ESTIMATING MULTIATTRIBUTE SPATIAL CHOICE MODELS

Michael Wegener
Friedrich Graef

September 1982
WP-82-93

INTERNATIONAL INSTITUTE FOR APPLIED SYSTEMS ANALYSIS
A-2361 Laxenburg, Austria

ABOUT THE AUTHORS

Michael Wegener was with the Human Settlements and Services Area of IIASA from February to July 1982 on leave from the Institute of Urban and Regional Planning of the University of Dortmund, FRG.

Friedrich Graef is with the Institute of Applied Mathematics of the University of Erlangen-Nürnberg, FRG.

FOREWORD

The public provision of urban facilities and services often takes the form of a few central supply points serving a large number of spatially dispersed demand points: for example, hospitals, schools, libraries, and emergency services such as fire and police. A fundamental characteristic of such systems is the spatial separation between suppliers and consumers. No market signals exist to identify efficient and inefficient geographical arrangements, thus the location problem is one that arises in both East and West, in planned and in market economies.

This problem is being studied at IIASA by the Public Facility Location Task which started in 1979. The expected results of this Task are a comprehensive state-of-the-art survey of current theories and applications, an established network of international contacts among scholars and institutions in different countries, a framework for comparison, unification, and generalization of existing approaches as well as the formulation of new problems and approaches in the field of optimal location theory.

This paper sets out a general method for maximizing the likelihood function of spatial choice models in an effort to bring together the many separate methods for modeling spatial choices or interactions that have been presented in the past. Also included is a computer program written for the calibration of these various models.

A list of related publications appears at the end of this paper.

Andrei Rogers
Chairman
Human Settlements
and Services Area

ABSTRACT

In this paper, an interactive computer program for estimating the parameters of spatial choice models with multiattribute utilities is presented. The models to be calibrated may be unconstrained, singly constrained, or doubly constrained random utility choice or entropy-maximizing interaction models. Utilities may be associated with choice alternatives (zones) or with the choices themselves (trips). The program maximizes the likelihood of the choice matrix (trip table) given observed choices (trips) using a combination of gradient search and Newton-Raphson iteration methods.

The paper contains a specification of the range of models that can be calibrated with the program and a description of its solution algorithm and organization, as well as an illustrative application and a listing of the source code.

CONTENTS

ESTIMATING MULTIATTRIBUTE SPATIAL CHOICE MODELS

INTRODUCTION

Recent advances in spatial theory have led to a unification of formerly separate approaches to modeling spatial choices or interactions. It has been shown that spatial choice models built on stochastic utility maximization (McFadden 1973) and spatial interaction models of the entropy-maximizing type (Wilson 1970) are formally identical (Williams 1977) and have the same parameters and results when applied at the same aggregation level (Anas 1981). Moreover, it has been observed that by varying the exponent parameter of these models, the whole continuum from indifferent to strict utility maximizing behavior can be represented, actual human behavior being something in between (Brotchie et al. 1980).

The model doing so many things is the *multinomial logit* model with its many variations. Because of its simplicity, its attractive mathematical properties, and its multiple interpretability, it has become the most widespread, almost universal approach to modeling spatial choices in transportation or residential and industrial location analysis.

This paper deals with the problem of *calibrating* a spatial choice model of the multinomial logit or entropy-maximizing type, i.e., of estimating its parameters such that it reproduces a given set of observed choices (trips) as closely as possible. Unfortunately, this is not a trivial problem, because the model is intrinsically nonlinear, i.e., cannot normally be linearized by logarithmic transformation. This means that there is no straightforward analytical technique to find the best-fit parameters, but that a numerical approximation technique has to be applied.

Earlier work in this field includes, among others, that of Hyman (1969), Evans (1971), Batty and Mackie (1972), Batty (1976), van Est and van Setten (1977, 1978), Putman and Ducca (1978a), and Openshaw (1979). Hyman (1969) and Evans (1971) proposed different but equivalent algorithms to estimate one-parameter production-attraction-constrained trip distribution models based on Bayesian statistics and on the principle of maximum likelihood, respectively. Batty and Mackie (1972) and Batty (1976) explored various numerical methods to estimate singly and doubly constrained models with one, two, or three parameters. Van Est and van Setten (1977, 1978) investigated maximum-likelihood and least-square methods for singly constrained models with multiple parameters. Putman and Ducca (1978a) proposed a maximum-likelihood method for estimating a production-constrained interaction model where not the interactions themselves but only the destinations are known. An evaluation of various calibration techniques is contained in Openshaw (1979).

A summary result of this accumulated research is that there is no single "correct" way of calibrating spatial choice or interaction models, as the choice of a calibration method heavily depends on the available data, the purpose of the model, and the specific preferences of the research. However, maximum-likelihood estimation seems to be the most widely accepted method. This is true also for nonspatial random-utility choice models where maximum-likelihood estimation now is a standard method

(cf. van Lierop and Nijkamp 1981). Therefore, in the approach presented in this paper, the maximum-likelihood criterion is used.

Maximizing the likelihood function of a spatial choice or interaction model is conceptually straightforward, and most of the references given above contain the necessary equations. However, all of them are specific, i.e., are restricted to a certain type of model, to a certain kind of constraint, or to a limited number of parameters. In contrast, the method presented in this paper is general. The models to be calibrated may be unconstrained, production-constrained, attraction-constrained, or production-attraction-constrained. Moreover, they may be single-attribute or multiattribute in the exponent, i.e., in the utility term, and the utility attributes may be associated either with the choice alternatives (zones) or with the choices themselves (trips). So the method encompasses most of the specialized models dealt with in the above references.

In addition, the paper differs from others by explicitly listing and explaining the computer program written for the calibration. Programs of this kind may exist at many places, but are not generally available. Many researchers must therefore either write their own programs or resort to less efficient trial-and-error methods.

The computer program presented in this paper maximizes the likelihood of the choice matrix (trip table) of a multinomial logit model with marginal constraints and multiattribute utilities given observed choices (trips) using a combination of gradient search and Newton-Raphson iteration methods. The program has been designed for interactive work at a computer terminal to allow for maximum control of the calibration process by the user.

1. THE PROBLEM

1.1 Choices and Interactions

Consider a population of *decision makers* who have to make choices in a spatial context. Let the decision makers be subdivided into groups or categories, which are assumed to display similar preferences and/or choice behaviors. Such categories may be made up of individuals of a certain kind, households of a certain type, or a population living in a certain location or zone of a city. In reference to transport modeling usage, the size of these groups is indicated by O_i, $i = 1,\ldots,I$, where O stands for origins.

The decision makers face *choice alternatives*. Choice alternatives, too, may be classified into groups of similar character, e.g., jobs of a certain kind, houses of a certain type, or facilities in a certain zone. Again in reference to transport modeling language, the size of these categories is indicated by D_j, $j = 1,\ldots,J$, where D stands for destinations.

The choice alternatives are characterized by *attributes*. It frequently requires more than one attribute to characterize an alternative. Some attributes are perceived similarly by all decision makers, i.e., they vary only over alternative groups j; others are perceived differently by each decision maker group, i.e., they vary over i *and* j. In transportation terms, some attributes are destination-specific (sometimes called attraction variables), some are origin- *and* destination-specific, i.e., associated with trips. To simplify the notation, both kinds of attributes are stored in a three-dimensional matrix $\underset{\sim}{x}$ where x_{ijk}, $k = 1,\ldots,K$ is the vector of K attributes of alternative group j as seen by decision maker group i. Note that for destination-specific attributes the x_{ijk} are equal for all i.

With this notation, a unified spatial choice or interaction model can be derived either as a random utility-maximizing choice model or as an entropy-maximizing spatial interaction model. The discussion partly follows Anas (1981).

1.2 Random Utility Choice Models

The random utility choice model is one possible approach to take account of the many deviations in human behavior from what seems to be the rational norm *within* the framework of the utility-maximizing paradigm. This is achieved by subsuming all unexplained behavior into a random component of the utility function:

$$u^*_{ij} = u_{ij} + \varepsilon_{ij} \tag{1}$$

where u^*_{ij} is the perceived utility of choice alternative group j for decision maker group i, and u_{ij} and ε_{ij} are its deterministic and stochastic components, respectively. The random term ε_{ij} is thought to represent all taste differences between individual decision makers in decision maker group i as well as all unobserved differences between alternatives in alternative group j, plus all measurement and specification errors associated with the u_{ij}.

Furthermore, it is postulated that the deterministic part of the utility function, u_{ij}, can be expressed as a linear function of the attributes of the alternatives:

$$u_{ij} = \sum_k \beta_k x_{ijk} = \underline{\beta}' \underline{x}_{ij} \tag{2}$$

where the β_k are, at the same time, scaling factors and weights needed to aggregate the attributes into a common measure of utility. The vector notation, with the prime indicating transposition, will be used henceforth for brevity.

The random utility model states that decision maker group i will choose alternative group j over alternative j' if $u^*_{ij} > u^*_{ij'}$. The probability that this occurs is

$$p_{j|i} = \text{Prob}(u_{ij} + \varepsilon_{ij} > u_{ij'} + \varepsilon_{ij'} \; ; \; j' = 1,\ldots,J) \tag{3}$$

where $p_{j|i}$ is a conditional probability such that for any i

$$\sum_j p_{j|i} = 1 \tag{4}$$

In addition, it is assumed that the stochastic terms of the utility function, ε_{ij}, are stochastically independent and identically distributed following an extreme value or Gumbel distribution (cf. Domencich and McFadden 1975):

$$\text{Prob}(\varepsilon_{ij} \leq \varepsilon) = \exp\left\{-\exp\left[-\left(\frac{\pi^2}{6\sigma^2}\right)^{\frac{1}{2}} \varepsilon\right]\right\} \tag{5}$$

where σ^2 is the variance of the distribution. If this assumption holds (which is impossible to test), the binomial *logit* model can be derived (cf. Domencich and McFadden 1975):

$$\ln(p_{j|i}/p_{j'|i}) = \left(\frac{\pi^2}{6\sigma^2}\right)^{\frac{1}{2}} (u_{ij} - u_{ij'}) \tag{6}$$

The binomial logit model is in agreement with the *choice axiom* by Luce (1959) stating that the choice ratio of two alternatives depends only on their relative utility and is independent of other alternatives of the choice set. More specifically, the binomial logit model says that the odds of alternative j being preferred over alternative j' are a log-linear function of the difference between the utilities of the two alternatives.

From (4) and (6) the multinomial logit choice model can be derived:

$$p_{j|i} = \frac{\exp\left[\left(\frac{\pi^2}{6\sigma^2}\right)^{\frac{1}{2}} u_{ij}\right]}{\sum_{j'} \exp\left[\left(\frac{\pi^2}{6\sigma^2}\right)^{\frac{1}{2}} u_{ij'}\right]} \tag{7}$$

where $p_{j|i}$ is the conditional probability (for decision maker group i) that of all alternative groups j', j = 1,...,J, alternative group j will be selected. Inserting (2) in (7) and incorporating the root into the β_k yields:

$$p_{j|i} = \frac{\exp(\underline{\beta}'\underline{x}_{ij})}{\sum_{j'} \exp(\underline{\beta}'\underline{x}_{ij'})} \tag{8}$$

where $\underline{\beta}'$ and $\underline{x}_{ij}$ are defined as in (2).

1.3 Entropy-maximizing Spatial Interaction Models

The same model can be derived from information-theoretic principles by using the entropy-maximizing (Wilson 1970) or information-minimizing (Snickars and Weibull 1977) approach. This approach determines the most random prediction of choices (trips) consistent with macro (i.e., aggregate) constraints on the choice matrix by minimizing the information or negative entropy H contained in it:

$$\underset{p_{ij}}{\text{Min}} \; -H = \sum_i \sum_j p_{ij} \ln p_{ij} \tag{9}$$

subject to:

$$p_{ij} \geq 0 \qquad i = 1,\ldots,I;\ j = 1,\ldots,J \tag{10}$$

$$\sum_i \sum_j p_{ij} = 1 \tag{11}$$

$$\sum_i \sum_j p_{ij} x_{ijk} = \bar{x}_k = \frac{1}{T^o} \sum_i \sum_j t^o_{ij} x_{ijk} \qquad k = 1,\ldots,K \tag{12}$$

where the t^o_{ij} are observed choices of decision maker groups i for alternative groups j, and $\bar{x}_k$ is the mean of attribute k over all observed choices. Constraints (10) and (11) state that the p_{ij} are probabilities - note that now absolute probabilities summing up to one over the whole choice matrix are used. Constraint (12) contains the available macro information about the choice matrix. The minimization uses the Lagrangian function

$$L_H = \sum_i \sum_j p_{ij} \ln p_{ij} - \gamma \left(\sum_i \sum_j p_{ij} - 1 \right) - \sum_k \beta_k \left(\sum_i \sum_j p_{ij} x_{ijk} - \sum_i \sum_j t^o_{ij} x_{ijk} \right) \tag{13}$$

with γ and β_k, $k = 1,\ldots,K$ as Lagrangian multipliers. Setting the first partial derivatives of this function to zero gives

$$\frac{\partial L_H}{\partial p_{ij}} = 1 + \ln p_{ij} - \gamma - \sum_k \beta_k x_{ijk} = 0 \quad i = 1,\ldots,I;\ j = 1,\ldots,J \tag{14}$$

$$\frac{\partial L_H}{\partial \gamma} = \sum_i \sum_j p_{ij} - 1 = 0 \tag{15}$$

Rearranging (14) yields

$$p_{ij} = \exp(\gamma - 1)\ \exp(\underline{\beta}'\underline{x}_{ij}) \tag{16}$$

and substituting into (15) and back into (16) gives the absolute choice probabilities

$$p_{ij} = \frac{\exp(\underline{\beta}'\underline{x}_{ij})}{\sum_i \sum_{j'} \exp(\underline{\beta}'\underline{x}_{ij'})} \tag{17}$$

The corresponding conditional choice probabilities are

$$p_{j|i} = \frac{\exp(\underline{\beta}'\underline{x}_{ij})}{\sum_{j'} \exp(\underline{\beta}'\underline{x}_{ij'})} \tag{18}$$

which is identical to (8).

1.4 Constraints and Expansions

By introducing additional constraints, the basic choice or interaction model (8) or (18), respectively, can be diversified to fit different planning problems or data situations. Moreover, by expanding the model by mass terms expressing the *number* or *size* of the decision maker and/or alternative groups, the model can be adapted to situations where the number of decision makers (demand) and the number of alternatives (supply) are not equal. Constraints and expansions introduce the dimensions of the problem into the model, which means that henceforth the model results are not choice probabilities p_{ij} or $p_{j|i}$, but predicted choices or trips t_{ij}.

Following Wilson's (1970) classification of spatial interaction models, six model types can be distinguished. They are summarized in Figure 1.

