponential separation function a (5,4) matrix,

(b) the Tanner separation function a (6,5) matrix and

(c) the generalized Tanner separation function a (7,6) matrix

has to be specified.

SIM provides default matrices, but of course these default matrices are not optimal in every situation, in which case the user can specify his own choice. As in all applications with simulated annealing, success or failure is quite often determined by the choice of the above mentioned parameters. Often only a slight change in the initial matrix is enough to receive better calibration results. Finally, to start the calibration, click on the *OK* button. Then SIM starts calibrating the model, computing the calibration results and diagnostic statistics, executes the user defined batch job (see section 5) and returns finally to the main menu.

In order to choose the **forecasting mode** the user has to click the *Forecasting* button. This causes SIM to compute the forecasting results and the out-of-sample performance statistics, and to execute the user defined batch job (see section 5). Then SIM returns to the main menu. Of course forecasting can be started only after estimation, in which case the last estimated model is used to forecast. To avoid inconsistencies between estimation and forecasting mode, the user selected options, specified by the three toggle buttons, must be the same for estimation and forecasting.

To quit SIM simply click the *quit* button and SIM immediately stopps.

5 Output

In terms of output, SIM basically provides the file *output.dat*, which contains all the calibration results and diagnostic statistics. The most recent informations about the actions currently taken by SIM are written to the standard output. An example of *output.dat* is given below:

```
############ ESTIMATION  ############
#Model Type: Production Constrained Model
#Separation Function: Power function
#Estimation Procedure: OLS
#
#I=   3
#J=   3
#Code_i    A2[i]_BALANCE_FACTOR
#        1          0.001478953
#        2          0.001643365
#        3          0.001450058
#
#ORIGIN DESTINATION OBSERVED_FLOW PREDICTED_FLOW ABSOLUTE_ERROR RELATIVE_ERROR
```

1	2	100.0	120.65	-20.65	-0.21
1	3	90.0	69.35	20.65	0.23
2	1	100.0	141.12	-41.12	-0.41
2	3	300.0	258.88	41.12	0.14
3	1	90.0	93.04	-3.04	-0.03
3	2	300.0	296.96	3.04	0.01

```
#
#The number of origin-destination pairs: 6
#
#The total interactions observed: 980.0000
#The total interactions estimated: 980.0000
#
#Percentage deviation of observed interactions
#from the mean (163.3333): 0.5578
#
#Percentage deviation of estimated interactions
#from the observed interactions: 0.1323
#
#Regressing the observed interactions on the predicted
#interactions yields RSquared(2) value of:0.9322
#
#T_real[i][j] = -16.6977 + 1.1022 * T_esti[i][j]
#                 t-value of intercept: -0.6108
#                 t-value of slope: 0.6881
#
#Distance parameter(power): -0.2904
#
#Optional Choice for the Performance Statistics
#RMSE_Statistic: 26.6248
#
#Standardized RMSE-Statistic: 2.7168
#
#ARV-Statistic: 0.0758
#
#R Square (1)-Statistic: 0.9242
#
#R Square (1)-adjusted Statistic: 0.9242
#
#R Square (2)-Statistic: 0.7673
#
#R Square (2)-adjusted Statistic: 0.7673
```

```
#
#FW-Statistic: 0.2327
#
#FW-adjusted Statistic: 0.2327
#
#Information Gain Statistic: 14.5337
#
#MDI: 0.0148
#
```

In addition SIM makes it possible for the user to define his own **batch jobs.** Execution of these jobs is started after writing the calibration results and diagnostic statistics to the file *output.dat.* The batch jobs have to be specified by the user in the files *estimation.txt* and *forecasting.txt.* Using the default files a graphical representation of the calibration results is provided by reading the file *output.dat* and using GNUPLOT or ACE/gr. See figure 5.1 for an example of such a graphical representation.

The definitions of the performance measures written to *output.dat* are given in appendix C.

6 Implementation

The implementation is structured in four modules:

1. The **main module** *sim.c* handling the input and output user interface,
2. the **ordinary least squares module** *ols.c* responsible for the ordinary least squares calibration of the models and the computation of diagnostic statistics,
3. the **weighted least squares module** *wls.c* responsible for the weighted least squares calibration of the models and the computation of diagnostic statistics, and
4. the **maximum likelihood module** *mle.c* responsible for the maximum likelihood calibration of the models and the computation of diagnostic statistics.

Sim.c handles the main menu and all subwindows, i.e. installs the windows and starts up the event handlers, reads in the input data file, calls one of the three calibration modules, and finally starts, after receipt of the control from *ols.c*, *wls.c* or *mle.c*, the processes defined in *estimation.txt* and *forecasting.txt.*

Ols.c, *wls.c* and *mle.c* receive the control from *sim.c*, calibrate the model in the estimation mode, produce the calibration results and diagnostic statistics, and write the results to *output.dat.*

Figure 5.1: The graphical representation of calibration results

Figure 6.1 gives a schematic overview of the interactions between the different modules and main routines in SIM. For more details about the implementation of SIM, we refer the reader to the provided source files.

7 Limits, Problems and Bugs

Limits might occur due to the user's specific hardware and software configurations. On a Sun SPARC 10 with 64 MB RAM, a standard SunOS installation and using the Gnu C compiler up to 1000 origins and destinations can be considered without difficulties.

We should mention that some problems might arise when using GNUPLOT and ACE/gr at the same time to produce graphical calibration results of SIM.

In order to work correctly in cooperation with programs like GNUPLOT and ACE/gr and to give SIM enough memory at the same time, one needs to limit the stacksize to a certain amount. In our configuration we used

```
% limit stacksize 40m
```

Please report any comments, suggestions and bugs to
adrian@wigeo1.wu-wien.ac.at.

8 Acknowledgement

This work has been supported by a grant from the Austrian Fonds zur Förderung der Wissenschaftlichen Forschung (P-09972-TEC).

Appendix

In the sequel $i = 1, \ldots, I$ denotes the origin zone and $j = 1, \ldots, J$ the destination zone, where the number of origin zones I may be unequal to the number of destination zones J.

