Overview of the Eora World MRIO Project

Daniel Morana, Arne Geschkea

a ISA, School of Physics, University of Sydney, Sydney, NSW 2006

Note: this paper also appears in the forthcoming book *Computationally Intelligent Data Analysis for Sustainable Development* edited by T. Yu (2012)

**Summary**

The Eora multi-region IO table (MRIO) is a new high-resolution table that records the bilateral flows between 15,000 sectors in 187 countries. Such a comprehensive, high-resolution model is advantageous for analysis and implementation of sustainability policy. This paper provides an overview of how the Eora IO table was built. A custom data processing language was developed to read, aggregate, disaggregate, and translate raw data from a number of government sources into a harmonized tensor. These raw data often conflict and do not result in a balanced matrix. A custom optimization algorithm was created to reconcile conflicting data and balance the table. Building and balancing the Eora MRIO is computationally intensive: it requires approximately 20 hours of compute time per data year on a cluster with 66 cores, 600GB of RAM and 15Tb of storage.

Globalization combined with an increasing human population and increased consumption means that our ecological footprint is now falling more widely and more heavily across the planet. Our environmental impact is increasing, and lengthening supply chains mean that consumers are often far removed from the impacts they drive. In order to manage and reduce our footprint we must first be able to measure it. This is the aim of environmentally extended input-output analysis.

We want to identify which countries and sectors are directly causing, and which are ultimately benefiting from, environmental harms. We want to link individuals, households, and companies to the upstream environmental harms they ultimately drive. This principle is called *consumer responsibility*: the idea that consumers, not producers, should be responsible for pollution. Successful sustainability policy will consider both produce and consumer responsibilities, and will need to address different challenges at the production, trade, and consumption points along the supply chain.

There are two approaches to measuring footprints: bottom-up and top-down. Bottom-up approaches add up the various components of the footprint. For example the CO2 footprint of a paper shopping bag might be the sum of CO2 emitted by logging vehicles, the paper factory, the transport from the factory to the shop, and the decomposition of the bag after it’s been used. Top-down approaches consider the total footprint first and then allocate to different products, for example of total CO2 emissions how much is from paper products, and what fraction of paper products are paper shopping bags.

Life-cycle analysis (LCA) is the main methodology for bottom-up analysis. LCA traces inputs, processes, and outputs at all points along a production-use-disposal chain. LCA studies are complex, data-hungry, and often involve difficult boundary issues where it is hard to extract a single flow without undercounting (missing some input or output flows) or double counting (when two stocks or flows overlap so the sum of the parts exceeds the sum of the whole).

Input-output analysis takes a top-down approach. Generally the problem with top-down approaches is that they are low resolution. A top down analysis may be able to tell you the CO2 footprint of the average passenger vehicle, but cannot differentiate between an SUV and a hybrid sedan.

The Eora project uses sophisticated data processing to keep raw data in its original form, resulting in an IO table that is substantially higher resolution than any other produced to date. This high-resolution table enables better top-down analysis, and makes it easier to combine with bottom-up approaches in a so-called *hybrid LCA*. IO analysis is comprehensive, does not suffer boundary issues, and is increasingly accurate.

Input-Output tables were originally conceived by Wasily Leontief as an economic tool to analyse the flows between sectors of the American economy. Using IO tables Leontief could estimate how shocks to one sector might affect others. A variety of modern economic models, including widely used Computable General Equilibrium (CGE) models, are built upon IO tables. Most national statistical bureaux produce IO tables and use them as the basis of their national accounts, following the UN standard System of National Accounts. A multi-region IO table (MRIO) brings together many national IO tables into one large whole-world IO table. MRIOs are not built by national governments but instead by a small number of academic research teams. They are a powerful tool for international policy analysis, and environmentally extended IO tables are extremely useful for sustainability analysis. IO-based analysis has become increasing popular in recent years as greater data availability and computational power enable the creation of higher resolution, more accurate tables. The Eora multi-region IO table (MRIO) is the largest and most detailed IO table yet assembled. It has been made feasible by using sophisticated data processing automation, mathematical rather than manual techniques to balance and resolve data conflict, and modern computer hardware.