As can be seen, there is one unconstrained model, two production-constrained and attraction-constrained models, and one production-attraction-constrained model. The unconstrained model is not really unconstrained, but is constrained only by the requirement that T, the total of all predicted choices (trips), equals T^O, the total of all observed trips. The two production-constrained models are constrained by the requirement that, in addition, the number of decision makers in each decision maker group (or the number of trip origins in each origin zone) is known and is to be matched in the predicted choice matrix (trip table). Similarly, in the two attraction-constrained models the number of alternatives in each alternative group (or the number of trip destinations in each destination zone) is known and is to be matched in the predicted choice matrix (trip table). In the doubly constrained model both the decision maker or origin vector as well as the alternative or destination vector are known and are to be reproduced in the choice matrix.

The two types of production-constrained models differ in that one is unexpanded and one is expanded. The unexpanded type is the *multinomial logit model* in its pure form, which allocates a known number of decision makers or trip origins O_i to alternatives of equal size, but possibly different utility. The expanded version includes a mass term D_j, which accounts for the fact that the choice set is subdivided into alternative groups or zones of possibly different size. Similarly, the expanded version of the attraction-constrained model includes a mass term O_i to account for decision maker groups or origin zones of possibly different size.

The notation in Figure 1 is the usual compressed form where the inverse of the denominator is called a *balancing factor* and is included in the enumerator as A_i, B_j, or C. Note that the doubly constrained model has two balancing factors, A_i and

MODEL 1: UNCONSTRAINED (COD)

$$T_{ij} = C\, O_i D_j \exp(\underline{\beta}' \underline{x}_{ij})$$

$$C = T^O / \sum_i \sum_j O_i D_j \exp(\underline{\beta}' \underline{x}_{ij})$$

MODEL 2: PRODUCTION-CONSTRAINED (AO)

$$T_{ij} = A_i O_i \exp(\underline{\beta}' \underline{x}_{ij})$$

$$A_i = 1 / \sum_j \exp(\underline{\beta}' \underline{x}_{ij})$$

MODEL 3: PRODUCTION-CONSTRAINED (AOD)

$$T_{ij} = A_i O_i D_j \exp(\underline{\beta}' \underline{x}_{ij})$$

$$A_i = 1 / \sum_j D_j \exp(\underline{\beta}' \underline{x}_{ij})$$

MODEL 4: ATTRACTION-CONSTRAINED (BD)

$$T_{ij} = B_j D_j \exp(\underline{\beta}' \underline{x}_{ij})$$

$$B_j = 1 / \sum_i \exp(\underline{\beta}' \underline{x}_{ij})$$

MODEL 5: ATTRACTION-CONSTRAINED (BOD)

$$T_{ij} = B_j O_i D_j \exp(\underline{\beta}' \underline{x}_{ij})$$

$$B_j = 1 / \sum_i O_i \exp(\underline{\beta}' \underline{x}_{ij})$$

MODEL 6: DOUBLY CONSTRAINED (ABOD)

$$T_{ij} = A_i B_j O_i D_j \exp(\underline{\beta}' \underline{x}_{ij})$$

$$A_i = 1 / \sum_j B_j D_j \exp(\underline{\beta}' \underline{x}_{ij})$$

$$B_j = 1 / \sum_i A_i O_i \exp(\underline{\beta}' \underline{x}_{ij})$$

Figure 1. The six model types.

B_j, which are mutually interdependent. The sequence of balancing factors and origin and destination terms in the model equations is used to identify an easy-to-remember acronym for each model type.

For spatial planning purposes, model types 3 (AOD) and 6 (ABOD) are most widely used. The production-constrained Model 3 (AOD) is a general *location* model distributing all kinds of activities O_i such as households, jobs, shops, or services over competing locations D_j such as zones, vacant dwellings, or vacant land. The doubly constrained Model 6 (ABOD) is the basic model for *trip distribution* in transportation planning, where both origins O_i and destinations D_j are projected exogenously. Various applications of these two model types are discussed, for instance, in Wilson (1974), Batty (1976), or Foot (1981). Model 2 (AO) predicts choices between equal sized alternatives and may thus be viewed as the disaggregate version of Model 3 (AOD). This model is extensively used in disaggregate travel demand modeling, in particular for mode and route choice (see, for instance, Domencich and McFadden 1975). The two attraction-constrained models, Model 4 (BD) and Model 5 (BOD), are used much less frequently, because they present some calibration problems (which will be shown later) and can be equally well reformulated as the corresponding production-constrained model, i.e., either as Model 2 (AO) or Model 3 (AOD), just by exchanging subscripts. The unconstrained Model 1 (COD) is of no practical importance and has been included only for demonstration purposes.

1.5 Variations

Although the six models presented above cover a wide range of potential applications, there are some widely used variations, which may continue to be of interest. Such variations include different forms of the utility function or of the attraction term D_j. It will now be shown that by a simple logarithmic transformation, some of these can be incorporated into the six standard models.

(a) The Power Function

In some applications, especially in trip distribution modeling, it may be desired to use the power function instead of the exponential function as the spatial deterrence term, e.g.:

$$t_{ij} = A_i B_j O_i D_j c_{ij}^{-\beta} \tag{19}$$

where c_{ij} is a measure of travel cost. It is easily seen that this is equivalent to

$$t_{ij} = A_i B_j O_i D_j \exp(-\beta \ln c_{ij}) \tag{20}$$

which is a one-parameter version of Model 6 (ABOD).

(b) The Tanner Function

Another alternative to the exponential form of the spatial deterrence function is the function proposed by Tanner (1961). A trip distribution model using the Tanner function

$$t_{ij} = A_i B_j O_i D_j c_{ij}^{-\beta_1} \exp(-\beta_2 c_{ij}) \tag{21}$$

can be transformed into a two-parameter version of Model 6 (ABOD):

$$t_{ij} = A_i B_j O_i D_j \exp(-\beta_1 \ln c_{ij} - \beta_2 c_{ij}) \tag{22}$$

(c) Weighted Attraction Terms

In production-constrained location models of the type of Model 3 (AOD), the term expressing the attraction of the competing alternatives or zones sometimes is not a single variable, D_j, but a multiplicative aggregate of attributes with exponents as weights (see, for instance, the residential location model by Putman and Ducca 1978b; Putman 1980):

$$t_{ij} = A_i O_i \prod_k x_{jk}^{\alpha_k} c_{ij}^{-\beta_1} \exp(-\beta_2 c_{ij}) \qquad (23)$$

In this case, the t_{ij} are residents employed in zone i allocated to zone j. This model is equivalent to the following multi-attribute version of Model 2 (AO):

$$t_{ij} = A_i O_i \exp(\sum_k \alpha_k \ln x_{jk} - \beta_1 \ln c_{ij} - \beta_2 c_{ij}) \qquad (24)$$

However, even if the attraction term is a single variable, it may have an exponent to account for, say, effects of scale as, for instance, in the following version of the Lakshmanan-Hansen (1965) shopping model:

$$t_{ij} = A_i O_i D_j^{\alpha} c_{ij}^{\beta} \qquad (25)$$

where the t_{ij} are shopping expenditures of customers from zone i in zone j, and the O_i are total expenditures of i. This model could be written as an AO or AOD model:

$$t_{ij} = A_i O_i \exp(\alpha \ln D_j - \beta \ln c_{ij}) \qquad (26)$$

or

$$t_{ij} = A_i O_i D_j \exp(\alpha \ln D_j - \beta \ln c_{ij}) \qquad (27)$$

The latter formulation distinguishes between quantitative and qualitative effects of the size of the shopping facilities in j.

(d) Other Variations

There are still other model variations that cannot be transformed into the six standard models. For instance, models that are multiplicative in the exponent (cf. Anas 1975) or have otherwise nonlinear utilities (cf. Wegener 1981) cannot be estimated directly. In these cases, the utilities have to be determined in a separate procedure before they can be entered into one of the models.

1.6 The Calibration Problem

In all of the above models, the parameter vector $\underline{\beta}$ determines how well the model reproduces actual choice or travel behavior subject to the given data and constraints. Calibrating a spatial choice model, therefore, means finding the set of values of $\underline{\beta}$ that yields the closest possible correspondence between the choices (trips) predicted by the model and actual choices (trips) observed in reality. It is assumed throughout that a matrix of observed choices (trips) considered relevant for the problem at hand is available.

It is the purpose of this paper to propose a method for estimating the optimal vector $\underline{\beta}$ for this range of models from a given choice matrix and to present a reliable and efficient computer program for executing this estimation.

2. THE ALGORITHM

2.1 The Super Model

To calibrate the range of models presented in the preceding section, a hybrid *super model* incorporating all terms of all six models has been devised:

$$t_{ij} = A_i B_j C O_i D_j \exp(\underline{\beta}' \underline{x}_{ij}) \tag{28}$$

where the constraints are:

$$A_i = 1 \Big/ \left[\sqrt{C} \sum_j B_j D_j \exp(\underline{\beta}' \underline{x}_{ij})\right] \tag{29}$$

$$B_j = 1 \Big/ \left[\sqrt{C} \sum_i A_i O_i \exp(\underline{\beta}' \underline{x}_{ij})\right] \tag{30}$$

$$C = T^O \Big/ \sum_i \sum_j A_i B_j O_i D_j \exp(\underline{\beta}' \underline{x}_{ij}) \tag{31}$$

the absolute choice probabilities (C cancels out)

$$p_{ij} = \frac{A_i B_j O_i D_j \exp(\underline{\beta}' \underline{x}_{ij})}{\sum_i \sum_j A_i B_j O_i D_j \exp(\underline{\beta}' \underline{x}_{ij})} \tag{32}$$

and the conditional choice probabilities (A_i, O_i, and C cancel out)

$$p_{j|i} = \frac{B_j D_j \exp(\underline{\beta}' \underline{x}_{ij})}{\sum_j B_j D_j \exp(\underline{\beta}' \underline{x}_{ij})} \tag{33}$$

From the super model, all other models can be derived by setting terms not needed to unity. This is illustrated in Table 1.

Table 1. Derivation of model types from the super model.

Model	A_i	B_j	C	O_i	D_j
1 (COD)	1	1	C	O_i	D_j
2 (AO)	A_i	1	1	O_i	1
3 (AOD)	A_i	1	1	O_i	D_j
4 (BD)	1	B_j	1	1	D_j
5 (BOD)	1	B_j	1	O_i	D_j
6 (ABOD)	A_i	B_j	1	O_i	D_j

2.2 Maximum Likelihood

The problem addressed in this paper can now be restated as finding the best-fit parameter vector $\underline{\beta}$ for the super model in its various realizations. As indicated earlier, the maximum likelihood of the choice matrix has been selected as the criterion of goodness-of-fit.

The maximum likelihood principle states that, given a stochastic model with unknown parameters, that set of parameter values is considered to be the best estimate that has the highest probability of reproducing the data. In this particular context, the stochastic model is the super model defined by (28)-(33),

the unknown parameters are the β, and the data are the observed choices t^o_{ij}. The probability that from T^o independent choices a choice matrix $\underset{\sim}{t}$ will be generated is (up to a constant)

$$L(\underline{\beta}) = \prod_i \prod_j p_{ij}(\underline{\beta})^{t^o_{ij}} \tag{34}$$

where $L(\underline{\beta})$ is the likelihood function. Maximum likelihood estimation of the parameter vector $\underline{\beta}$ consists of finding that vector $\underline{\beta}$ that maximizes the likelihood function or its logarithm

$$L^*(\underline{\beta}) = \sum_i \sum_j t^o_{ij} \ln p_{ij}(\underline{\beta}) \tag{35}$$

Maximum likelihood estimation of the parameters of a choice or interaction model then means to maximize the loglikelihood function (35) for the super model (32)

$$\max_{\underline{\beta}} L^*(\underline{\beta}) = \sum_i \sum_j t^o_{ij} \ln \left[\frac{A_i B_j O_i D_j \exp(\underline{\beta}' \underline{x}_{ij})}{\sum_i \sum_j A_i B_j O_i D_j \exp(\underline{\beta}' \underline{x}_{ij})} \right] \tag{36}$$

subject to the constraints (29) and (30)

$$A_i = \frac{1}{\sum_j B_j D_j \exp(\underline{\beta}' \underline{x}_{ij})} \tag{37}$$

$$B_j = \frac{1}{\sum_i A_i O_i \exp(\underline{\beta}' \underline{x}_{ij})} \tag{38}$$

where A_i, B_j, O_i, and D_j are set to unity as specified in Table 1 to account for the different model types.

2.3 Derivatives

For maximizing (36), first it is temporarily assumed that A_i, B_j, O_i, and D_j, if present, are constants, although A_i and B_j, as can be seen from (37) and (38), depend on $\underline{\beta}$. Under this assumption, the function (36) is continuous and differentiable

and has a unique maximum. To solve this unconstrained nonlinear optimization problem, a procedure built on a combination of n-dimensional gradient search and Newton-Raphson iteration methods has been developed. Both methods require the first and second derivatives of (36) to be calculated.

For this, equation (36) is rewritten

$$\begin{aligned}\max_{\underline{\beta}} L^*(\underline{\beta}) &= \sum_i \sum_j t^o_{ij} \ln (A_i B_j O_i D_j) + \sum_i \sum_j t^o_{ij} (\underline{\beta}' \underline{x}_{ij}) \\ &\quad - \sum_i \sum_j t^o_{ij} \ln \left[\sum_i \sum_j A_i B_j O_i D_j \exp(\underline{\beta}' \underline{x}_{ij}) \right]\end{aligned} \tag{39}$$

The gradient of first derivatives of this function is

$$g_k(\underline{\beta}) = \frac{\partial L^*(\underline{\beta})}{\partial \beta_k} = \sum_i \sum_j t^o_{ij} x_{ijk} - T^o \sum_i \sum_j p_{ij} x_{ijk} \tag{40}$$

and the Hessian matrix of second derivatives

$$\begin{aligned}h_{k\ell}(\underline{\beta}) &= \frac{\partial g_k(\underline{\beta})}{\partial \beta_\ell} = \frac{\partial^2 L^*(\underline{\beta})}{\partial \beta_k \partial \beta_\ell} = \\ &- T^o \left[\sum_i \sum_j p_{ij} x_{ijk} x_{ij\ell} - \left(\sum_i \sum_j p_{ij} x_{ijk} \right) \sum_i \sum_j p_{ij} x_{ij\ell} \right]\end{aligned} \tag{41}$$

where $k, \ell = 1, \ldots, K$. The maximum of the loglikelihood function (36) is where its gradient (40)

$$g_k(\underline{\beta}) = 0 \qquad k = 1, \ldots, K \tag{42}$$

The derivatives calculated according to (40) and (41) could be used for the estimation of the parameters of all six models contained in the super model. However, for reasons of computational efficiency, it is preferable to work with the conditional probabilities $p_{j|i}$ instead of the absolute probabilities p_{ij}, where this is possible. This is the case with models 2 (AO), 3 (AOD), and 6 (ABOD), in which the production constraint ensures that

$$\sum_j t^o_{ij} = \sum_j t_{ij} \qquad i = 1,\ldots,I \tag{43}$$

and thus

$$T^o \sum_i \sum_j p_{ij} x_{ijk} = \sum_i \left(\sum_j t^o_{ij} \right) \sum_j p_{j|i} x_{ijk} \qquad k = 1,\ldots,K \tag{44}$$

Taking advantage of (44), the gradient (40) can be expressed in terms of conditional probabilities:

$$g_k(\underline{\beta}) = \frac{\partial L^*(\underline{\beta})}{\partial \beta_k} = \sum_i \sum_j t^o_{ij} x_{ijk} - \sum_i \left(\sum_j t^o_{ij} \right) \sum_j p_{j|i} x_{ijk} \tag{45}$$

with the Hessian matrix

$$h_{k\ell}(\underline{\beta}) = \frac{\partial g_k(\underline{\beta})}{\partial \beta_\ell} = \frac{\partial^2 L^*(\underline{\beta})}{\partial \beta_k \partial \beta_\ell} =$$

$$- \sum_i \left(\sum_j t^o_{ij} \right) \left[\sum_j p_{j|i} x_{ijk} x_{ij\ell} - \left(\sum_j p_{j|i} x_{ijk} \right) \sum_j p_{j|i} x_{ij\ell} \right] \tag{46}$$

It can be seen that these are the derivatives of the loglikelihood function of the super model expressed in conditional probabilities (33):

$$\max_{\underline{\beta}} L^*(\underline{\beta}) = \sum_i \sum_j t^o_{ij} \ln \left[\frac{B_j D_j \exp(\underline{\beta}' \underline{x}_{ij})}{\sum_j B_j D_j \exp(\underline{\beta}' \underline{x}_{ij})} \right] \tag{47}$$

In the maximization procedure described below, the conditional probabilities (33) and their derivatives (46) and (47) are used for models 2 (AO), 3 (AOD), and 6 (ABOD), while for the remaining three models the absolute probabilities (32) and their derivatives (40) and (41) are used.