A Spatial Interaction Models

A.1 General Model

The general formula for spatial interaction models is

$$\hat{T}_{ij} = V_i \, W_j \, F_{ij}, \tag{1}$$

where $\hat{T}_{ij}$ is the estimated size (volume) of a flow (e.g. people, goods, money or information) from zone i to zone j, V_i denotes the origin factor, W_j the destination factor and F_{ij} the spatial separation factor.

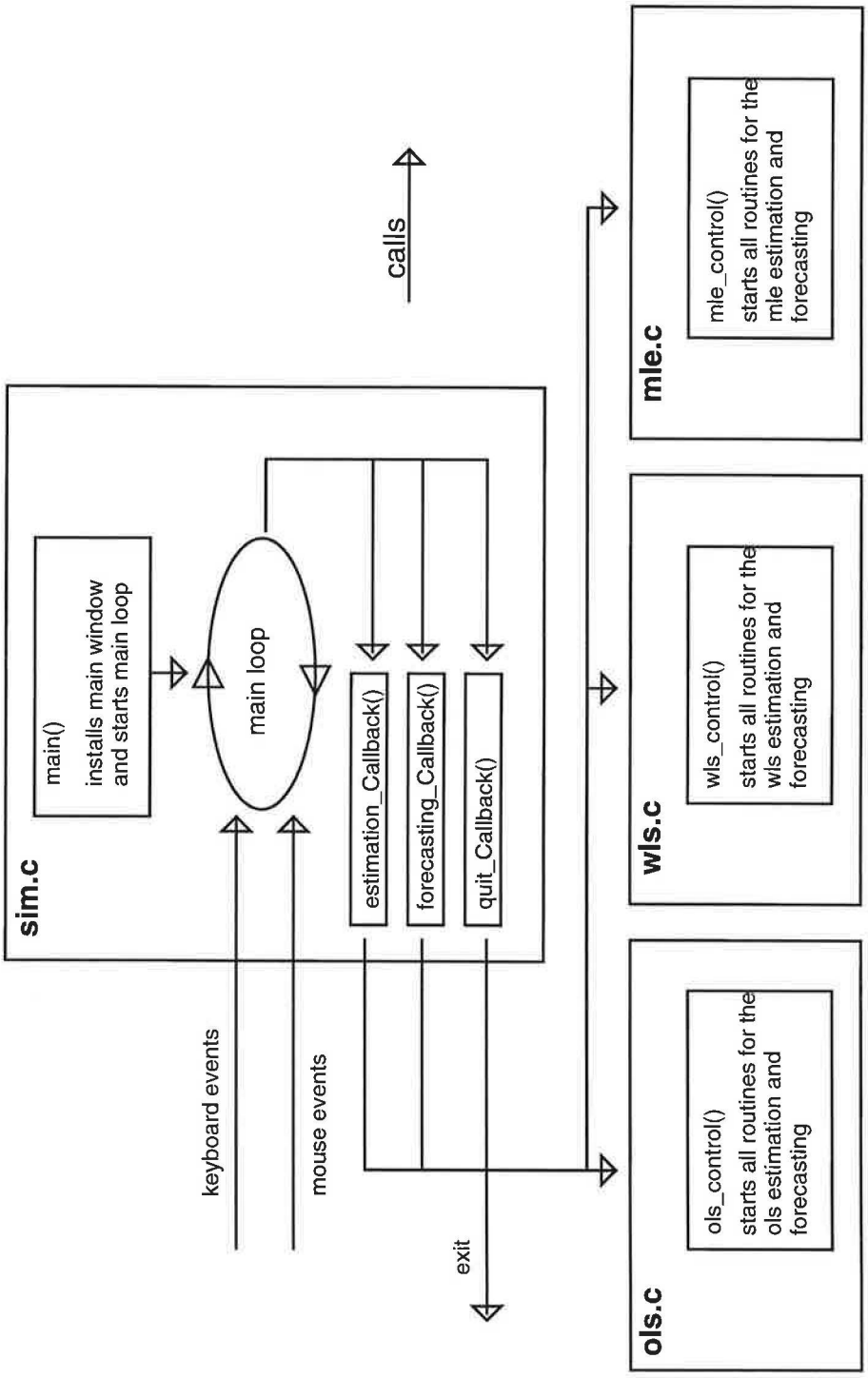

Figure 6.1: A schematic overview of the implementation of SIM

Variables	Model Type: Unconstrained	Production Constrained	Attraction Constrained	Doubly Constrained
d_{ij}	X	X	X	X
O_i		X		X
V_i	X		X	
D_j			X	X
W_j	X	X		

Figure A.1: Input variables

The separation factor F_{ij} is assumed to be a function $F(d_{ij})$ of some measure d_{ij} of separation from i to j. The separation measure d_{ij} has usually been taken to be univariate (e.g. distance, travel time, transportation costs or generalized costs). We distinguish the following separation functions based on a univariate separation measure:

$$F_{ij} = F(d_{ij}) = d_{ij}^{\alpha} \qquad \text{power function,} \tag{2a}$$

$$= \exp(\beta d_{ij}) \qquad \text{exponential function,} \tag{2b}$$

$$= d_{ij}^{\alpha} \exp(\beta d_{ij}) \qquad \text{Tanner function,} \tag{2c}$$

$$= d_{ij}^{\alpha} \exp(\beta d_{ij}^{\gamma}) \qquad \text{generalized Tanner function.} \tag{2d}$$

Four model types are available in SIM:

- the **doubly constrained model** (see appendix A.2),
- the **production constrained model** (see appendix A.3),
- the **attraction constrained model** (see appendix A.4) and
- the **unconstrained model** (see appendix A.5).

Figure A.1 outlines the variables included in the various model types, where O_i and D_j denote the total outflows and inflows in the spatial interaction system, respectively.

A.2 Doubly Constrained Model

If information is available on outflow and inflow totals, then the doubly constrained model is defined by

$$\hat{T}_{ij} = A_i \, O_i \, B_j \, D_j \, F(d_{ij}) \tag{3}$$

with

$$A_i = (\sum_j B_j D_j F(d_{ij}))^{-1}, \tag{4}$$

$$B_j = (\sum_i A_i O_i F(d_{ij}))^{-1}, \tag{5}$$

where $\hat{T}_{ij}$ denotes the estimated size (volume) of a flow from zone i to zone j, $F(.)$ the separation function, and d_{ij} the separation measure from i to j. A_i is an origin-specific balancing factor which ensures that

$$\sum_j \hat{T}_{ij} = O_i, \tag{6}$$

where O_i represents the total outflow of origin zone i. B_j is a destination-specific balancing factor which guarantees that

$$\sum_i \hat{T}_{ij} = D_j, \tag{7}$$

where D_j represents the total inflow into destination zone j.