# Structure of IO tables

This section provides a brief overview of the structure of IO tables. To learn more, the authoritative text by Miller and Blair ([2009](#_ENREF_3)) is recommended.

The elements of an IO matrix are the sum of sales from one sector to another. Each row in the matrix represents the goods and services produced by one sector, i.e. that sector’s output. The columns along that row are the various sectors which consume those products. Each element in the matrix thus represents the total value of transactions between sector A and sector B. In a single-country IO table the rows and columns represent sectors within that economy. The matrix is not triangular because the flows from A to B need not (and almost never do) equal the flows from B to A.

This transactions matrix **T** is the main block of an IO table. Appended to it on the right is another block of columns **Y** representing final consumption by households, government, inventories, and so on. Also added below is a block of rows **V** for so-called “primary inputs”. The most important of these is value added. By treating value added as an input to production, a sector can sell its output for more than the sum of its other inputs. Figure 1 shows the layout of a single country IO table and illustrates the element of the transaction matrix recording the total value of Steel sector inputs to the Vehicles production sector.

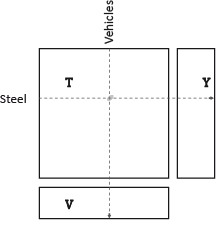


Figure 1 - A basic IO table with annotation showing where in the transactions matrix the total value of outputs from the steel sector to the vehicle production sector is recorded.

In a multi-region IO table (MRIO) the sectors represent each country’s sectors. For example, instead of one steel sector, Japanese steel and Australian steel are two separate rows and two columns. This is illustrated in Figure 2. It is possible to construct an MRIO where the regions are not countries but states, cities, or companies. This is only a challenge of data availability. It is also possible to use additional data to further disaggregate the table and thus gaining higher resolution, for example by finding additional sales and purchasing data that would allow the Japanese Steel sector to be split into Traditional Steel and Speciality Steel sectors.

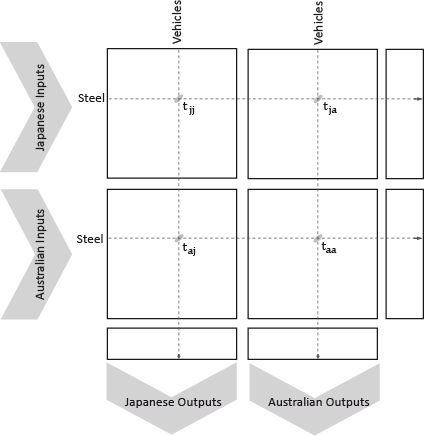


Figure 2 - A two-country MRIO. The transactions **t**jj, **t**ja, **t**aj, and **t**aa record the value of steel used in domestic vehicle industries and exported to the other country’s vehicle industry.

Resting below an IO table may be an additional block of rows for nonmonetary inputs such as labour hours or water usage. The rows of this so-called satellite indicator block **Q** are treated as additional inputs into production. In IO analysis pollution, too, is regarded not as a consequence of production but as a necessary input, and thus added as an indicator row. The Eora model has nearly 2000 indicator rows covering nonmonetary inputs such as energy use, water use, land area required (ecological footprint), GHG emissions, and air pollutants.



Figure 3 – Diagram of an MRIO with satellite accounts for threatened species

Figure 3 illustrates one path through an environmentally extended MRIO. The white blocks on the diagonal are domestic IO tables, the light grey blocks are trade blocks, and the shaded bottom row contains satellite indicator blocks. The data point at (1) records how many species are threatened by Brazilian coffee growers. Satellite accounts can contain any type of metric; this could just as easily be tons of CO2 emitted by Brazilian coffee growers or cubic liters of water used. Data point (2) records how many dollars worth of coffee Brazil sells to US food and drink companies. These two points make up a supply chain connecting US coffee consumers with species threats due to coffee plantations in Brazil. The structure of environmentally extended IO tables is outlined by the United Nations System of Environmental-Economic Accounts (SEEA). In this structure every environmental indicator (the satellite account row sums) is allocated amongst the various sectors (columns) which directly cause that impact.