2.4 Maximization

The maximization procedure used is a combination of n-dimensional gradient search and Newton-Raphson iteration methods. The reason for using two different numerical techniques lies in the

different characteristics of these two methods; the gradient search method in general has a high probability of convergence, but tends to be slow on flat solution surfaces. The Newton-Raphson technique is usually much faster, but can diverge from bad starting values. For a discussion of these and other techniques for nonlinear parameter estimation, see among others Bard (1974), Schwetlick (1979), Stopher and Meyburg (1979), Manski and McFadden (1981), and Churchhouse (1981).

Both methods start from initial estimates of the values of $\underline{\beta}$ and proceed by iteratively improving them, until an optimum is reached:

$$\beta_k(n+1) = \beta_k(n) + \Delta\beta_k(n) \qquad k = 1,\ldots,K \tag{48}$$

where n is the number of the iteration. The two methods differ, however, in the manner by which the parameter increments $\Delta\beta_k$ are determined.

The *gradient* method uses the gradient $\underline{g}$ to establish the direction of steepest ascent on the solution surface and uses the Hessian matrix $\underset{\sim}{h}$ to determine a stepsize for movement along that direction:

$$\Delta\beta_k = \frac{\underline{g}'\underline{g}}{|\underline{g}'\underset{\sim}{h}\underline{g}|}\, g_k \qquad k = 1,\ldots,K \tag{49}$$

where again the prime denotes transposition, and $\underline{g}'\underline{g}$ is the gradient norm.

The *Newton-Raphson* method, on the other hand, solves the system of K nonlinear equations (42) by approximation using Taylor series expansion truncated after the first order term:

$$g_k(\underline{\beta}) + \sum_{\ell} h_{k\ell}(\underline{\beta})\Delta\beta_\ell = 0 \qquad k = 1,\ldots,K \tag{50}$$

or

$$\sum_{\ell} h_{k\ell}(\underline{\beta})\Delta\beta_\ell = -g_k(\underline{\beta}) \qquad k = 1,\ldots,K \tag{51}$$

This system of K linear equations with K unknowns can be solved by a standard numerical technique like the Gauss-Jordan method (cf. Churchhouse 1981) to yield a set of $\Delta\beta_k$.

Thus both methods are interchangeable, but of different effectiveness. The strategy is to use the more efficient Newton-Raphson technique as much as possible and to use the gradient technique only where divergence seems to occur. The optimization process comes to an end when a convergence criterion is met.

2.5 Introducing the Constraints

The optimization procedure described so far is unconstrained, i.e., it disregards constraints (29)-(31). These constraints are introduced numerically *between* iterations by calculating new balancing factors A_i and B_j, where applicable, after each iteration and feeding them back into the calculation of the probabilities (32) or (33) in the next iteration. This is equivalent to numerical approximation of the Lagrangian multipliers of the corresponding constrained optimization problem.

The balancing factors A_i and B_i are calculated using the biproportional adjustment technique known as Furness or Fratar method in transportation planning and as the RAS technique in input-output analysis. Willekens et al. (1979) and Willekens (1980) have shown the close relationship of this technique to the entropy-maximizing method. Here the original RAS algorithm suggested by Stone (1963) for the updating of input-output matrices is used.

The RAS algorithm, like all biproportional adjustment techniques, adjusts the elements of a given two-dimensional matrix such that (a) given constraints on the marginal sums of the matrix are satisfied and (b) the adjusted elements of the matrix stay as close as possible to their prior values. In other words, if $\underset{\sim}{a}_o$ is the prior matrix, the RAS algorithm seeks to determine two vectors $\underline{r}$ and $\underline{s}$ such that

$$\underset{\sim}{a} = \underline{r}' \underset{\sim}{a}_o \underline{s} \tag{52}$$

where $\underset{\sim}{a}$ is the posterior (adjusted) matrix satisfying the constraints, hence the algorithm's name. It is obvious that, if $\underset{\sim}{a}_o$ is replaced by the predicted choice matrix $\underset{\sim}{t}$, $\underline{r}$ and $\underline{s}$ contain the desired balancing factors A_i and B_j.

The RAS algorithm proceeds by iteration. In each iteration, first the rows and then the columns of the matrix are adjusted such that

$$t_{ij}(m+1) = t_{ij}(m) O_i \; / \sum_j t_{ij}(m) \qquad i = 1,\ldots,I \quad (53)$$

and

$$t_{ij}(m+2) = t_{ij}(m+1) D_j \; / \sum_i t_{ij}(m+1) \qquad j = 1,\ldots,J \quad (54)$$

where m is the number of half-iterations. The algorithm is certain to converge to a unique optimum and ends when a convergence criterion is met.

The balancing factors A_i and D_j can be derived by calculating in each iteration

$$A_i(m+1) = A_i(m) O_i \; / \sum_j t_{ij}(m) \qquad i = 1,\ldots,I \quad (55)$$

and

$$B_j(m+2) = B_j(m+1) D_j \; / \sum_j t_{ij}(m+1) \qquad j = 1,\ldots,J \quad (56)$$

Note that initially all A_i and B_j are set to unity and that they retain their updated values between calls of the RAS algorithm to speed up convergence.

Of course, the above iterative adjustment applies only to the doubly constrained Model 6 (ABOD) where O_i and D_j are to be matched by the row and column sums, respectively. For the two attraction-constrained Models 4 (BD) and 5 (BOD), only the second half of the RAS algorithm, i.e., equations (54) and (56), need to be passed, and this requires no iteration. No pass through the RAS algorithm is necessary for estimating the remaining Models 1 (COD), 2(AO), and 3 (AOD), because in these models the balancing factors do not affect the estimation results.

3. THE PROGRAM

3.1 Program Organization

The above calibration algorithm has been implemented in a computer program called LOGIT.

LOGIT is written in Fortran. It consists of a short main program and 19 subroutines. Each subroutine performs a specific task and returns its result to the calling program. Figure 2 represents the hierarchical organization of LOGIT:

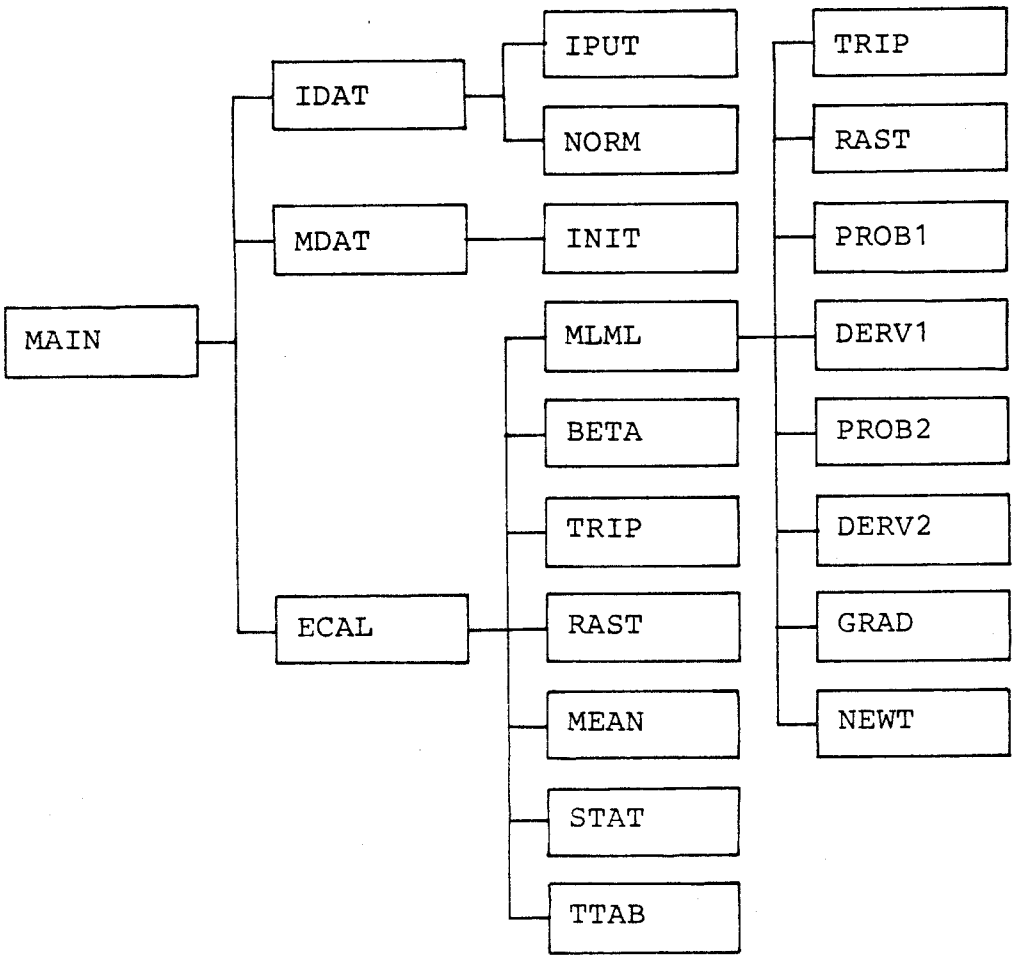

Figure 2. Program organization of LOGIT.

The program is organized such that with one set of input data several *calibrations* using different constraints and/or attributes can be performed. This is the outer loop of the program. Within each calibration, the optimal-fit parameter vector is approached by iteration. This is the inner loop. Figure 3 is a flow diagram showing the normal flow of information within the program. In addition, but not shown in Figure 3, there are options permitting the user to interrupt the iteration process, inspect intermediate results and then continue, restart, or end the calibration.

Two kinds of subprograms can be distinguished. The first one performs calculations and input and output operations. The second one leads a dialog with the user. The main program and the first-level subroutines IDAT, MDAT, and ECAL are of the second kind. They prompt the user for information or decisions necessary for running the program. While everything has been done in these subprograms to make interaction with the program as convenient as possible, no effort has been made to anticipate or correct user errors.

3.2 Subprograms

In this section, the 20 subprograms of LOGIT are briefly discussed. The source code of all subprograms is listed in the Appendix.

MAIN: The main program controls the outer loop of the program. It calls IDAT once and MDAT and ECAL once for each calibration.

IDAT: IDAT prompts the user for the dimension of the problem: the number of origins (decision maker groups), destinations (alternative groups), and attributes. The number of origins need not to be equal to the number of destinations, i.e., the trip table (choice matrix) need not to be square. Two kinds of attributes are distinguished: attributes that vary only over destinations and attributes that vary over origins *and* destinations (see section 1.1).

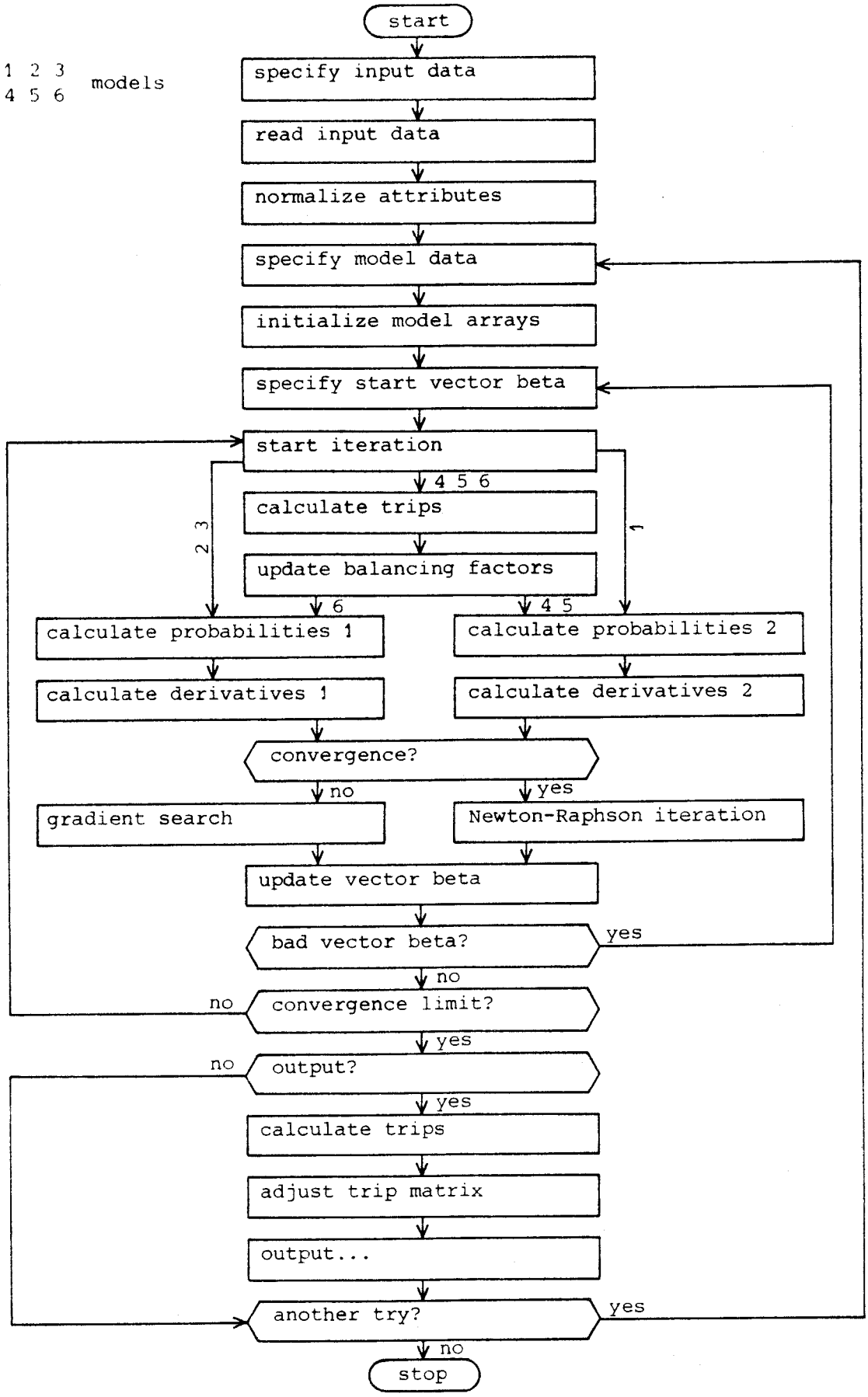

Figure 3. Flow diagram of LOGIT.