Application domain: Trip distribution problems (e.g. forecasting traffic problems, trade patterns).

A.3 Production Constrained Model

In this case it is assumed that information is available on the outflow totals for each origin i. The model is defined by

$$\hat{T}_{ij} = A_i \, O_i \, W_j \, F(d_{ij}) \tag{8}$$

with

$$A_i = (\sum_j W_j F(d_{ij}))^{-1}, \tag{9}$$

where $\hat{T}_{ij}$ denotes the estimated size (volume) of a flow from zone i to zone j, $F(.)$ the separation function, d_{ij} the separation measure from i to j, and A_i the origin-specific balancing factor which ensures (6). O_i denotes the size of outflows from i to any j (outflow total of i), and W_j the attraction of destination zone j (destination factor).

Application domain: Forecasting destination inflow totals (e.g. modelling shopping expenditures to forecast the revenues generated by particular shopping locations to determine the optimal size of a shopping development, facility location).

A.4 Attraction Constrained Model

In this case it is assumed that information is available on the inflow totals for each destination j. The model is defined by

$$\hat{T}_{ij} = V_i \, B_j \, D_j \, F(d_{ij}) \tag{10}$$

with

$$B_j = (\sum_i V_i F(d_{ij}))^{-1}, \tag{11}$$

where $\hat{T}_{ij}$ denotes the estimated size (volume) of a flow from zone i to zone j, $F(.)$ the separation function, d_{ij} the separation measure from i to j, and B_j the destination-specific balancing factor which enures (7). D_j denotes the size of inflows to j from any i (inflow total of j), and V_i is a measure of the propulsiveness of origin zone i (origin factor).

Application domain: Forecasting total outflows from origins (e.g. forecasting the effects of locating a new industrial park within a city, forecasting university enrollment patterns).

A.5 Unconstrained Model

The model is unconstrained in terms of the production of flows from origins and the attraction of flows to destinations. It is assumed that information is available only on the total number of interactions in the system (i.e. $T_{\bullet\bullet} = \sum_{i,j} T_{ij}$). Then the model is defined by

$$\hat{T}_{ij} = K\, V_i^{\mu}\, W_j^{\nu}\, F(d_{ij}), \tag{12}$$

where $\hat{T}_{ij}$ denotes the estimated size (volume) of a flow from zone i to zone j, $F(.)$ the separation function, d_{ij} the separation measure from i to j, V_i the propulsiveness of origin zone i (origin factor), W_j the attraction of destination zone j (destination factor), μ the parameter reflecting the relationship between T_{ij} and V_i, ν the parameter reflecting the relationship between T_{ij} and W_j, and K the scale parameter.

B Estimation Techniques

SIM provides three different approaches to estimate the parameters of the models described above:

- **ordinary least squares (OLS)** and
- **weighted least squares (WLS)** regression with the odds ratio procedure and
- **maximum likelihood estimation (MLE)** by simulated annealing combined with the downhill simplex method.

B.1 Least Squares Estimation

In order to be calibrated by regression, a model must be in a linear format, that is, linear in terms of its parameters. The transformation is described in the sequal for each of the spatial interaction models.

B.1.1 Unconstrained Model

The transformation of (12) with any separation function into a linear format is easily achieved by simply taking logarithms of both sides of the equation. For illustration purposes we assume the choice of the power function (2a), then (12) becomes

$$\ln \hat{T}_{ij} = \ln K + \mu \ln V_i + \nu \ln W_j + \alpha \ln d_{ij}, \tag{13}$$

where ln denotes the natural logarithm. Note that the choice of another separation function would not alter the basic transformations which are the focus of this discussion (Fotheringham and O'Kelly, 1989).

The method by which the parameters $\ln K$, μ, ν and α are estimated in (13) is the least-squares technique. This technique provides estimates of the parameters such that the sum of all squared residuals, S, is minimized:

$$\begin{aligned} S &= \sum_{i,j} (\ln T_{ij} - \ln \hat{T}_{ij})^2 \\ &= \sum_{i,j} (\ln T_{ij} - \alpha_0 - \alpha_1 \ln V_i - \alpha_2 \ln W_j - \alpha_3 \ln d_{ij})^2, \end{aligned} \tag{14}$$

where $\alpha_0 := \ln K$, $\alpha_1 := \mu$, $\alpha_2 := \nu$ and $\alpha_3 := \alpha$.

Minimizing a function is a common problem in calculus and can be achieved by finding the partial derivatives of S with respect to the unknowns

$$\frac{\partial S}{\partial \alpha_0} = 2 \sum_{i,j} (\ln T_{ij} - \alpha_0 - \alpha_1 \ln V_i - \alpha_2 \ln W_j - \alpha_3 \ln d_{ij})(-1), \tag{15a}$$

$$\frac{\partial S}{\partial \alpha_1} = 2 \sum_{i,j} (\ln T_{ij} - \alpha_0 - \alpha_1 \ln V_i - \alpha_2 \ln W_j - \alpha_3 \ln d_{ij})(-\ln V_i), \tag{15b}$$

$$\frac{\partial S}{\partial \alpha_2} = 2 \sum_{i,j} (\ln T_{ij} - \alpha_0 - \alpha_1 \ln V_i - \alpha_2 \ln W_j - \alpha_3 \ln d_{ij})(-\ln W_j), \tag{15c}$$

$$\frac{\partial S}{\partial \alpha_3} = 2 \sum_{i,j} (\ln T_{ij} - \alpha_0 - \alpha_1 \ln V_i - \alpha_2 \ln W_j - \alpha_3 \ln d_{ij})(-\ln d_{ij}), \tag{15d}$$

and setting these equal to zero.