The Eora MRIO covers 187 countries with a total of over 15,000 sectors. Heatmaps help visualize and validate IO tables. Figure 4 shows a heatmap visualization of Eora with a zoom-in on the Korea IO table. The Eora MRIO is not homogenous. Some countries have large high-resolution IO tables while other countries have small tables where the sectors are broadly defined. In order to offer the highest resolution and data integrity Eora maintains each national IO table in the national economic classification of the country providing it.



Figure 4 - Heatmap of Eora and zoom-in on the Japan IO table. Each pixel represents one element in the transaction matrix, with darker red colors representing larger transaction values.

The strong downward diagonal is a notable feature. One reason for the diagonal is that transactions between companies within the same sector can be large, particularly if the sectors are broadly defined e.g. if the Vehicles production sector includes companies selling automotive components as well as assembled vehicles. Same-sector transactions account for much of the diagonal. The diagonal also arises from the structure of common economic classification systems. Most sector classifications start with basic industries, ascend through intermediate processing industries, and end with final production and retail industries. To the extent that economic value creation follows this same stepwise path, a diagonal line will be seen as value cascades from primary industries in the upper left to tertiary production industries in the lower right. For example if the Wheat, Flour, Baking, and Retail Bakery sectors are adjacent in the classification system then much of the output from each will directly be used as input into the next, forming a diagonal in the table.

IO tables should be balanced. The sum of each row should equal the sum of each column. If an industry is not balanced it means the industry is either receiving inputs from, or delivering outputs to, an unknown source. A significant challenge in assembling a large MRIO table has been to ensure this balancing. We discuss below how the optimizer ensures the table is balanced.

The Eora MRIO is available in five valuations: sales prices as recorded by sellers, the additional cost of taxes paid, the value of subsidies received, transportation costs, and retail & distribution costs. These margins allow distinction of the several types of middleman price markups between producers and purchasers. Taxes, subsidies, transportation, and retail and distribution costs are added to the producer’s price to arrive at the purchaser’s price. These markups are represented by four additional sheets on top of the basic price IO table (Figure 5). Separating these markups into separate sheets is useful for analysing the effects of policy changes on taxes, subsidies, and transportation costs, and how these changes differentially affect sellers and purchasers.