IPUT: This subroutine reads the input data. The program accepts three kinds of data:

(a) Origins and/or destinations: These are optional. If absent, origins and destinations will be inferred by aggregation from the observed trip matrix.

(b) Observed trip matrix: Row and column sums of this matrix need not coincide with the above origins and destinations.

(c) Attributes: These can be either attributes of destinations or of trips.

Format and organization of the input dataset are specified in section 3.3.

NORM: All attributes read are subsequently normalized such that the value halfway between their extremes is between -1 and +1. This serves three purposes. First, it separates the scaling and the weighting function of the parameters and thus makes them comparable. Second, it contributes to keeping parameters in a range acceptable for exponents by computers. Third, it increases the precision of the parameter estimates; precision is expressed in *significant* digits rather than in digits. The normalizing factor of each attribute is stored for later use.

MDAT: For each new calibration, MDAT prompts the user for model type, specification of origins and destinations, and selection of attributes. Attributes may be selected from the attributes present on the input dataset in any order.

INIT: Depending on these specifications, INIT initializes model arrays, in particular origins, destinations, and balancing factors. The balancing factors are always set to unity.

ECAL: This subroutine controls the execution of one calibration. It asks for a start vector of betas as initial values and calls MLML, which performs the parameter estimation. In addition, ECAL handles the options permitting the user to look at intermediate results, restart the calibration, or specify the program output.

MLML: This subroutine controls the parameter estimation process, i.e., the iterative maximization of the likelihood function. Depending on the model type selected, MLML in each iteration calls the requisite subroutines calculating balancing factors, probabilities, and derivatives. For the calculation of parameter changes, MLML normally calls the Newton-Raphson procedure NEWT. However, if the parameter changes, instead of getting smaller, continue to increase considerably over more than one iteration, MLML assumes that divergence is occurring and calls the more reliable gradient search procedure GRAD. In the first iteration, GRAD is always called to avoid divergence due to bad starting values. MLML also checks parameter values and if they become too large asks for a new start vector. If the parameter changes approach zero at the level of five significant digits, it is assumed that the optimum has been reached and the iteration process is stopped.

PROB1: This subroutine calculates the choice probabilities following equation (33). These probabilities add up to unity in each row and thus are appropriate for model types 2, 3, and 6.

DERV1: In this subroutine, the first and second partial derivatives of the likelihood function are calculated according to equations (45) and (46), respectively, using the probabilities calculated in PROB1.

PROB2: PROB2 is equivalent to PROB1, except that the probabilities are calculated as in equation (32), i.e., add up to unity over all rows of the choice matrix. These probabilities are used for model types 1, 4, and 5.

DERV2: DERV2 is equivalent to DERV1, except that equations (40) and (41) and the probabilities calculated in PROB2 are used.

GRAD: Subroutine GRAD is called when the gradient search method is to be applied. GRAD calculates an increment to each parameter according to equation (49).

NEWT: Subroutine NEWT is called when the Newton-Raphson method is to be applied. NEWT calculates the parameter increments by solving the system of linear equations (51) using the Gauss-Jordan method (cf. Churchhouse 1981).

BETA: BETA writes the estimated parameter values on the terminal and/or on the output printer file. Note that these parameter values have to be multiplied by their associated normalizing factors.

TRIP: This subroutine generates the trip table or choice matrix following equation (28) using the new parameter estimates and the balancing factors of the previous iteration. *During* the estimation process, calculating trips is necessary only for model types 4 (BD), 5 (BOD), and 6 (ABOD), while *after* the estimation the subroutine is used to generate the trip table for the output dataset regardless of model type.

RAST: This is the RAS algorithm for matrix adjustment according to equations (53)-(56). *During* the estimation process, the subroutine is called only for model types 4 (BD), 5 (BOD), and 6 (ABOD). *After* the estimation, when preparing the output dataset, the subroutine is used for all model types except Model 1 (COD).

MEAN: This subroutine calculates totals and/or means of trips and attributes of the observed and predicted trip table. Observed and predicted values are equal if origins and destinations are taken from the observed trip table, but may differ if other origins and/or destinations have been specified. Note that the normalized attributes now have been restored to their original magnitudes.

STAT: In this subroutine, a number of statistics expressing the goodness-of-fit between the observed and the predicted trip table are calculated and written on the terminal and/or the output printer file. The following statistics are used:

(a) loglikelihood ratio: the ratio between the maximum value of the loglikelihood function achieved in the calibration and the maximum possible value. A ratio of one would result if both trip tables were identical:

$$LLR \equiv \frac{\sum_i \sum_j t_{ij}^o \ln t_{ij}}{\sum_i \sum_j t_{ij}^o \ln t_{ij}^o} \qquad (57)$$

Note that the constant term $-T^o \ln T^o$ has been dropped in the denominator and in the numerator to make the measure more sensitive.

(b) slope b and intercept a of a regression line

$$t_{ij}^o = bt_{ij} + a$$

(c) correlation coefficient r, coefficient of determination r^2*, and t of* r^2 in their usual meaning

(d) mean absolute percentage error calculated as

$$MAPE \equiv \frac{\sum_i \sum_j |t_{ij}^o - t_{ij}|}{\sum_i \sum_j t_{ij}^o} 100 \qquad (58)$$

These statistics are most meaningful if origins and destinations are aggregated from the observed trip table. If other origins and destinations are specified, the statistics represent the combined effect of differences in the constraints *and* of errors in the prediction.

TTAB: This subroutine writes the predicted trip table (choice matrix) on the output dataset. The format of a trip table on this dataset is identical to that of the observed trip table on the input data set (see section 3.3).

3.3 Input

Input is entered to the program through the user's terminal (Fortran number 5) and an input dataset (Fortran number 8). While input requests by the program at the terminal are self-explanatory, the organization and format of the input dataset need to be specified.

The input dataset is organized by record groups. Each record carries a record group identification. Within each record group, the records are sorted in ascending order. Record group identification and sorting number are not read by the program.

The dataset consists of 80-byte card-image records with the following format:

column 1-4	record group identification
column 5-8	sorting number
column 9-10	blank
column 11-70	10 data fields, 6 columns each
column 71-72	blank
column 73-80	sequence number

The data fields may or may not contain a decimal point at any desired position. On some computers, no sequence numbers are recognized.

There are four kinds of data on the input dataset: (1) origins and destinations, (2) observed trips, (3) attributes of

zones, and (4) attributes of trips. They are stored in this order on the dataset:

(1) *Origins and destinations:* The first records contain first the origins, ten to a record, and then the destinations. If I is the number of origin zones, (I - 1)/10 + 1 records are needed for the origins. If J is the number of destination zones, (J - 1)/10 + 1 records are needed for the destinations. Origin and destination records may be blank if origins and destinations are to be taken from the observed trip table.

(2) *Observed trip table:* Observed trips are stored ten to a record in (J - 1)/10 + 1 record groups, each containing I records. Hence, I(J - 1)/10 + 1 records are needed to store the trip table. Within each record group, ten columns of the trip table are stored (possibly less in the final record group). To give an example, for a 30-zone system the following records will result:

record 1-30	trips to zones 1-10
record 31-60	trips to zones 11-20
record 61-90	trips to zones 21-30

(3) *Attributes of zones:* Zonal attributes are stored from left to right on one record per zone. Thus J records are needed for storing zonal attributes. If no zonal attributes are present, no zonal attribute records must be included in the dataset.

(4) *Attributes of trips:* Trip attributes are stored as matrices of exactly the same format as the observed trip table. Any number of attribute matrices up to the maximum number of attributes may be present. If no trip attributes exist, the dataset ends after the zonal attributes records.

The program is presently dimensioned to handle up to 30 origin zones (decision maker groups), up to 30 destination zones (alternative groups), and up to 8 zonal or trip attributes.

The last page of the Appendix contains a test dataset for a 10-zone system with no zonal attributes and four trip attributes.

3.4 Output

Output is written by the program to the user's terminal (Fortran number 6), to a printer file (Fortran number 7), and to a card-image output dataset (Fortran number 9). Except terminal output, all output is optional.

The printer file contains the estimation results and statistics as produced by subroutines BETA, MEAN, and STAT. The card-image dataset contains the predicted trip table in the same format as the observed trip table on the input dataset.

3.5 Portability

The program LOGIT is written in a subset of Fortran 77 that should be compatible with any Fortran IV compiler (if the few CHARACTER specifications are removed). The program requires no other subroutines or functions except standard functions.

Although the program is presently dimensioned to handle a 30-zone system and up to 8 attributes, these dimensions can easily be adapted to larger problems.

To facilitate portability, input and output have been deliberately kept primitive on the assumption that researchers working in this field have at their disposal programs for processing and displaying data of this kind.

4. AN APPLICATION

4.1 The Data

In this section, an illustrative application of the program LOGIT will be presented. The data for this application have been taken from a project on spatial change processes in the urban region of Dortmund, FRG (cf. Wegener 1982). The region has a population of about 2.4 million and is subdivided into 30 zones in the project.

The data used in this application are work trip data of the year 1970 and travel times and travel costs of both the public transport and the highway system. No origins or destinations different from those of the observed work trip table are provided, nor are any zonal attributes or attraction variables. Thus, the input dataset for this 30-zone system consists of six blank records (substituting for the origins and destinations) plus five blocks of matrix data with 90 records each (one matrix containing observed work trips and four matrices containing trip attributes).

For demonstration purposes, also a 20-zone system and a 10-zone system have been artifically created by taking the innermost 20 or 10 zones of the 30-zone system, respectively, discarding the rest of the region. The test dataset listed in the Appendix is the input dataset of the 10-zone system.

The travel time and travel cost data were derived from a transportation model based on public transport and highway link data and employing minimum-path and congestion-sensitive assignment techniques. Public transport travel times include access, waiting, in-vehicle, and transfer waiting time. Car travel times include access, driving, congestion, and parking-search time. Public transport costs are based on a flat fare plus a distance-dependent component. Car travel costs only include out-of-pocket costs of a car trip, i.e., gasoline costs and parking fees.

4.2 A Small Model

First, a very small application example will be presented in detail. It uses the 10-zone system with a reduced dataset: only the two travel time attributes are considered. The reduced input dataset is shown in Figure 4.

The task is to estimate the parameters of a *trip distribution* model from these two attributes such that (a) the predicted origins and destinations equal the origins and destinations of the observed trip table, and (b) the predicted flows are as close to the observed flows as possible. Obviously, the appropriate model type is Model 6 (ABOD).

oo 1 1											origins
do 1 1											destinations
to 1 1 1	15490	2900	2957	285	504	471	231	634	375	454	
to 1 1 2	6442	14601	3203	647	228	627	292	590	336	238	
to 1 1 3	8552	3284	10780	241	153	996	385	880	341	204	
to 1 1 4	3000	3133	1199	7183	225	260	106	245	124	74	
to 1 1 5	3333	4210	1917	307	5404	1063	357	493	153	100	
to 1 1 6	4827	3125	3188	269	536	6721	733	901	221	138	
to 1 1 7	3275	1474	1807	144	102	518	6275	2997	264	107	
to 1 1 8	3465	1408	2105	171	53	313	736	9249	467	120	
to 1 1 9	5173	2372	1947	232	87	298	226	1390	5452	237	
to 1 110	3979	1397	1088	152	371	221	118	315	260	7098	observed trips
tt 1 1 1	13.0	26.5	24.6	44.1	47.4	34.7	48.7	37.5	40.9	45.6	
tt 1 1 2	26.5	9.0	29.8	38.5	41.9	41.6	57.2	42.8	47.3	58.8	
tt 1 1 3	24.6	31.4	15.0	51.2	53.4	29.1	52.6	37.1	50.2	45.6	
tt 1 1 4	44.1	38.6	51.2	18.5	55.8	65.8	84.6	64.2	76.2	72.8	
tt 1 1 5	47.4	41.9	51.8	57.7	25.0	45.5	55.4	59.5	69.3	81.5	
tt 1 1 6	34.6	42.0	29.1	67.2	45.4	23.0	44.7	42.4	61.7	59.6	
tt 1 1 7	48.7	57.2	52.7	80.3	55.4	44.7	22.0	40.7	65.9	82.6	
tt 1 1 8	37.6	43.2	37.1	64.1	59.4	42.4	40.7	18.0	49.9	66.6	
tt 1 1 9	40.9	46.9	50.5	72.0	68.9	60.6	65.9	49.9	23.0	52.4	
tt 1 110	45.6	58.8	45.6	73.7	79.5	59.6	82.6	66.6	52.4	20.0	attribute 1
tt 2 1 1	13.2	17.7	17.9	21.2	23.9	28.2	25.5	24.2	21.1	22.4	
tt 2 1 2	16.4	10.5	18.1	16.3	19.6	28.0	25.8	25.1	23.2	24.5	
tt 2 1 3	17.7	18.9	13.7	21.1	22.5	22.8	20.8	22.3	21.3	24.1	
tt 2 1 4	23.4	18.8	24.2	15.5	19.5	31.8	31.6	31.2	30.0	31.1	
tt 2 1 5	27.0	22.1	26.5	19.5	22.0	20.0	25.8	34.8	33.6	36.2	
tt 2 1 6	32.7	32.8	26.4	31.8	20.0	20.0	21.6	30.8	34.2	37.6	
tt 2 1 7	28.0	29.5	22.9	31.6	25.9	21.9	20.0	20.9	26.3	32.8	
tt 2 1 8	27.9	29.4	24.2	32.0	32.9	28.9	19.7	16.0	23.0	31.6	
tt 2 1 9	23.9	26.9	24.6	30.7	32.9	32.9	27.0	23.4	20.0	21.5	
tt 2 110	24.7	27.7	26.3	31.5	34.9	34.1	31.4	30.1	21.6	17.0	attribute 2

Figure 4. Test dataset for small model.