Dividing by 2 and bringing the negative terms to the right hand side of

the equation yields the normal equations in matrix form:

$$\begin{pmatrix} IJ & J\sum_i \ln V_i & I\sum_j \ln W_j & \sum_{i,j} \ln d_{ij} \\ J\sum_i \ln V_i & J\sum_i (\ln V_i)^2 & \sum_{i,j} \ln V_i \ln W_j & \sum_{i,j} \ln V_i \ln d_{ij} \\ I\sum_j \ln W_j & \sum_{i,j} \ln W_j \ln V_i & I\sum_j (\ln W_j)^2 & \sum_{i,j} \ln W_j \ln d_{ij} \\ \sum_{i,j} \ln d_{ij} & \sum_{i,j} \ln d_{ij} \ln V_i & \sum_{i,j} \ln d_{ij} \ln W_j & \sum_{i,j} (\ln d_{ij})^2 \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix}$$

$$= \begin{pmatrix} \sum_{i,j} \ln T_{ij} \\ \sum_{i,j} \ln T_{ij} \ln V_i \\ \sum_{i,j} \ln T_{ij} \ln W_j \\ \sum_{i,j} \ln T_{ij} \ln d_{ij} \end{pmatrix}. \tag{16}$$

The way to solve the linear set of equations

$$AX = B \tag{17}$$

is to use a decomposition of $A = LU$ where L is lower triangular (has elements only on the diagonal and below) and U is upper triangular (has elements only on the diagonal and above):

$$AX = (LU)X = L(UX) = B \tag{18}$$

and first solve for the vector Y such that

$$LY = B \tag{19}$$

and then

$$UX = Y. \tag{20}$$

The advantage of breaking up one linear set into two successive sets is that the solution of a triangular set of equations is quite trivial. (19) can be solved by forward substitution and (20) by backward substitution (Press et al., 1992).

Note that the WLS procedure implemented takes the underestimation of the constant term, K, in the following manner into consideration:

$$\hat{K}_{\text{new}} = \hat{K}_{\text{old}} \frac{\sum_{i,j} T_{ij}}{\sum_{i,j} \hat{T}_{ij}}. \tag{21}$$

B.1.2 Constrained Models

At a first glance the constrained models look to be intrinsically nonlinear in their parameters (i.e. nonlinear equations which can not be linearized by transformation). The equations of the constrained models are complicated by the balancing factors which involve summations of the models

parameters. Sen and Sööt (1981) have described a technique to achieve the linearization of the constrained models. The technique is termed the odds ratio technique and involves taking ratios of interactions so that the A_iO_i and/or the B_jD_j terms in the models cancel out.

In the sequal $k = 1, \ldots, K$ denotes the index for the k'th component of a vector. Let us consider a rather general separation function

$$F_{ij} = F(\vec{d}_{ij}) = \exp(\sum_k \Theta_k d_{ij}^{(k)}) \tag{22}$$

with a vector-valued separation measure

$$\vec{d}_{ij} = (d_{ij}^{(1)}, \ldots, d_{ij}^{(K)}), \tag{23}$$

where components could be variables like distances, travel time, costs and other measures of separation. $\vec{\Theta} = (\Theta_1, \ldots, \Theta_K)$ is the separation function parameter vector. Equation (22) evidently includes the power function (2a) as a special case for $\Theta_1 = \alpha$ and $d_{ij}^{(1)} = \ln d_{ij}$,

$$F_{ij} = F(\vec{d}_{ij}) = \exp(\alpha \ln d_{ij}) = d_{ij}^{\alpha}, \tag{24}$$

and the exponential function (2b) as a special case for $\Theta_1 = \beta$ and $d_{ij}^{(1)} = d_{ij}$,

$$F_{ij} = F(\vec{d}_{ij}) = \exp(\beta d_{ij}), \tag{25}$$

and the Tanner function (2c) as a special case for $\Theta_1 = \alpha$, $\Theta_2 = \beta$, $d_{ij}^{(1)} = \ln d_{ij}$ and $d_{ij}^{(2)} = d_{ij}$,

$$F_{ij} = F(\vec{d}_{ij}) = \exp(\alpha \ln d_{ij} + \beta d_{ij}) = d_{ij}^{\alpha} \exp(\beta d_{ij}). \tag{26}$$

Consider the production and attraction constrained model (8)-(11) with the general separation function (22). The transformation of this general structure into a form linear in parameters may be obtained as follows. Multiply together the set of flows emanating from each origin to the J destinations, take the Jth root of both sides of the equation, divide both sides of the equation into $\hat{T}_{ij}$ and substitute for $\hat{T}_{ij}$ in the right-hand side and then take logarithms of both sides and rearrage, then

$$\begin{aligned} \hat{t}_{ij} + \hat{t}_{\bullet\bullet} - \hat{t}_{i\bullet} - \hat{t}_{\bullet j} &= f_{ij} + f_{\bullet\bullet} - f_{i\bullet} - f_{\bullet j} \\ &= \sum_k \Theta_k (d_{ij}^{(k)} + d_{\bullet\bullet}^{(k)} - d_{i\bullet}^{(k)} - d_{\bullet j}^{(k)}), \end{aligned} \tag{27}$$

where $\hat{t}_{ij} = \ln \hat{T}_{ij}$, $\hat{t}_{\bullet\bullet} = I^{-1}J^{-1} \sum_{i,j} \ln \hat{T}_{ij}$, $\hat{t}_{i\bullet} = J^{-1} \sum_j \ln \hat{T}_{ij}$, $\hat{t}_{\bullet j} = I^{-1} \sum_i \ln \hat{T}_{ij}$, $f_{ij} = \ln F_{ij}$, $f_{\bullet\bullet} = I^{-1}J^{-1} \sum_{i,j} \ln F_{ij}$, $f_{i\bullet} = J^{-1} \sum_j \ln F_{ij}$,