Figure 5 is a protocol of the dialog between the user and the program LOGIT as it would appear on a hardcopy terminal. It can be seen that the program, after having received its directions, writes the results of each inner-loop iteration on the terminal: the first column (*it*) is the number of the current iteration. The second column (*ras*) shows the number of iterations required in the RAS algorithm to reach convergence; this number starts at one (a consequence of choosing two zeros as starting values), then jumps to six and gradually returns to one. The third column monitors the *accumulated absolute change* of the parameter values occurring in this iteration. It can be observed that the gradient search method (which is always called into action in the first iteration) pushes the parameter values from the arbitrarily selected starting values already very close to their final position. The rest is accomplished by the Newton-Raphson method in six more iterations, and it can be seen that the parameter changes decrease rapidly from one iteration to the other.

The program then displays the result of the calibration, i.e., the final parameter values with their normalizing factors (see subroutine NORM). The notation used means that in this case both parameter values have to be multiplied by 10^{-2}. As one might expect, the parameters of both attributes carry a minus sign.

The user may then ask for some statistics about the solution and he will see the results of subroutines MEAN and STAT displayed on the terminal. As is to be expected, total trips, trips per observation, and the means of both attributes are identical for the observed and for the predicted trip matrix. The high loglikelihood ratio seems to indicate a very good fit. Also the slope and intercept of the regression line are very close to one and zero, respectively, where they belong. The correlation coefficient and the r^2-statistic, too, convey a close correspondence between observed and predicted trips. However, the mean average percentage error could be less for a doubly constrained model.

```
 specify input data:
 enter no. of origins (decisionmaker groups)
10
 enter no. of destinations (alternatives)
10
 enter no. of attributes of alternatives
0
 enter no. of trip (distance) attributes
2
 specify model data:
 select model type:
  1  unconstrained (cod)
  2  production-constrained (ao)
  3  production-constrained (aod)
  4  attraction-constrained (bd)
  5  attraction-constrained (bod)
  6  doubly constrained (abod)
6
 specify origins:
  1  take from input origins
  2  aggregate from observed trips (choices)
2
 specify destinations:
  1  take from input destinations
  2  aggregate from observed trips (choices)
2
 attribute selection?
n
 enter start vector beta:  2 number(s)
0,0
 select one:
  1  stop after intermediate steps
  2  continue until final solution
2
   it ras  change  beta 1  beta 2
    1   1  8.8903 -6.7484 -2.1419
    2   6  0.9479 -7.0615 -2.7768
    3   4  0.1464 -7.0589 -2.9205
    4   3  0.0221 -7.0567 -2.9404
    5   2  0.0028 -7.0564 -2.9429
    6   2  0.0004 -7.0563 -2.9432
    7   1  0.0000 -7.0563 -2.9433
 gradient search terminated at iteration   7
 parameter changes less than specified limit

 doubly constrained model (abod):

 beta 1 (1.e-02) . . . . . .   -7.0563
 beta 2 (1.e-02) . . . . . .   -2.9433
```

Figure 5. Calibration of small model.

```
 select one:
  1  new start vector beta
  2  repeat with present balancing factor(s)
  3  statistics
  4  continue until final solution
  5  exit
3

 totals and means:                observed predicted
 observations  . . . . .               100       100
 trips (choices) . . . .            198329    198329
 trips/observation . . .           1983.29   1983.29
 mean of attribute 1 . .            29.932    29.932
 mean of attribute 2 . .            20.068    20.068

 statistics:
 loglikelihood ratio . . . . . .                0.9971
 slope . . . . . .                              1.0021
 intercept . . . . . .                          -4.14
 correlation coefficient r . . . .              0.9892
 r-squared . . . . . .                          0.9784
 t of r-squared  . . . . . .                    66.70
 mean absolute percentage error  .              13.46

 select one:
  1  new start vector beta
  2  repeat with present balancing factor(s)
  3  statistics
  4  continue until final solution
  5  exit
5
 output?
y
 another try?
n
```

Figure 5. Continued.

In Table 2, observed and predicted trips are compared. As specified in the constraints, row and column totals of both trip tables are equal (except for round-off errors during printout). However, a flow-by-flow comparison between observed and predicted trips reveals that the predictive power of the calibrated model, despite most goodness-of-fit statistics being excellent, is no more than satisfactory. This suggests caution towards most goodness-of-fit statistics of spatial interaction models except the MAPE statistic. This view has been expressed also by other researchers (see, for instance, Smith and Hutchinson 1979).

4.3 More Models

In a similar way as demonstrated with the small model, the program was tested with a variety of zonal systems, numbers of attributes, and model types. In particular, it was investigated how sensitive the calibration results are to variations in zones, attributes, or model type, keeping everything else equal. The results of these experiments are summarized in Tables 3-6.

In the first experiment, one model type, Model 6 (ABOD), was tested with different zonal systems and between one and four attributes. It may be recollected that the three zonal systems used are not really different, but that the two smaller ones are subsets of the 30-zone system. Table 3 shows the parameter values estimated in the calibration, and Table 4 shows selected goodness-of-fit measures associated with them.

It can be observed that adding more attributes to the model in general increases its explanatory power, but not in all cases. Most often, a considerable improvement in fit is achieved by adding the second attribute, but only very little is contributed by the third and fourth one. The parameters of three of the four attributes always have the expected negative sign, while the parameters of the third attribute, obviously due to interactions between the attributes, always turn out to be positive. This is disturbing, since this attribute represents public transport fares.

Table 2. Calibration results of small model: observed versus predicted trips.

Observed trips

	1	2	3	4	5	6	7	8	9	10	∑
1	15,490	2,900	2,957	285	504	471	231	634	375	454	24,301
2	6,442	14,601	3,203	647	228	627	292	590	336	238	27,204
3	8,552	3,284	10,780	241	153	996	385	880	341	204	25,816
4	3,000	3,133	1,199	7,183	225	260	106	245	124	74	15,549
5	3,333	4,210	1,917	307	5,404	1,063	357	493	153	100	17,337
6	4,827	3,125	3,188	269	536	6,721	733	901	221	138	20,659
7	3,275	1,474	1,807	144	102	518	6,275	2,997	264	107	16,963
8	3,465	1,408	2,105	171	53	313	736	9,249	467	120	18,087
9	5,173	2,372	1,947	232	87	298	226	1,390	5,452	237	17,414
10	3,979	1,397	1,088	152	371	221	118	315	260	7,098	14,999
∑	57,536	37,904	30,191	9,631	7,663	11,488	9,459	17,694	7,993	8,770	198,329

Predicted trips

	1	2	3	4	5	6	7	8	9	10	∑
1	15,009	3,315	3,110	375	224	581	205	811	399	273	24,302
2	6,005	16,054	2,442	734	427	409	127	619	272	115	27,204
3	8,359	3,264	9,988	329	220	1,457	258	1,272	297	374	25,818
4	3,171	3,499	1,013	6,921	361	149	35	257	65	79	15,550
5	3,670	4,085	1,473	629	4,782	1,435	526	523	155	60	17,338
6	5,246	2,028	5,022	153	824	4,808	868	1,346	178	185	20,658
7	3,100	1,064	1,466	85	476	1,369	6,285	2,827	233	58	16,963
8	3,551	1,496	2,213	138	152	684	884	8,457	414	98	18,087
9	5,627	2,205	1,511	146	139	299	214	1,273	5,362	638	17,414
10	3,796	895	1,954	122	60	298	56	310	618	6,890	14,999
∑	57,534	37,905	30,192	9,632	7,665	11,489	9,458	17,695	7,993	8,770	198,333

Table 3. Calibration results of Model 6 (ABOD) for different zonal systems: parameters.

Number of zones	Number of attributes	beta 1 × 0.01	beta 2 × 0.01	beta 3 × 0.1	beta 4 × 0.1
10	1	-7.9009			
	2	-7.0563	-2.9433		
	3	-7.3850	-3.2000	+0.5592[a]	
	4	-7.4676	-2.5328	+0.5878[a]	-0.1269[a]
20	1	-8.4531			
	2	-6.2621	-6.5190		
	3	-6.2789	-6.5717	+0.2707	
	4	-6.7472	-2.7470	+3.2603	-6.9092
30	1	-9.0090			
	2	-6.0485	-8.2175		
	3	-6.1418	-8.4361	+0.9683	
	4	-6.3721	-6.7160	+2.2035	-2.8184

[a] × 1.0

Table 4. Calibration results of Model 6 (ABOD) for different zonal systems: statistics.

Number of zones	Number of attributes	LLR	Slope	Intercept	r^2	MAPE
10	1	0.9968	1.0028	-16.28	0.9730	14.89
	2	0.9971	1.0021	-4.14	0.9784	13.46
	3	0.9973	0.9910	17.87	0.9780	13.43
	4	0.9973	0.9924	15.10	0.9782	13.48
20	1	0.9936	1.0314	-28.17	0.9781	19.67
	2	0.9954	1.0143	-12.78	0.9868	15.56
	3	0.9954	1.0137	-12.27	0.9868	15.57
	4	0.9956	1.0144	-12.92	0.9877	15.29
30	1	0.9936	1.0127	-12.25	0.9949	17.43
	2	0.9956	1.0070	-6.72	0.9973	13.01
	3	0.9956	1.0072	-6.97	0.9973	13.10
	4	0.9956	1.0088	-8.48	0.9973	13.06

Table 5. Calibration results of different model types for 30-zone system: parameters.

Model	Number of attributes	beta 1 × 0.01	beta 2 × 0.01	beta 3 × 0.1	beta 4 × 0.1
1 (COD)	1	-7.6210			
	2	-7.0815	-1.0815		
	3	-5.5109	-0.9941	-10.0975	
	4	-4.8385	-6.8728	-16.2628	+11.3280
2 (AO)	1	-9.3440			
	2	-8.8248	-1.4775		
	3	-8.0403	-0.1299	-7.2623	
	4	-10.0317	+16.5646	+4.9628	-29.5222
3 (AOD)	1	-9.1192			
	2	-6.0140	-8.0077		
	3	-5.9998	-7.9840	-0.1267	
	4	-5.8502	-9.2271	-1.3382	+2.4552
4 (BD)	1	-9.2288			
	2	-8.1183	-3.0334		
	3	-7.0052	-1.7155	-9.2298	
	4	-7.7352	+4.0910	-4.5550	-10.4606
5 (BOD)	1	-8.3357			
	2	-6.5403	-4.6992		
	3	-5.8937	-3.7171	-5.7258	
	4	-6.8097	+2.8541	+0.1330	-11.8827
6 (ABOD)	1	-9.0090			
	2	-6.0485	-8.2175		
	3	-6.1418	-8.4361	+0.9683	
	4	-6.3721	-6.7160	+2.2035	-2.8184

Table 6. Calibration results of different model types for 30-zone system: statistics.

Model	Number of attributes	LLR	Slope	Intercept	r^2	MAPE
1 (COD)	1	0.9702	0.8238	170.21	0.8692	57.97
	2	0.9704	0.8417	152.94	0.8594	57.88
	3	0.9721	0.8194	174.42	0.8810	55.41
	4	0.9728	0.8272	166.91	0.8825	55.60
2 (AO)	1	0.9778	1.1425	-137.64	0.9633	40.94
	2	0.9779	1.1451	-140.15	0.9640	40.73
	3	0.9786	1.1300	-125.55	0.9712	38.99
	4	0.9819	1.1232	-119.02	0.9701	34.63
3 (AOD)	1	0.9910	1.0010	-0.94	0.9889	22.04
	2	0.9935	1.0127	-12.31	0.9914	18.80
	3	0.9935	1.0126	-12.18	0.9914	18.79
	4	0.9935	1.0126	-12.16	0.9916	18.62
4 (BD)	1	0.9831	1.1127	-108.91	0.9653	35.79
	2	0.9835	1.1153	-110.42	0.9640	36.01
	3	0.9846	1.1015	-98.08	0.9764	33.60
	4	0.9849	1.1098	-106.06	0.9782	32.60
5 (BOD)	1	0.9845	0.9765	22.71	0.9767	33.32
	2	0.9845	0.9802	19.10	0.9760	32.89
	3	0.9858	0.9775	27.75	0.9782	32.62
	4	0.9863	0.9831	16.34	0.9811	31.56
6 (ABOD)	1	0.9936	1.0127	-12.25	0.9949	17.43
	2	0.9956	1.0070	-6.72	0.9973	13.01
	3	0.9956	1.0072	-6.97	0.9973	13.10
	4	0.9956	1.0088	-8.48	0.9973	13.06

As may be expected, the parameter values change if more attributes are added to the model. However, they also change when more zones are added to the spatial system. This is no less disturbing, since it means that either different spatial behavior is present in different parts of the urban region or that the spatial deterrence function of this kind of model is dependent on the trip length distribution in the system, or both.

In the second experiment, not the system size, but the model type was varied. Now all six model types of Figure 1 were applied to the full 30-zone system. Tables 5 and 6 summarize the results of these calibrations.

It can be seen that the different constraints imposed by the different models have an even stronger effect on the parameter values than changes of the zonal system. Now the parameters of some attributes even change their sign. Moreover, the magnitudes of the same parameters differ considerably between model types. Not surprisingly, the goodness-of-fit measures achieved with the six models differ widely. The model type consuming the maximum amount of exogenous information, Model 6 (ABOD), is the most successful in reproducing the observed trip matrix, while the model that uses the least such information, Model 1 (COD), performs worst. This conforms with the findings of Openshaw (1976). As in the first experiment, the most sensitive goodness-of-fit measure seems to be the MAPE statistic, which varies between 13 percent for the two-parameter ABOD model and 58 percent for the one-parameter COD model, or by a ratio of almost 1:5.

The results of both experiments are rather depressing. They say that the precision with which the model parameters are estimated in the calibration procedure is a spurious one. In fact, the estimated parameters are more likely to be an artefact of the accidental combination of zones, attributes, and constraints than a true representation of spatial behavior.

If this is correct, the usefulness of models estimated by these (and similar) techniques would be severely limited. For forecasting purposes, they would only be applicable if no major

changes in the zones, attributes, or constraints occurred in the forecasting period—a very unlikely and uninteresting case. They would be completely useless where the impacts of major changes of the model environment are the object of investigation. For instance, forecasting the impacts of rising gasoline prices on residential location using the four-parameter Model 3 (AOD) would lead to strange results, since in this model car trip costs happen to figure positively. Even the crudest heuristic choice of parameter values guided by common sense (e.g., equal weighting of attributes) would lead to a more plausible forecast!

4.4 Program Performance

Throughout the above experiments, LOGIT proved to be a reliable and efficient program. In all cases, the program reached convergence even from remote starting values. The program never needed to ask for a new start vector.