$f_{\bullet j} = I^{-1}\sum_i \ln F_{ij}$, $d^{(k)}_{\bullet\bullet} = I^{-1}J^{-1}\sum_{i,j} d^{(k)}_{ij}$, $d^{(k)}_{i\bullet} = J^{-1}\sum_j d^{(k)}_{ij}$ and $d^{(k)}_{\bullet j} = I^{-1}\sum_i d^{(k)}_{ij}$. Defining

$$y_{ij} := \hat{t}_{ij} + \hat{t}_{\bullet\bullet} - \hat{t}_{i\bullet} - \hat{t}_{\bullet j} = f_{ij} + f_{\bullet\bullet} - f_{i\bullet} - f_{\bullet j} \tag{28}$$

and

$$X^{(k)}_{ij} := d^{(k)}_{ij} + d^{(k)}_{\bullet\bullet} - d^{(k)}_{i\bullet} - d^{(k)}_{\bullet j} \tag{29}$$

then (27) might be written as

$$y_{ij} = \sum_k \Theta_k X^{(k)}_{ij}. \tag{30}$$

In the case of the **production constrained model** we get

$$y_{ij} = \hat{t}_{ij} - \hat{t}_{i\bullet} - (\ln W_j - J^{-1}\sum_j \ln W_j) \tag{31}$$

and

$$X^{(k)}_{ij} := d^{(k)}_{ij} - d^{(k)}_{i\bullet} \tag{32}$$

and in the case of the **attraction constrained model**

$$y_{ij} = \hat{t}_{ij} - \hat{t}_{\bullet j} - (\ln V_i - I^{-1}\sum_i \ln V_i) \tag{33}$$

and

$$X^{(k)}_{ij} := d^{(k)}_{ij} - d^{(k)}_{\bullet j}. \tag{34}$$

Note that (28)-(30) represents the doubly constrained model.

Now let us examine how we may obtain estimates for Θ_k in the production constrained, attraction constrained and doubly constrained cases.

Let denote

$$\begin{aligned} S &= \sum_{i,j}(t_{ij} - \hat{t}_{ij})^2 \\ &= \sum_{i,j}[t_{ij} - (\hat{t}_{i\bullet} + \hat{t}_{\bullet j} - \hat{t}_{\bullet\bullet} + \sum_k \Theta_k(d^{(k)}_{ij} + d^{(k)}_{\bullet\bullet} - d^{(k)}_{i\bullet} - d^{(k)}_{\bullet j}))]^2, \end{aligned} \tag{35}$$

where $t_{ij} = \ln T_{ij}$. Minimizing (35) with respect to Θ_k, we obtain the

following normal equations in matrix form:

$$\begin{pmatrix} \sum_{i,j} X_{ij}^{(1)} X_{ij}^{(1)} & \sum_{i,j} X_{ij}^{(1)} X_{ij}^{(2)} & \cdots & \sum_{i,j} X_{ij}^{(1)} X_{ij}^{(K)} \\ \sum_{i,j} X_{ij}^{(2)} X_{ij}^{(1)} & \sum_{i,j} X_{ij}^{(2)} X_{ij}^{(2)} & \cdots & \sum_{i,j} X_{ij}^{(2)} X_{ij}^{(K)} \\ \vdots & \vdots & & \vdots \\ \sum_{i,j} X_{ij}^{(K)} X_{ij}^{(1)} & \sum_{i,j} X_{ij}^{(K)} X_{ij}^{(2)} & \cdots & \sum_{i,j} X_{ij}^{(K)} X_{ij}^{(K)} \end{pmatrix} \begin{pmatrix} \Theta_1 \\ \Theta_2 \\ \vdots \\ \Theta_K \end{pmatrix}$$

$$= \begin{pmatrix} \sum_{i,j} y_{ij} X_{ij}^{(1)} \\ \sum_{i,j} y_{ij} X_{ij}^{(2)} \\ \vdots \\ \sum_{i,j} y_{ij} X_{ij}^{(K)} \end{pmatrix}. \tag{36}$$

This linear set of equations is solved in the same way as (16), i.e. by decomposing the coefficient matrix, breaking up the linear set into two successive sets, and using forward and backward substitution.

In the **univariate case** (i.e. $K = 1$), for example, we obtain

$$\Theta_1 = \frac{\sum_{i,j} y_{ij} X_{ij}^{(1)}}{\sum_{i,j} X_{ij}^{(1)} X_{ij}^{(1)}}, \tag{37}$$

and in the **bivariate case** (i.e. $K = 2$)

$$\begin{aligned} \Theta_1 &= \frac{\sum_{i,j} y_{ij} X_{ij}^{(1)} \sum_{i,j} X_{ij}^{(2)} X_{ij}^{(2)} - \sum_{i,j} y_{ij} X_{ij}^{(2)} \sum_{i,j} X_{ij}^{(2)} X_{ij}^{(1)}}{\sum_{i,j} X_{ij}^{(1)} X_{ij}^{(1)} \sum_{i,j} X_{ij}^{(2)} X_{ij}^{(2)} - \sum_{i,j} X_{ij}^{(1)} X_{ij}^{(2)} \sum_{i,j} X_{ij}^{(2)} X_{ij}^{(1)}}, \\ \Theta_2 &= \frac{\sum_{i,j} y_{ij} X_{ij}^{(2)} \sum_{i,j} X_{ij}^{(1)} X_{ij}^{(1)} - \sum_{i,j} y_{ij} X_{ij}^{(1)} \sum_{i,j} X_{ij}^{(1)} X_{ij}^{(2)}}{\sum_{i,j} X_{ij}^{(1)} X_{ij}^{(1)} \sum_{i,j} X_{ij}^{(2)} X_{ij}^{(2)} - \sum_{i,j} X_{ij}^{(1)} X_{ij}^{(2)} \sum_{i,j} X_{ij}^{(2)} X_{ij}^{(1)}}. \end{aligned} \tag{38}$$

Once $\vec{\Theta} = (\Theta_1, \ldots, \Theta_K)$ is estimated, the balancing factors A_i and B_j can be obtained by iterating (4) and (5) in the case of the doubly constrained model, from (9) in the case of the production constrained model and from (11) in the case of the attraction constrained model.

Weighted least squares may be preferable to ordinary least squares to counteract the heteroscedastic error terms caused by the logarithmic transformation with the weight being $(T_{ij}^{-1} + T_{ji}^{-1} + T_{ii}^{-1} + T_{jj}^{-1})^{-0.5}$ and the assumption that the T_{ij}'s have a Poisson distribution (see Sen and Pruthi, 1983).

B.2 Maximum Likelihood Estimation

In essence, the MLE technique is to find parameter estimates that maximise the likelihood of observing a sample set of interactions from a theoretical

distribution. The steps involved in the calibration include identifying a theoretical distribution for the interactions, maximising the likelihood function of this distribution with respect to the parameters of the interaction model and then deriving equations which ensure the maximisation of the likelihood function. For convenience, the logarithm of the likelihood function is usually used since this is at a maximum whenever the likelihood function is at a maximum. Parameter estimates that maximise the likelihood function are termed maximum likelihood estimates. Maximum likelihood estimates have several desirable properties: they are consistent, asymptotically efficient and asymptotically normally distributed. This latter property is particulary useful in significance testing. The method of obtaining ML-estimates is described for each of the four model types (see Fotheringham and O'Kelly, 1989).

B.2.1 The Principle of Maximum Likelihood

To introduce the principle of maximum likelihood as a tool for finding parameter estimates, it is necessary to convert the models to a probabilistic form.

The Unconstrained Model

Let T_{ij} denote a flow from i to j. T_{ij} might be considered to be the outcome of a Poisson process if it is assumed that there is a constant probability of any individual in i moving to j, that the population of i is large, and that the number of individuals interacting is an independent process (see Flowerdew and Aitkin, 1982). Then the probability that T_{ij} is the number of people recorded as moving from i to j is given by

$$p(T_{ij}) = \frac{\hat{T}_{ij}^{T_{ij}} \exp(-\hat{T}_{ij})}{T_{ij}!}, \tag{39}$$

where $\hat{T}_{ij}$ is the expected outcome of the Poisson process and T_{ij} the observed value which is subject to sampling and measuring errors and thus fluctuates around the expected value. Since $\hat{T}_{ij}$ is unknown and unobservable, it has to be estimated from some theoretical model such as the unconstrained model (12).

Consider the log-likelihood of a set of observed flows T_{ij}, where each flow is the outcome of a particular Poisson process. This log-likelihood, $\log L$, can be represented as

$$\begin{aligned} \log L &= \sum_{i,j} \ln\left(\frac{\hat{T}_{ij}^{T_{ij}} \exp(-\hat{T}_{ij})}{T_{ij}!}\right) \\ &= \sum_{i,j} (-\hat{T}_{ij} + T_{ij} \ln \hat{T}_{ij} - \ln(T_{ij}!)). \end{aligned} \tag{40}$$

Since T_{ij} is given, $\ln(T_{ij}!)$ can be ignored in the maximisation, and $\log L$ will be at a maximum when

$$Z = \sum_{i,j} (T_{ij} \ln \hat{T}_{ij} - \hat{T}_{ij}) \tag{41}$$

is at a maximum. Thus, the parameter estimates associated with $\hat{T}_{ij}$, which maximize Z, are received.

The Constrained Models

There exists several possibilities of defining a likelihood function for the constrained models. In SIM equation (41) is used for all model types.

Batty and Mackie (1972) assumed interactions having a multinomial distribution, in which case the log-likelihood of observing a set of flows is

$$\log L = \sum_{i,j} T_{ij} \ln p_{ij}, \tag{42}$$

where p_{ij} represents the predicted probability of moving between i and j, and is defined as

$$p_{ij} = \frac{\hat{T}_{ij}}{\sum_{i,j} \hat{T}_{ij}}. \tag{43}$$

Alternatively, p_{ij} can be defined as the product of two other probabilities

$$p_{ij} = p_{j|i}\, p_i, \tag{44}$$

where $p_{j|i}$ is the conditional probability of interacting with j given one originates at i, and p_i is the probability of an interaction originating in i. In a production constrained model,

$$\sum_j p_{j|i} = 1, \tag{45}$$

and, clearly,

$$\sum_i p_i = 1. \tag{46}$$

The probability $p_{j|i}$ is given by the production constrained model

$$p_{j|i} = A_i O_i W_j F(d_{ij}), \tag{47}$$

so that the objective is to determine the parameters of the separation function $F(d_{ij})$ which maximize (42) subject to the constraints in (45) and (46).

The method is identical for the production constrained (described here) and attraction constrained model. The only difference in the calibration of the doubly constrained model from that of the production constrained model is that an extra set of parameters, the B_j's, is estimated from the destination constraint set (see equation 7).

B.2.2 The Algorithm for Maximum Likelihood Calibration

In SIM the **simulated annealing algorithm** (originally developed by Kirkpatrick et al., 1983) is used in combination with a modification of the standard **downhill simplex method** (due to Nelder and Mead, 1965) to maximize Z, respectively minimize $-Z$ (see equation 41), i.e. to find the ML-estimates of the model (for the combination see Press et al., 1992).

The **simulated annealing algorithm** has attracted significant attention as suitable for optimization problems of large scale, especially ones where a desired global extremum is hidden among many, poorer, local extrema. In contrast to conventional iterative optimization algorithms, simulated annealing shows the attractive feature not to get stuck in a local minima, since transition out of a local minima is always possible when the system operates at a non-zero temperature.

The simulated annealing algorithm is based on the analogy between the behavior of a physical system with many degrees of freedom in thermal equilibrium at a series of finite temperature as encountered in statistical physics and the problem of finding the minimum of a given function depending on many parameters as in combinatorial optimization (Kirkpatrick et al., 1983). In condensed-matter physics, annealing refers to a physical process that proceeds as follows (Laarhoven and Aarts, 1988). First, a solid in a heath bath is heated by raising the temperature to a maximum value at which all particles of the solid arrange randomly in the liquid phase. Second, then the temperature of the heath bath is lowered, allowing all particles to arrange themselves in the low-energy ground state of a corresponding lattice. This assumes that the maximum temperature in phase one is sufficiently high, and the cooling in phase two is carried out sufficiently slowly. If the cooling is too rapid, the crystal will have many defects (Kirkpatrick et al., 1983). The simulated annealing algorithm developed by Kirkpatrick et al. (1983) is a variant (with time dependent temperature) of the Metropolis algorithm proposed by Metropolis et al. (1953) for efficient simulation of the evolution to thermal equilibrium of a solid for a given temperature, based on Monte Carlo techniques.

In practice, one has to resort to a finite-time approximation of the asymptotic convergence of the simulated annealing algorithm. To implement such an approximation one needs to specify a set of parameters governing the convergence of the algorithm. These parameters are combined in a so-called annealing schedule. The search for adequate annealing schedules has been the subject of an active research area for several years (see Laarhoven and Aarts, 1988).