The number of iterations required for each calibration is comparable to numbers reported by other authors (Batty 1976; van Est and van Setten 1977). Remarkably, the number of iterations was found to be almost independent of the starting values selected or of the number of zones or attributes. However, in the model types 4 through 6 requiring the calculation of balancing factors in the RAS procedure, the number of iterations increases with the number of attributes and with the number of zones. The largest number of iterations was required for the doubly constrained Model 6 (ABOD).

Table 7 summarizes the performance of the program for Model 3 (AOD) and Model 6 (ABOD) with the three zonal systems used in the experiments. In all cases, zeros were entered as starting values. The table shows for each model and each combination of zones and attributes the number of iterations and the computing time. The computing time includes the time for reading the input file and writing the output files and represents processing time in seconds on the IIASA VAX 11/780 computer. The VAX is reported to be about four times slower than the IBM/370-168. The Model 3

results are representative also for Models 1 (COD) and 2 (AO), while the results for Models 4 (BD) and 5 (BOD) lie between the results listed for Models 3 and 6.

At the beginning of the experiments, the program converged much slower in the four-parameter ABOD model, regardless of the number of zones. Equally slow convergence was observed when attribute 4 was entered as the only attribute. A closer inspection of the travel cost matrix of attribute 4 revealed that excessively high parking fees had been assumed for zone 1, the central business district, producing extreme imbalances in the cost matrix. After these distortions had been removed, attribute 4 behaved reasonably.

Table 7. Number of iterations and computing time of Models 3 (AOD) and 6 (ABOD) for different zonal systems and numbers of attributes.

		Model 3 (AOD)		Model 6 (ABOD)	
Number of zones	Number of attributes	Number of iterations	Time[a]	Number of iterations	Time[a]
10	1	4	0.7	6	1.1
	2	4	0.9	7	1.5
	3	5	1.3	7	1.9
	4	5	1.6	24	6.5
20	1	5	1.8	10	6.4
	2	5	2.7	21	14.2
	3	6	4.2	21	17.6
	4	6	5.4	32	33.0
30	1	6	4.1	13	17.3
	2	6	6.4	31	43.8
	3	6	8.6	28	50.9
	4	6	11.5	41	88.6

[a] processing time on the IIASA VAX 11/780 computer in seconds

CONCLUSIONS

In this paper, a reliable and efficient computer program for estimating spatial choice models with multiattribute utilities has been presented. The application of this program for a wide range of spatial systems, attributes, and model types has been demonstrated. However, the experiments also revealed serious problems connected with the stability and interpretability of the parameter estimates in the face of changing model environments and in the presence of interactions among attributes.

Two, perhaps complementary, strategies for further research may be derived from these results. One strategy would go in the direction of further refinement of the models and their calibration. One important issue under this strategy relates to feedbacks that exist between demand variables, such as trip distribution or locational choice, on the one hand, and supply constraints *in the interactions*, such as highway congestion, on the other hand. In the application presented this would mean to feed the estimated model parameters back into the congestion-sensitive transport model used to determine travel times and travel costs. Ideally, the transport model would, for a given set of parameters, simultaneously solve the trip distribution and trip assignment problem under given link capacity constraints yielding equilibrium travel times and costs. Algorithms that do this consistently and efficiently have been available for some time (Evans 1976; Florian and Nguyen 1977). If the equilibrium travel times and costs are used to recalibrate the model with the LOGIT program, a different set of model parameters may result. These then can again be fed into the transport model, and so on until the process converges, i.e., true equilibrium model parameters *and* travel times and costs are derived. Such a procedure has been suggested, for instance, by Boyce et al. (1981).

However, it remains to be seen if such a complicated procedure would indeed substantially improve the explanatory power of this kind of models. The results of the two experiments presented in the preceding section suggest a rather cautious

view on this matter. Therefore, a second research strategy may also be pursued that, instead of getting the maximum fit out of the data as they are, seeks to improve the model by respecifying the data. This would include experimentation with less rigorous, but more behaviorally oriented methods for attribute selection, transformation, and aggregation making use of judgment, plausibility considerations, and sensitivity analysis. It can be shown that calibration results achieved with such "softer" methods can be comparable or even better than results derived from rigorous statistical estimation (cf. Wegener 1981). It is hoped that by combining soft calibration methods with efficient statistical techniques like the one presented in this paper, more meaningful and consistent models, which also make better predictions, can be developed.

REFERENCES

Anas, A. (1975) Empirical Calibration and Testing of a Simulation Model of Residential Location. *Environment and Planning A* 7:899-920.

Anas, A. (1981) *Discrete Choice Theory, Information Theory, and the Multinomial Logit and Gravity Models.* Paper presented at the North American Meetings of the Regional Science Association, Montreal.

Bard, Y. (1974) *Nonlinear Parameter Estimation.* New York: Academic Press.

Batty, M. (1976) *Urban Modelling: Algorithms, Calibration, Predictions.* Cambridge: Cambridge University Press.

Batty, M., and S. Mackie (1972) The Calibration of Gravity, Entropy, and Related Models of Spatial Interaction, *Environment and Planning* 4:205-233.

Boyce, D.E., L.J. LeBlanc, K.S. Chon, Y.L. Lee, and K.T. Lin (1981) *Combined Models of Location, Destination, Mode and Route Choice: Implementation Issues Related to a Generalized Algorithm.* Publication No. 4. Urbana, Illinois: Department of Civil Engineering, University of Illinois at Urbana-Champaign.

Brotchie, J.F., J.W. Dickey, and R. Sharpe (1980) *TOPAZ - General Planning Technique and its Applications at the Regional, Urban, and Facility Planning Levels.* Berlin/Heidelberg/New York: Springer Verlag.

Churchhouse, R.F., ed. (1981) *Handbook of Applicable Mathematics, Vol. III: Numerical Methods.* New York: Wiley.

Domencich, F.A., and D. McFadden (1975) *Urban Travel Demand.* Amsterdam: North Holland.

Evans, A.W. (1971) The Calibration of Trip Distribution Models with Exponential or Similar Cost Functions. *Transportation Research* 5:15-38.

Evans, S.P. (1976) Derivation and Analysis of Some Models for Combining Trip Distribution and Assignment. *Transportation Research* 10:37-57.

Florian, M., and S. Nguyen (1977) A Combined Trip Distribution, Model Split and Trip Assignment Model. *Transportation Research* 12:241-246.

Foot, D. (1981) *Operational Urban Models.* London: Methuen.

Hyman, G.M. (1969) The Calibration of Trip Distribution Models. *Environment and Planning* 1:105-112.

Lakshmanan, T.R., and W.G. Hansen (1965) A Retail Market Potential Model. *Journal of the American Institute of Planners* 31: 134-143.

Luce, R.D. (1959) *Individual Choice Behavior.* New York: Wiley.

Manski, C.F., and D. McFadden, eds. (1981) *Structural Analysis of Discrete Data with Econometric Applications.* Cambridge, Mass.: MIT Press.

McFadden, D. (1973) Conditional Logit Analysis of Qualitative Choice Behavior. Pages 105-142 in, *Frontiers in Econometrics*, edited by P. Zarembka. New York: Academic Press.

Openshaw, S. (1976) An Empirical Study of Some Spatial Interaction Models. *Environment and Planning A* 8:23-41.

Openshaw, S. (1979) Alternative Methods of Estimating Spatial Interactions Models: An Empirical Study of Performance in Short Term Forecasting. Pages 201-225 in, *Exploratory and Explanatory Statistical Analysis of Spatial Data*, edited by C.P.A. Bartels and R.H. Ketellaper. Leiden: M. Nijhoff.

Putman, S.H. (1980) Calibrating Urban Residential Models 3: Empirical Results for Non-US Cities. *Environment and Planning A* 12:813-827.

Putman, S.H., and F. Ducca (1978a) Calibrating Urban Residential Models 1: Procedures and Strategies. *Environment and Planning A* 10:633-650.

Putman, S.H., and F. Ducca (1978b) Calibrating Urban Residential Models 2: Empirical Results. *Environment and Planning A* 10:1001-1014.

Smith, D.P., and B.G. Hutchinson (1979) *Goodness of Fit Statistics for Trip Distribution Models.* Working Paper. Waterloo, Ontario: Department of Civil Engineering, University of Waterloo.

Snickars, F., and J.W. Weibull (1977) A Minimum Information Principle: Theory and Practice. *Regional Science and Urban Economics* 7:137-168.

Schwetlick, H. (1979) *Numerische Lösung nichtlinearer Gleichungen* (Numerical Solution of Nonlinear Equations). München/Wien: Oldenbourg.

Stone, R., ed. (1963) *Input-Output Relations 1954-1966.* London: Chapman and Hall.

Stopher, R.R., and A.H. Meyburg (1979) *Survey Sampling and Multivariate Analysis for Social Scientists and Engineers.* Lexington, Mass.: Lexington Books.

Tanner, J.C. (1961) *Factors Affecting the Amount of Travel.* Road Research Laboratory Technical Paper No. 51. London: H.M.S.O.

van Est, J., and J. van Setten (1977) *Calibration Methods: Application for some Spatial Distribution Models.* Working Paper No. 6. Delft: Research Center for Physical Planning TNO.

Van Est, J., and J. van Setten (1978) Two Estimation Methods for Singly Constrained Spatial Distribution Models. Pages 2-16 in, *Exploratory and Explanatory Statistical Analysis of Spatial Data,* edited by G.P.A. Bartels and R.H. Ketellaper. Leiden: M. Nijhoff.

van Lierop, W., and P. Nijkamp (1981) *Disaggregate Models of Choice in a Spatial Context.* Research Memorandum 1981-4. Amsterdam: Department of Economics, Free University.

Wegener, M. (1981) *A Simulation Study of Movement in the Dortmund Housing Market.* Paper presented at the International Seminar on Migration and Small Area Population Forecasting, Rotterdam (forthcoming in *Tijdschrift voor Economische en Sociale Geografie*).

Wegener, M. (1982) *Aspects of Urban Decline: Experiments with A Multilevel Economic-Demographic Model for the Dortmund Region.* WP-82-17. Laxenburg, Austria: International Institute for Applied Systems Analysis.

Willekens, F. (1980) Entropy, Multiproportional Adjustment and Analysis of Contingency Tables. *Sistemi Urbani* 2/3: 171-201.

Willekens, F., A. Pór, and R. Raquillet (1979) *Entropy, Multiproportional, and Quadratic Techniques for Inferring Detailed Migration Patterns from Aggregate Data. Mathematical Theories, Algorithms, Applications, and Computer Programs.* WP-79-88. Laxenburg, Austria: International Institute for Applied Systems Analysis.

Williams, H.C.W.L. (1977) On the Formation of Travel Demand Models and Economic Evaluation Measures of User Benefit. *Environment and Planning A* 9:285-344.

Wilson, A.G. (1970) *Entropy in Urban and Regional Modelling.* London: Pion.

Wilson, A.G. (1974) *Urban and Regional Models in Geography and Planning.* London: Wiley.

APPENDIX: Fortran Listing of the LOGIT Program

```
c     program logit
c
c     estimation of the parameters
c     of a multinomial logit model
c     with marginal constraints
c     and multiattribute utilities
c
c     authors: michael wegener
c              friedrich graef
c     date:    14 july 1982
c
      dimension mv(8),bf(8)
c
      character*1 ma,no
c
      data no /'n'/
c
      call idat(nt,na,nx,mv,bf)
      mm = 0
   10 mm = mm+1
      call mdat(mm,mc,mt,nt,na,nx,mx,mv)
      call ecal(mm,mc,mt,nt,na,mx,mv,bf)
      write (6,6000)
      read (5,5000) ma
      if (ma.ne.no) goto 10
      stop
c
 5000 format (a1)
 6000 format (' another try?')
      end


      subroutine idat(nt,na,nx,mv,bf)
c
c     specifying the input data
c
      dimension mv(8),bf(8)
c
      write (6,6000)
      write (6,6010)
      read (5,*) nt
      write (6,6020)
      read (5,*) na
      write (6,6030)
      read (5,*) nxj
      write (6,6040)
      read (5,*) nxij
      call iput(nt,na,nxj,nxij,mv)
      nx = nxj+nxij
      call norm(nt,na,nx,bf)
      return
c
 6000 format (' specify input data:')
 6010 format (' enter no. of origins (decisionmaker groups)')
 6020 format (' enter no. of destinations (alternatives)')
 6030 format (' enter no. of attributes of alternatives')
 6040 format (' enter no. of trip (distance) attributes')
      end
```

```
      subroutine iput(nt,na,nxj,nxij,mv)
c
c     reading the input data
c
      common a(30),b(30),o(30),d(30),
     *oo(30),do(30),to(30,30),x(30,30,8)
c
      dimension mv(8),xj(8)
c
      read (8,8000) (oo(i),i=1,nt)
      read (8,8000) (do(j),j=1,na)
      nk = (na-1)/10+1
      do 100 ik=1,nk
      ja = (ik-1)*10+1
      je = min0(ja+9,na)
      do 100 i=1,nt
      read (8,8000) (to(i,j),j=ja,je)
  100 continue
      if (nxj.eq.0) goto 10
      nk = (nxj-1)/10+1
      do 200 ik=1,nk
      ka = (ik-1)*10+1
      ke  = min0(ka+9,nxj)
      do 200 j=1,na
      read (8,8000) (xj(k),k=ka,ke)
      do 200 i=1,nt
      do 200 k=ka,ke
      x(i,j,k) = xj(k)
  200 continue
      do 300 k=1,nxj
      mv(k) = k
  300 continue
   10 if (nxij.eq.0) return
      nk = (na-1)/10+1
      do 400 k=1,nxij
      mv(nxj+k) = nxj+k
      do 400 ik=1,nk
      ja = (ik-1)*10+1
      je = min0(ja+9,na)
      do 400 i=1,nt
      read (8,8000) (x(i,j,nxj+k),j=ja,je)
  400 continue
      return
c
 8000 format (10x,10f6.0)
      end


      subroutine norm(nt,na,nx,bf)
c
c     normalizing all attributes
c
      common a(30),b(30),o(30),d(30),
     *oo(30),do(30),to(30,30),x(30,30,8)
c
      dimension bf(8)
c
      do 200 k=1,nx
      xmax = -1.e30
      xmin = 1.e30
      do 210 i=1,nt
      do 210 j=1,na
      xmax = amax1(xmax,x(i,j,k))
      xmin = amin1(xmin,x(i,j,k))
  210 continue
      ama = xmin+(xmax-xmin)*0.5
      ilog = alog10(ama)+1.
      bf(k) = 1./10**ilog
      do 220 i=1,nt
      do 220 j=1,na
      x(i,j,k) = x(i,j,k)*bf(k)
  220 continue
  200 continue
      return
      end
```

```
      subroutine mdat(mm,mc,mt,nt,na,nx,mx,mv)
c
c     specifying the model data
c
      dimension mv(8)
c
      character*1 ma,no
c
      data no /'n'/
c
      write (6,6000)
      write (6,6010)
      read (5,*) mt
      mo = 3
      md = 3
      if (mt.ne.4) write (6,6020)
      if (mt.ne.4) read (5,*) mo
      if (mt.ne.2) write (6,6030)
      if (mt.ne.2) read (5,*) md
      call init(mt,mo,md,nt,na)
      if (mm.gt.1) goto 10
      mc = 0
      mx = nx
      write (6,6040)
      read (5,5000) ma
      if (ma.eq.no) return
      goto 20
   10 mc = 1
      write (6,6050)
      read (5,5000) ma
      if (ma.ne.no) return
   20 mc = 0
      write (6,6060)
      read (5,*) mx
      write (6,6070) mx
      read (5,*) (mv(k),k=1,mx)
      return
c
 5000 format (a1)
 6000 format (' specify model data:')
 6010 format (' select model type:'/
     *        '  1  unconstrained (cod)'/
     *        '  2  production-constrained (ao)'/
     *        '  3  production-constrained (aod)'/
     *        '  4  attraction-constrained (bd)'/
     *        '  5  attraction-constrained (bod)'/
     *        '  6  doubly constrained (abod)')
 6020 format (' specify origins:'/
     *        '  1  take from input origins'/
     *        '  2  aggregate from observed trips (choices)')
 6030 format (' specify destinations:'/
     *        '  1  take from input destinations'/
     *        '  2  aggregate from observed trips (choices)')
 6040 format (' attribute selection?')
 6050 format (' same attribute(s) as before?')
 6060 format (' how many attributes?')
 6070 format (' enter',i3,' attribute number(s)')
      end
```

```
      subroutine init(mt,mo,md,nt,na)
c
c     initializing origins, destinations,
c     and balancing factors
c
      common a(30),b(30),o(30),d(30),
     *oo(30),do(30),to(30,30)
c
      do 100 i=1,nt
      a(i) = 1.
      o(i) = oo(i)
      goto (100,102,103),mo
  102 o(i) = 0.
      do 110 j=1,na
      o(i) = o(i)+to(i,j)
  110 continue
      goto 100
  103 o(i) = 1.
  100 continue
      do 200 j=1,na
      b(j) = 1.
      d(j) = do(j)
      goto (200,202,203),md
  202 d(j) = 0.
      do 210 i=1,nt
      d(j) = d(j)+to(i,j)
  210 continue
      goto 200
  203 d(j) = 1.
  200 continue
      if (mt.ne.6) return
      so = 0.
      do 300 i=1,nt
      so = so+o(i)
  300 continue
      sd = 0.
      do 400 j=1,na
      sd = sd+d(j)
  400 continue
      cf = sd/so
      cfp = (cf-1.)*100.
      if (abs(cfp).lt.1.) return
      do 500 i=1,nt
      o(i) = o(i)*cf
  500 continue
      write (6,6060) cfp
      return
c
 6060 format (' warning: origins adjusted by',f8.1,' percent'/)
      end
```

```
      subroutine ecal(mm,mc,mt,nt,na,mx,mv,bf)
c
c     execution of the calibration
c
      dimension mv(8),bf(8),bt(8)
c
      real*8 bt
c
      character*1 ma,no
c
      data no /'n'/
c
   10 mb = 0
      mp = 0
      mq = 0
      if (mc.eq.0) goto 11
      write (6,6000)
      read (5,5000) ma
      if (ma.ne.no) goto 12
   11 write (6,6010) mx
      read (5,*) (bt(k),k=1,mx)
   12 write (6,6020)
      read (5,*) ms
      mq = ms+2
   20 call mlml(mb,mq,nr,mt,nt,na,mx,mv,bt)
      if (mb.eq.1) goto 10
      call beta(6,mt,mx,mv,bf,bt)
      if (mq.eq.4) goto 50
      write (6,6030)
      read (5,*) mp
      goto (10,20,30,40,60),mp
   30 call trip(nr,nt,na,mx,mv,bt)
      if (mt.gt.1) call rast(0,nr,mt,nt,na)
      call mean(6,nt,na,mx,mv,bf)
      call stat(6,nt,na)
      goto 50
   40 mq = 4
      goto 20
   50 write (6,6040)
      read (5,*) mq
      goto (10,20,30,20,60),mq
   60 write (6,6050)
      read (5,5000) ma
      if (ma.eq.no) return
      call beta(7,mt,mx,mv,bf,bt)
      call trip(nr,nt,na,mx,mv,bt)
      if (mt.gt.1) call rast(0,nr,mt,nt,na)
      call mean(7,nt,na,mx,mv,bf)
      call stat(7,nt,na)
      call ttab(9,nt,na)
      return
c
 5000 format (a1)
 6000 format (' continue with present beta(s)?')
 6010 format (' enter start vector beta:',i3,' number(s)')
 6020 format (' select one:'/
     *        '  1  stop after intermediate steps'/
     *        '  2  continue until final solution')
 6030 format (' select one:'/
     *        '  1  new start vector beta'/
     *        '  2  repeat with present beta(s)'/
     *        '  3  statistics'/
     *        '  4  continue until final solution'/
     *        '  5  exit')
 6040 format (' select one:'/
     *        '  1  new start vector beta'/
     *        '  2  repeat with present balancing factor(s)'/
     *        '  3  statistics'/
     *        '  4  continue until final solution'/
     *        '  5  exit')
 6050 format (' output?')
      end
```

```
      subroutine mlml(mb,mq,nr,mt,nt,na,mx,mv,bt)
c
c     maximizing the loglikelihood
c     of the multinomial logit model
c     either with gradient search
c     or newton-raphson iteration
c
      common a(30),b(30),o(30),d(30),
     *oo(30),do(30),to(30,30),x(30,30,8),
     *p(30,30)
c
      dimension mv(8),bt(8),g(8),h(8,8)
c
      real*8 p,bt,g,h,eps
c
      character*4 beta
c
      data beta,eps,nit /'beta',.00005,200/
c
      it = 0
      mn = 0        <- nr = 0
      cn = 9999.
   20 it = it+1
      if (mq.ne.4.or.mt.lt.4) goto 10
      call trip(nr,nt,na,mx,mv,bt)
      call rast(0,nr,mt,nt,na)
   10 goto (12,11,11,12,12,11),mt
   11 call prob1(nt,na,mx,mv,bt)
      call derv1(nt,na,mx,mv,g,h)
      goto 13
   12 call prob2(nt,na,mx,mv,bt)
      call derv2(nt,na,mx,mv,g,h)
   13 if (mn.eq.0) call grad(mx,g,h)
      if (mn.gt.0) call newt(mx,g,h)
      ca = cn
      if (mn.ne.1) mn = 2
      cn = 0.
      do 100 k=1,mx
      cn = cn+dabs(g(k))
  100 continue
      if (cn.gt.ca*2.) mn = mn-1
      if (mn.eq.0) goto 10
      do 200 k=1,mx
      bt(k) = bt(k)-g(k)
      if (dabs(bt(k)).gt.100.) goto 95
  200 continue
      if (it.eq.1) write (6,6000) (beta,mv(k),k=1,mx)
      write (6,6010) it,nr,cn,(bt(k),k=1,mx)
      if (it.eq.nit) goto 99
      if (nr.gt.1.or.cn.gt.eps*mx) goto 20
      write (6,6020) it
      write (6,6030)
      return
   95 write (6,6040)
      mb = 1
      return
   99 write (6,6020) nit
      write (6,6050)
      return
c
 6000 format ('  it ras  change',8(2x,a4,i2))
 6010 format (' ',i3,i4,9f8.4)
 6020 format (' gradient search terminated at iteration',i4)
 6030 format (' parameter changes less than specified limit')
 6040 format (' bad start vector: try other beta(s)')
 6050 format (' maximum number of iterations exceeded')
      end
```

```
      subroutine prob1(nt,na,mx,mv,bt)
c
c     calculating choice probabilities
c     summing up to one over each row
c
      common a(30),b(30),o(30),d(30),
     *oo(30),do(30),to(30,30),x(30,30,8),
     *p(30,30)
c
      dimension mv(8),bt(8)
c
      real*8 p,bt,xv,xw
c
      do 100 i=1,nt
      do 100 j=1,na
      xw = 0.
      do 110 k=1,mx
      xw = xw+bt(k)*x(i,j,mv(k))
  110 continue
      bd = b(j)*d(j)
      p(i,j) = bd*dexp(xw)
  100 continue
      do 200 i=1,nt
      xv = 0.
      do 210 j=1,na
      xv = xv+p(i,j)
  210 continue
      do 220 j=1,na
      p(i,j) = p(i,j)/xv
  220 continue
  200 continue
      return
      end



      subroutine derv1(nt,na,mx,mv,g,h)
c
c     calculating the gradient vector
c     and the hessian matrix for prob1
c
      common a(30),b(30),o(30),d(30),
     *oo(30),do(30),to(30,30),x(30,30,8),
     *p(30,30)
c
      dimension mv(8),g(8),h(8,8),sto(30),xm(8),xs(8,8)
c
      real*8 p,g,h,xm,xs
c
      do 100 k=1,mx
      g(k) = 0.
      do 100 l=1,mx
      h(k,l) = 0.
  100 continue
      do 200 i=1,nt
      sto(i) = 0.
      do 200 j=1,na
      sto(i) = sto(i)+to(i,j)
  200 continue
      do 300 i=1,nt
      do 310 k=1,mx
      xm(k) = 0.
      do 311 l=1,mx
      xs(k,l) = 0.
  311 continue
      do 312 j=1,na
      g(k) = g(k)+to(i,j)*x(i,j,mv(k))
      xm(k) = xm(k)+p(i,j)*x(i,j,mv(k))
      do 312 l=1,mx
      xs(k,l) = xs(k,l)+p(i,j)*x(i,j,mv(k))*x(i,j,mv(l))
  312 continue
      g(k) = g(k)-sto(i)*xm(k)
  310 continue
      do 320 k=1,mx
      do 320 l=1,mx
      h(k,l) = h(k,l)-sto(i)*(xs(k,l)-xm(k)*xm(l))
  320 continue
  300 continue
      return
      end
```

```
      subroutine prob2(nt,na,mx,mv,bt)
c
c     calculating choice probabilities
c     summing up to one over all rows
c
      common a(30),b(30),o(30),d(30),
     *oo(30),do(30),to(30,30),x(30,30,8),
     *p(30,30)
c
      dimension mv(8),bt(8)
c
      real*8 p,bt,xv,xw
c
      do 100 i=1,nt
      do 100 j=1,na
      xw = 0.
      do 110 k=1,mx
      xw = xw+bt(k)*x(i,j,mv(k))
  110 continue
      abod = a(i)*b(j)*o(i)*d(j)
      p(i,j) = abod*dexp(xw)
  100 continue
      xv = 0.
      do 200 i=1,nt
      do 200 j=1,na
      xv = xv+p(i,j)
  200 continue
      do 300 i=1,nt
      do 300 j=1,na
      p(i,j) = p(i,j)/xv
  300 continue
      return
      end


      subroutine derv2(nt,na,mx,mv,g,h)
c
c     calculating the gradient vector
c     and the hessian matrix for prob2
c
      common a(30),b(30),o(30),d(30),
     *oo(30),do(30),to(30,30),x(30,30,8),
     *p(30,30)
c
      dimension mv(8),g(8),h(8,8),xm(8),xs(8,8)
c
      real*8 p,g,h,xm,xs
c
      do 100 k=1,mx
      xm(k) = 0.
      g(k) = 0.
      do 100 l=1,mx
      xs(k,l) = 0.
      h(k,l) = 0.
  100 continue
      sto = 0.
      do 200 i=1,nt
      do 200 j=1,na
      sto = sto+to(i,j)
      do 200 k=1,mx
      xm(k) = xm(k)+p(i,j)*x(i,j,mv(k))
      do 200 l=1,mx
      xs(k,l) = xs(k,l)+p(i,j)*x(i,j,mv(k))*x(i,j,mv(l))
  200 continue
      do 300 i=1,nt
      do 300 j=1,na
      do 300 k=1,mx
      g(k) = g(k)+to(i,j)*x(i,j,mv(k))
  300 continue
      do 400 k=1,mx
      g(k) = g(k)-sto*xm(k)
      do 400 l=1,mx
      h(k,l) = h(k,l)-sto*(xs(k,l)-xm(k)*xm(l))
  400 continue
      return
      end
```

```
      subroutine grad(mx,g,h)
c
c     gradient search method:
c     search along the steepest ascent
c
      dimension g(8),h(8,8)
c
      real*8 g,h,gsq,hsq,xv,qt
c
      gsq = 0.
      do 100 k=1,mx
      gsq = gsq+g(k)*g(k)
  100 continue
      hsq = 0.
      do 200 k=1,mx
      xv = 0.
      do 210 l=1,mx
      xv = xv+h(k,l)*g(l)
  210 continue
      hsq = hsq+g(k)*xv
  200 continue
      hsq = dabs(hsq)
      qt = gsq/hsq
      do 300 k=1,mx
      g(k) = -qt*g(k)
  300 continue
      return
      end


      subroutine newt(mx,g,h)
c
c     newton-raphson method:
c     solving mx linear equations
c     using the gauss-jordan method
c
      dimension g(8),h(8,8)
c
      real*8 g,h,t
c
      do 100 k=1,mx
      k1 = k+1
      if (k.eq.mx) goto 101
      l = k
      do 110 j=k1,mx
      if (dabs(h(j,k)).gt.dabs(h(l,k))) l = j
  110 continue
      if (l.eq.k) goto 101
      do 120 j=k,mx
      t = h(k,j)
      h(k,j) = h(l,j)
      h(l,j) = t
  120 continue
      t = g(k)
      g(k) = g(l)
      g(l) = t
  101 if (h(k,k).eq.0.) h(k,k) = 1.e-10
      g(k) = g(k)/h(k,k)
      if (k.eq.mx) goto 100
      do 130 j=k1,mx
      h(k,j) = h(k,j)/h(k,k)
  130 continue
      do 140 i=k1,mx
      do 141 j=k1,mx
      h(i,j) = h(i,j)-h(i,k)*h(k,j)
  141 continue
      g(i) = g(i)-h(i,k)*g(k)
  140 continue
  100 continue
      if (mx.eq.1) return
      do 200 j=2,mx
      k = mx+2-j
      k1 = k-1
      do 200 i=1,k1
      g(i) = g(i)-h(i,k)*g(k)
  200 continue
      return
      end
```

```
      subroutine beta(nf,mt,mx,mv,bf,bt)
c
c     output of beta vector
c
      dimension mv(8),bt(8),bf(8)
c
      real*8 bt
c
      if (mt.eq.1) write (nf,9000)
      if (mt.eq.2) write (nf,9010)
      if (mt.eq.3) write (nf,9020)
      if (mt.eq.4) write (nf,9030)
      if (mt.eq.5) write (nf,9040)
      if (mt.eq.6) write (nf,9050)
      do 100 k=1,mx
      write (nf,6000) mv(k),bf(mv(k)),bt(k)
  100 continue
      return
c
 6000 format (' beta',i2,' (',1pe6.0,')',9(' .'),0pf10.4)
 9000 format (/' unconstrained model (cod):'/)
 9010 format (/' production-constrained model (ao):'/)
 9020 format (/' production-constrained model (aod):'/)
 9030 format (/' attraction-constrained model (bd):'/)
 9040 format (/' attraction-constrained model (bod):'/)
 9050 format (/' doubly constrained model (abod):'/)
      end


      subroutine trip(nr,nt,na,mx,mv,bt)
c
c     generation of trip table (choice matrix)
c
      common a(30),b(30),o(30),d(30),
     *oo(30),do(30),to(30,30),x(30,30,8),
     *p(30,30),t(30,30)
c
      dimension mv(8),bt(8)
c
      real*8 p,bt
c
      nr = 0
      st = 0.
      so = 0.
      sd = 0.
      do 100 i=1,nt
      so = so+o(i)
      sd = sd+d(i)
      do 100 j=1,na
      xw = 0.
      do 110 k=1,mx
      xw = xw+bt(k)*x(i,j,mv(k))
  110 continue
      t(i,j) = a(i)*b(j)*o(i)*d(j)*exp(xw)
      st = st+t(i,j)
  100 continue
      sto = amax1(so,sd)
      cf = sto/st
      do 200 i=1,nt
      do 200 j=1,na
      t(i,j) = t(i,j)*cf
  200 continue
      return
      end
```

```
      subroutine rast(mr,nr,mt,nt,na)
c
c     biproportional (uniproportional) adjustment
c     of the trip table (choice matrix) using the
c     ras-technique and calculation of balancing
c     factors
c
      common a(30),b(30),o(30),d(30),
     *oo(30),do(30),to(30,30),x(30,30,8),
     *p(30,30),t(30,30)
c
      dimension so(30),sd(30)
c
      real*8 p
c
      data eps,nit /0.0001,200/
c
      do 100 it=1,nit
      se = 0.
      if (mt.eq.4.or.mt.eq.5) goto 101
      do 200 i=1,nt
      so(i) = 0.
      do 210 j=1,na
      if (t(i,j).lt.0.0001) t(i,j) = 0.0001
      so(i) = so(i)+t(i,j)
  210 continue
      se = se+abs(o(i)-so(i))
      so(i) = o(i)/so(i)
      a(i) = a(i)*so(i)
  200 continue
      do 300 i=1,nt
      do 300 j=1,na
      t(i,j) = t(i,j)*so(i)
  300 continue
      if (mt.lt.4) goto 10
  101 sod = 0.
      do 400 j=1,na
      sod = sod+d(j)
      sd(j) = 0.
      do 410 i=1,nt
      if (t(i,j).eq.0.) t(i,j) = 0.0001
      sd(j) = sd(j)+t(i,j)
  410 continue
      se = se+abs(d(j)-sd(j))
      sd(j) = d(j)/sd(j)
      b(j) = b(j)*sd(j)
  400 continue
      do 500 j=1,na
      do 500 i=1,nt
      t(i,j) = t(i,j)*sd(j)
  500 continue
      if (mt.eq.4.or.mt.eq.5) goto 10
      if (se.lt.eps*it*sod) goto 10
  100 continue
      it = nit
   10 if (mr.gt.0) write (6,6000) it
      nr = it
      return
c
 6000 format (' no. of iterations in ras:',i4)
      end
```

```
      subroutine stat(nf,nt,na)
c
c     goodness-of-fit statistics
c
      common a(30),b(30),o(30),d(30),
     *oo(30),do(30),to(30,30),xx(30,30,8),
     *p(30,30),t(30,30)
c
      real*8 p
c
      sx = 0.
      sy = 0.
      sxy = 0.
      sxx = 0.
      syy = 0.
      sax = 0.
      rlx = 0.
      rly = 0.
      af = 0.
      nb = nt*na
      do 100 i=1,nt
      do 100 j=1,na
      x = to(i,j)
      if (x.eq.0.) x = 0.0001
      y = t(i,j)
      if (y.eq.0.) y = 0.0001
      sx = sx+x
      sy = sy+y
      sxy = sxy+x*y
      sxx = sxx+x*x
      syy = syy+y*y
      sax = sax+abs(x)
      rlx = rlx+x*alog(x)
      rly = rly+x*alog(y)
      af = af+abs(x-y)
  100 continue
      xm = sx/nb
      ym = sy/nb
      vxy = (sxy-xm*sy)/(nb-1)
      vx = (sxx-xm*sx)/(nb-1)
      vy = (syy-ym*sy)/(nb-1)
      rl = rly/rlx
      af = af/sax*100.
      rc = vxy/vy
      cn = xm-rc*ym
      cc = vxy/sqrt(vx*vy)
      rr = cc*cc
      tt = cc*sqrt(nb-2.)/sqrt(1.-rr)
      write (nf,6000) rl,rc,cn,cc,rr,tt,af
      return
c
 6000 format (/' statistics:'/
     *         ' loglikelihood ratio . . . . . .',f10.4/
     *         ' slope . . . . . .',f10.4/
     *         ' intercept . . . . . .',f10.2/
     *         ' correlation coefficient r . . . .',f10.4/
     *         ' r-squared . . . . . .',f10.4/
     *         ' t of r-squared  . . . . . .',f10.2/
     *         ' mean absolute percentage error  .',f10.2/)
      end
```

```
      subroutine mean(nf,nt,na,mx,mv,bf)
c
c     comparison of observed and
c     predicted totals and means
c
      common a(30),b(30),o(30),d(30),
     *oo(30),do(30),to(30,30),x(30,30,8),
     *p(30,30),t(30,30)
c
      dimension mv(8),bf(8),sx(8),sy(8)
c
      real*8 p
c
      sto = 0.
      st = 0.
      nb = nt*na
      do 100 i=1,nt
      do 100 j=1,na
      sto = sto+to(i,j)
      st = st+t(i,j)
  100 continue
      do 200 k=1,mx
      sx(k) = 0.
      sy(k) = 0.
      do 210 i=1,nt
      do 210 j=1,na
      sx(k) = sx(k)+to(i,j)*x(i,j,mv(k))
      sy(k) = sy(k)+t(i,j)*x(i,j,mv(k))
  210 continue
      sx(k) = sx(k)/(sto*bf(mv(k)))
      sy(k) = sy(k)/(st*bf(mv(k)))
  200 continue
      isto = sto+0.5
      ist = st+0.5
      sto = sto/nb
      st = st/nb
      write (nf,6000) nb,nb,isto,ist,sto,st
      do 300 k=1,mx
      write (nf,6010) mv(k),sx(k),sy(k)
  300 continue
      return
c
 6000 format (/' totals and means:',8x,'observed predicted'/
     *         ' observations  . . . . .',2i10/
     *         ' trips (choices) . . . .',2i10/
     *         ' trips/observation . . .',2f10.2)
 6010 format ( ' mean of attribute',i2,' . .',2f10.3)
      end



      subroutine ttab(nf,nt,na)
c
c     output of trip table (choice matrix)
c
      common a(30),b(30),o(30),d(30),
     *oo(30),do(30),to(30,30),x(30,30,8),
     *p(30,30),t(30,30)
c
      dimension it(30)
c
      real*8 p
c
      nk = (na-1)/10+1
      do 100 ik=1,nk
      ja = (ik-1)*10+1
      je = min0(ja+9,na)
      do 100 i=1,nt
      do 110 j=ja,je
      it(j) = t(i,j)+0.5
  110 continue
      write (nf,9000) ik,i,(it(j),j=ja,je)
  100 continue
      return
c
 9000 format ('t  ',i1,i4,2x,10i6)
      end
```