The annealing schedule that we adopt is based on a number of conceptually simple empirical rules. The **annealing schedule** specifies the parameters of interest as follows:

1. **Initial value of temperature T_0:**

The initial value T_0 of the temperature is chosen high enough to ensure that virtually all proposed transitions are accepted by the simulated annealing algorithm. There exists no exact rule for determining T_0, but for SIM one can use the following rule of thumb,

$$T_0 = 20\ \Delta \log L, \tag{48}$$

where $\Delta \log L$ denotes the largest difference between any of the log-likelihood function values evaluated at the initial parameter values (default: $T_0 = 1000$).

2. **Decrement of the temperature ΔT:**
   Ordinarily, the cooling is performed experimentally, with the changes made in the value of the temperature being small. Especially, the decrement function is defined by

$$T_l = T_{l-1} - \Delta T, \qquad l = 1, \ldots, \lfloor T_0/\Delta T \rfloor \tag{49}$$

   where $\lfloor T_0/\Delta T \rfloor$ denotes the largest cardinal number less than or equal $T_0/\Delta T$. Typical values of ΔT lie between 5 and 40 (default: $\Delta T = 10$).

3. **Number of iterations at each temperature τ:**
   At each temperature, τ transitions are computed. τ depends as all annealing schedule parameters on the thermodynamical properties of the likelihood function and must be large enough to allow the system to reach the thermal equilibrium at a given temperature. Typical values of τ lie between 5 and 30 (default: $\tau = 20$).

It is worthwile to note that, as in all applications of simulated annealing, success or failure is quite often determined by the choice of the annealing schedule. There exists also a tradeoff between computing time (large values of T_0 and τ and small values of ΔT) and the accuracy of the final results (small values of T_0 and τ and large values of ΔT).

In SIM the simulated annealing algorithm is used in combination with a modification of the **downhill simplex method** to improve the performance of the algorithm in situations where local downhill moves exists (as suggested by Press et al., 1992).

The standard downhill simplex method due to Nelder and Mead (1965) is a multidimensional optimization algorithm that requires only function evaluations, not derivatives, and amounts to replace the single point $\vec{\Theta} = (\Theta_1, \ldots, \Theta_K)$, representing the parameter vector, by a simplex of $K + 1$ points $\vec{\Theta}_l = (\Theta_{1l}, \ldots, \Theta_{Kl})$ $(l = 1, \ldots, K + 1)$. Starting with an initial simplex, the algorithm takes a series of steps, most steps just moving the point of the simplex where the function is largest (highest point) through the opposite face of the simplex to a lower point. These steps are called reflections, and they are constructed to conserve the volume of the simplex.

When it can do so, the method expands the simplex in one or another direction to take larger steps. When it reaches a "valley floor", the method contracts itself in the transverse direction and tries to ooze down the valley. If there is a situation where the simplex is trying to "pass through the eye of a needle", it contracts itself in all directions, pulling itself in around its lowest (best) point. The algorithm terminates when e.g. the decrease in the function value becomes smaller than some tolerance.

In SIM the algorithm developed by Press et al. (1992) is used. In contrast to the standard downhill simplex method a positive, logarithmically distributed random variable proportional to the temperature is added to the stored function value associated with every vertex of the simplex, and a similar random variable is subtracted from the function value of every new point that is tried as a replacement point. This method always accepts a true downhill step, but sometimes also accepts an uphill one. In the limit as the temperature goes to zero, this algorithm reduces exactly to the standard downhill simplex method and converges to a local minimum. At a finite temperature, the simplex expands to a scale that approximates the size of the region that can be reached at this temperature, and then executes a stochastic, tumbling Brownian motion within that region, sampling new, approximately random, points as it does so. If the temperature is reduced sufficiently slowly, it becomes highly likely that the simplex will shrink into that region containing the lowest relative minimum encountered.

C Performance Statistics

An important element of SIM is the assessment of the model's ability to replicate a known set of flows. Accurate replication supports the theoretical propositions on which the model is based, i.e. it supports one particular model form over others. However the results and their interpretation depend to a certain degree on the way in which the separation between the zones is defined (see e.g. Fischer et al., 1992).

Many performance statistics have been employed in spatial interaction modelling (see e.g. Fotheringham and O'Kelly, 1989). All such statistics involve a quantitative description of some aspect of the difference between the matrices of predicted and observed flows. It is important to emphasize that the use of different performance statistics might lead to different conclusions concerning the goodness of fit of a model under consideration.

SIM provides an output file consisting of the following components:
Only for the **constrained model types**:

- Balancing factors A_i and B_j (see equations 4, 5, 9 and 11).

For **all model types**:

- Observed flow T_{ij},

- Estimated flow $\hat{T}_{ij}$,
- Total observed interactions $\sum_{i,j} T_{ij}$,
- Total estimated interactions $\sum_{i,j} \hat{T}_{ij}$,
- Absolute error:

$$E_{\text{abs}} = T_{ij} - \hat{T}_{ij}, \tag{50}$$

- Relative error:

$$E_{\text{rel}} = \frac{T_{ij} - \hat{T}_{ij}}{T_{ij}}, \tag{51}$$

- Percentage deviation of observed interactions from the mean:

$$D_{\text{obs-mean}} = \frac{\sum_{i,j} \mid T_{ij} - \bar{T} \mid}{\sum_{i,j} T_{ij}}, \tag{52}$$

  where $\bar{T} = (IJ)^{-1} \sum_{i,j} T_{ij}$,

- Percentage deviation of estimated from observed interactions:

$$D_{\text{est-obs}} = \frac{\sum_{i,j} \mid T_{ij} - \hat{T}_{ij} \mid}{\sum_{i,j} T_{ij}}, \tag{53}$$

- Regressing the observed on the predicted interactions (see e.g. Neter et al., 1985):

$$\begin{aligned} T_{ij} &= a + b\hat{T}_{ij} + \varepsilon_{ij}, \\ \tilde{T}_{ij} &= a + b\hat{T}_{ij}, \\ \text{t-value}_a &= \frac{a}{\sqrt{\text{avar}(a)}}, \\ \text{t-value}_b &= \frac{b-1}{\sqrt{\text{avar}(b)}}, \\ R^2(2) &= \frac{\sum_{i,j}(\tilde{T}_{ij} - \bar{T})^2}{\sum_{i,j}(T_{ij} - \bar{T})^2}, \end{aligned} \tag{54}$$

  where $\text{avar}(a)$ and $\text{avar}(b)$ denote estimators of the asymptotic variance of a and b, respectively,