Test Dataset

oo	1		1								
do	1		1								
to	1	1	1	15490	2900	2957	285	504	471	231	634	375	454
to	1	1	2	6442	14601	3203	647	228	627	292	590	336	238
to	1	1	3	8552	3284	10780	241	153	996	385	880	341	204
to	1	1	4	3000	3133	1199	7183	225	260	106	245	124	74
to	1	1	5	3333	4210	1917	307	5404	1063	357	493	153	100
to	1	1	6	4827	3125	3188	269	536	6721	733	901	221	138
to	1	1	7	3275	1474	1807	144	102	518	6275	2997	264	107
to	1	1	8	3465	1408	2105	171	53	313	736	9249	467	120
to	1	1	9	5173	2372	1947	232	87	298	226	1390	5452	237
to	1	1	10	3979	1397	1088	152	371	221	118	315	260	7098
tt	1	1	1	13.0	26.5	24.6	44.1	47.4	34.7	48.7	37.5	40.9	45.6
tt	1	1	2	26.5	9.0	29.8	38.5	41.9	41.6	57.2	42.8	47.3	58.8
tt	1	1	3	24.6	31.4	15.0	51.2	53.4	29.1	52.6	37.1	50.2	45.6
tt	1	1	4	44.1	38.6	51.2	18.5	55.8	65.8	84.6	64.2	76.2	72.8
tt	1	1	5	47.4	41.9	51.8	57.7	25.0	45.5	55.4	59.5	69.3	81.5
tt	1	1	6	34.6	42.0	29.1	67.2	45.4	23.0	44.7	42.4	61.7	59.6
tt	1	1	7	48.7	57.2	52.7	80.3	55.4	44.7	22.0	40.7	65.9	82.6
tt	1	1	8	37.6	43.2	37.1	64.1	59.4	42.4	40.7	18.0	49.9	66.6
tt	1	1	9	40.9	46.9	50.5	72.0	68.9	60.6	65.9	49.9	23.0	52.4
tt	1	1	10	45.6	58.8	45.6	73.7	79.5	59.6	82.6	66.6	52.4	20.0
tt	2	1	1	13.2	17.7	17.9	21.2	23.9	28.2	25.5	24.2	21.1	22.4
tt	2	1	2	16.4	10.5	18.1	16.3	19.6	28.0	25.8	25.1	23.2	24.5
tt	2	1	3	17.7	18.9	13.7	21.1	22.5	22.8	20.8	22.3	21.3	24.1
tt	2	1	4	23.4	18.8	24.2	15.5	19.5	31.8	31.6	31.2	30.0	31.1
tt	2	1	5	27.0	22.1	26.5	19.5	22.0	20.0	25.8	34.8	33.6	36.2
tt	2	1	6	32.7	32.8	26.4	31.8	20.0	20.0	21.6	30.8	34.2	37.6
tt	2	1	7	28.0	29.5	22.9	31.6	25.9	21.9	20.0	20.9	26.3	32.8
tt	2	1	8	27.9	29.4	24.2	32.0	32.9	28.9	19.7	16.0	23.0	31.6
tt	2	1	9	23.9	26.9	24.6	30.7	32.9	32.9	27.0	23.4	20.0	21.5
tt	2	1	10	24.7	27.7	26.3	31.5	34.9	34.1	31.4	30.1	21.6	17.0
tc	1	1	1	0.36	0.36	0.36	0.40	0.57	0.38	0.96	0.67	0.37	0.49
tc	1	1	2	0.36	0.36	0.36	0.36	0.48	0.43	1.00	0.43	0.47	0.71
tc	1	1	3	0.36	0.36	0.36	0.52	0.61	0.36	0.67	0.36	0.40	0.57
tc	1	1	4	0.40	0.36	0.52	0.36	0.53	0.79	1.27	0.77	0.99	0.84
tc	1	1	5	0.57	0.48	0.61	0.53	0.36	0.49	0.58	0.84	0.90	1.04
tc	1	1	6	0.38	0.43	0.36	0.79	0.49	0.36	0.36	0.45	0.71	1.02
tc	1	1	7	0.96	1.00	0.67	1.27	0.58	0.36	0.36	0.39	0.82	1.26
tc	1	1	8	0.67	0.43	0.36	0.77	0.84	0.45	0.39	0.36	0.56	1.00
tc	1	1	9	0.37	0.47	0.46	0.78	0.90	0.71	0.82	0.56	0.36	0.58
tc	1	1	10	0.49	0.71	0.57	0.84	1.04	1.02	1.26	1.00	0.58	0.36
tc	2	1	1	0.17	0.21	0.28	0.45	0.54	1.08	0.70	0.70	0.49	0.75
tc	2	1	2	0.21	0.08	0.36	0.23	0.33	1.05	0.71	0.78	0.66	0.92
tc	2	1	3	0.24	0.33	0.21	0.50	0.53	0.76	0.44	0.64	0.60	0.89
tc	2	1	4	0.45	0.23	0.56	0.35	0.37	1.03	1.03	0.98	0.92	0.94
tc	2	1	5	0.54	0.33	0.53	0.37	0.63	0.33	0.69	1.07	1.01	1.30
tc	2	1	6	1.01	0.89	0.76	1.03	0.33	0.54	0.46	0.83	1.27	1.60
tc	2	1	7	0.64	0.73	0.46	1.03	0.68	0.46	0.54	0.46	0.74	1.23
tc	2	1	8	0.69	0.78	0.53	0.98	1.06	0.83	0.42	0.38	0.61	1.22
tc	2	1	9	0.49	0.66	0.67	0.95	1.03	1.31	0.78	0.58	0.54	0.67
tc	2	1	10	0.75	0.92	0.98	1.00	1.28	1.60	1.22	1.22	0.67	0.42

PUBLICATIONS IN THE PUBLIC FACILITY LOCATION SERIES

1. Giorgio Leonardi, *On the Formal Equivalence of Some Simple Facility Location Models.* WP-80-21.

2. Tony J. Van Roy and Donald Erlenkotter, *A Dual-Based Procedure for Dynamic Facility Location.* WP-80-31.

3. Donald Erlenkotter, *On the Choice of Models for Public Facility Location.* WP-80-47.

4. Giorgio Leonardi, *A Multiactivity Location Model with Accessibility- and Congestion-Sensitive Demand.* WP-80-124.

5. Yuri Ermoliev and Giorgio Leonardi, *Some Proposals for Stochastic Facility Location Models.* WP-80-176.

6. Giorgio Leonardi and Cristoforo Sergio Bertuglia, *Optimal High School Location: First Results for Turin, Italy.* WP-81-5.

7. Yuri Ermoliev, Giorgio, Leonardi, and Juhani Vira, *The Stochastic Quasi-Gradient Method Applied to a Facility Location Problem.* WP-81-14.

8. Giorgio Leonardi, *The Use of Random-utility Theory in Building Location-Allocation Models.* WP-81-28.

9. John Beaumont, *Towards a Comprehensive Framework for Location-Allocation Models.* CP-81-22.

10. Donald Erlenkotter and Giorgio Leonardi, *Facility Location with Spatially Interactive Travel Behavior.* WP-81-97.

11. Giorgio Leonardi, *A Unifying Framework for Public Facility Location Problems.* RR-81-28. Reprinted from *Environment and Planning A* 13:1001-1028, 1085-1108, 1981.

12. Giorgio Leonardi, editor, *Public Facility Location: Issues and Approaches.* RR-82-23.