- Estimated model parameters (see appendix A):

  All model types: Distance parameter: power function: α; exponential function: β; power-exponential function: γ,

Unconstrained model: Constant $\ln K$, origin parameter μ, destination parameter ν,

- RMSE (Root Mean Square Error) statistic:

$$\text{RMSE} = \sqrt{\sum_{i,j}(IJ)^{-1}(T_{ij} - \hat{T}_{ij})^2}, \tag{55}$$

- SRMSE (Standardized Root Mean Square Error) statistic:

$$\text{SRMSE} = \frac{\text{RMSE}}{\overline{\hat{T}}}100, \tag{56}$$

where $\overline{\hat{T}} = (IJ)^{-1}\sum_{i,j}\hat{T}_{ij}$,

- ARV (Average Relative Variance) statistic:

$$\text{ARV} = \frac{\sum_{i,j}(T_{ij} - \hat{T}_{ij})^2}{\sum_{i,j}(T_{ij} - \overline{T})^2}, \tag{57}$$

where $\bar{T} = (IJ)^{-1}\sum_{i,j}T_{ij}$,

- $R^2(1)$ statistic:

$$R^2(1) = 1 - \text{ARV}, \tag{58}$$

- $R^2(1)_{\text{adjusted}}$ statistic:

$$R^2(1)_{\text{adjusted}} = R^2(1) - \frac{K-1}{IJ-K}(1 - R^2(1)), \tag{59}$$

where K denotes the number of parameters,

- $R^2(2)$ statistic:

$$R^2(2) = \frac{\sum_{i,j}(\hat{T}_{ij} - \overline{T})^2}{\sum_{i,j}(T_{ij} - \overline{T})^2}, \tag{60}$$

- $R^2(2)_{\text{adjusted}}$ statistic:

$$R^2(2)_{\text{adjusted}} = R^2(2) - \frac{K-1}{IJ-K}(1 - R^2(2)), \tag{61}$$

- FW statistic:

$$\text{FW} = \left|1 - R^2(2)\right|, \tag{62}$$

- $\mathrm{FW}_{\mathrm{adjusted}}$ statistic:

$$\mathrm{FW}_{\mathrm{adjusted}} = \mathrm{FW} - \frac{K-1}{IJ-K}(1-\mathrm{FW}), \tag{63}$$

- Information gain statistic:

$$I_{\mathrm{gain}} = \sum_{i,j} T_{ij} \ln\left(\frac{T_{ij}}{\hat{T}_{ij}}\right), \tag{64}$$

- MDI (Minimum Discrimination Information) statistic:

$$\mathrm{MDI} = \frac{I_{\mathrm{gain}}}{\sum_{i,j} T_{ij}}. \tag{65}$$

Only for **OLS** and **WLS** and the **unconstrained model**:

- Log versions of the above performance measures: T_{ij} and $\hat{T}_{ij}$ replaced by $\ln T_{ij}$ and $\ln \hat{T}_{ij}$, respectively.

Only for **MLE**:

- Log-likelihood function values $\log L(\vec{\hat{\Theta}})$ and $\log L(\vec{\hat{\Theta}}_i)$ with all estimated parameters $\vec{\hat{\Theta}} = (\hat{\Theta}_1, \ldots, \hat{\Theta}_K)$ and with one parameter set equal to zero $\vec{\hat{\Theta}}_i = (\hat{\Theta}_1, \ldots, \hat{\Theta}_{i-1}, 0, \hat{\Theta}_{i+1}, \ldots, \hat{\Theta}_K)$, respectively,

- Relative likelihood statistics:

$$\tau_i = 2\,(\sum_{i,j} T_{ij})\,(\log L(\vec{\hat{\Theta}}) - \log L(\vec{\hat{\Theta}}_i)), \tag{66}$$

  where τ_i is asymptotically χ^2 with one degree of freedom under the null hypothesis H_0: $\hat{\Theta}_i = 0$.

References

Batty, M. and Mackie, S. (1972): The calibration of gravity, entropy, and related models of spatial interaction, **Environment and Planning A** 4, pp. 205-233.

Fischer, M.M., Essletzbichler, J., Gassler, H. and Trichtl, G. (1992): Interregional and international telephone communications: aggregate traffic models and empirical evidence for Austria, **Sistemi Urbani** 15, pp. 121-135.

Flowerdew, R. and Aitkin, M. (1982): A method of fitting the gravity model based on the Poisson distribution, **Journal of Regional Science** 22, pp. 191-202.

Fotheringham, A.S. and O'Kelly, M.E. (1989): **Spatial Interaction Models: Formulations and Applications**. Dordrecht, Boston and London: Kluwer Academic Publishers.

Kirkpatrick, S., Gelatt, C.D. Jr and Vecchi, M.P. (1983): Optimization by simulated annealing, **Science** 220, pp. 671-680.

Laarhoven, P.J.M. Van and Aarts, E.H.L. (1988): **Simulated Annealing: Theory and Applications**. Boston (Ma.): Kluwer Academic Publishers.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A. and Teller, E. (1953): Equation of state calculations by fast computing machines, **Journal of Chemical Physics** 21, pp. 1087-1092.

Nelder, J.A. and Mead, R. (1965): A simplex method for function minimisation, **The Computer Journal** 7, pp. 308-313.

Neter, J., Wasserman, W. and Kutner M.H. (1985): **Applied Linear Statistical Models. Regression, Analysis of Variance, and Experimental Designs**. Homewood, Illinois: Irwin.

Press, W.H., Teukolsky, S.A., Vetterling, W.T. and Flannery, B.T. (1992): **Numerical Recipes in C. The Art of Scientific Computing. Second Edition**. Cambridge: Cambridge University Press.

Sen, A. and Pruthi, R.K. (1983): Least squares calibration of the gravity model when intrazonal flows are unknown, **Environment and Planning A** 15, pp. 1545-1550.

Sen, A. and Sööt, S. (1981): Selected procedures for calibrating the generalized gravity model, **Papers of the Regional Science Association** 48, pp. 165-176.