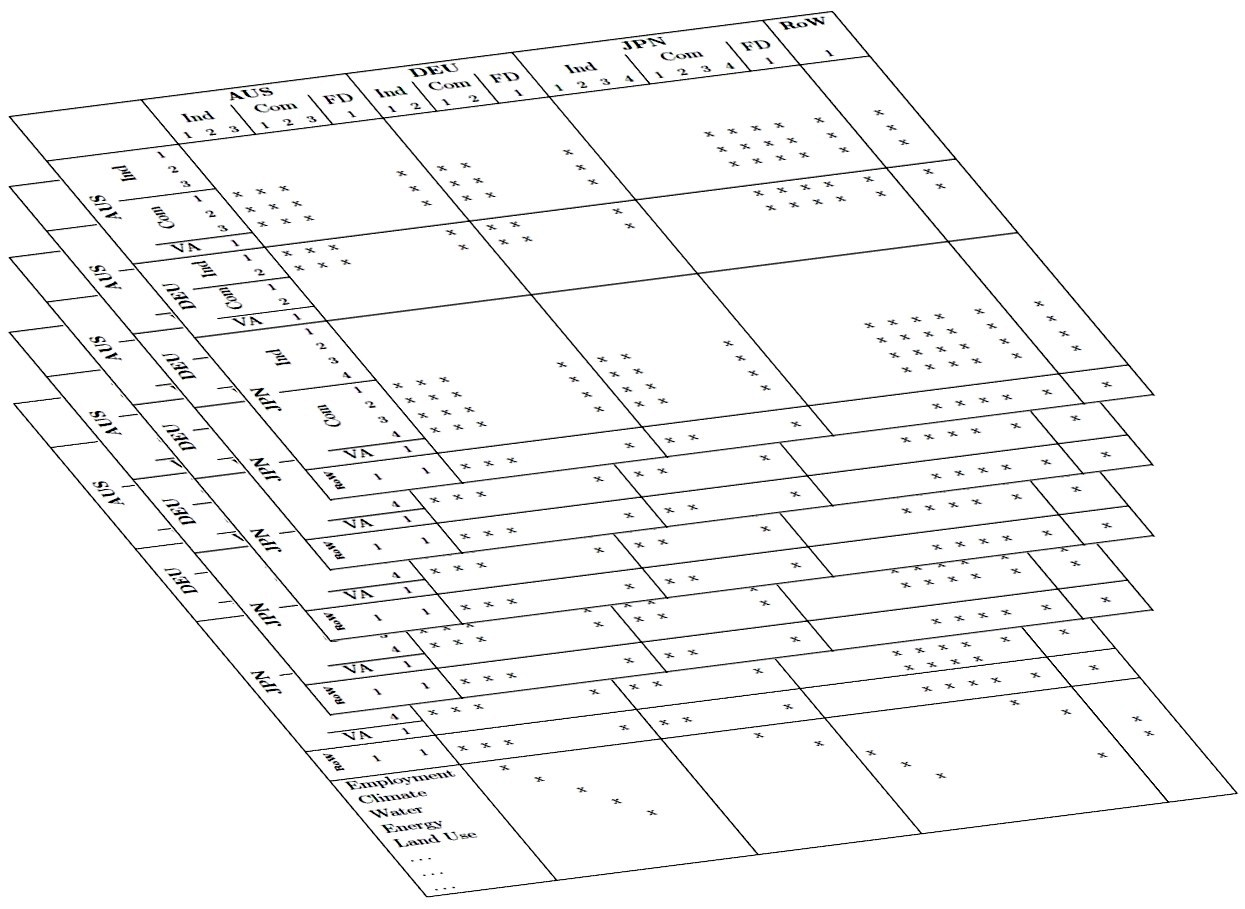


Figure 5 - An MRIO with five valuation sheets. Structurally, a 3 dimensional tensor.

Structurally Eora is a tensor. In the most intuitive representation it is a 3-dimensional tensor: a vertical stack of five matrices. However we address it as an 8-dimensional tensor. The rows and columns have a logical hierarchical grouping illustrated Figure 6.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Australia | | | | | | | | | | Japan | | | | | | Country N | |
| Industries | | | Commodities | | | Final Demand | | | | Industries | | | Final Demand | | | Industries | |
| Industry 1 | Industry 2 | Industry N | Commodity 1 | Commodity 2 | Commodity N | Households | Government | Inventories | Other | Industry 1 | Industry 2 | Industry N | Households | Government | Inventories | Industry 1 | … |

Figure 6 - Rows and columns on the sheets follow a tree structure hierarchy

To quickly locate a particular element in the tensor we use an 8-dimensional address of the form:

Input Country / block / sector index

Output country / block / sector index

Sheet

Year

The block dimensions group sectors as value add sectors, final demand sectors, industries, commodities, or satellite (**Q** block) sectors.

This 8-dimensional addressing is used in the data processing language to specify regions of the Eora tensor. Being able to efficiently locate sections of the tensor is important for being able to automatically populate the MRIO using many different data inputs.

# Populating the MRIO

## Data processing language, distributed database server

We want to populate the Eora tensor using data from a variety of sources (e.g. United Nations, national economic bureaux, etc.). The task is essentially the reverse of a database query. SQL and MDX are languages that help users easily address and aggregate subsections of a multidimensional tensor. We want to do the opposite: populate subsections of the tensor.

We faced two challenges. The first is that the source data we wanted to use are in a variety of formats and aggregation levels. We had to aggregate, disaggregate, and reclassify these input datasets so that each nation and its trading partner use the same classification scheme. The second challenge was more substantial: the raw data inputs often conflict.

The first challenge to building an MRIO is harmonizing all the input data so they may be combined in one table. A primary goal of the Eora project is to incorporate as many available economic and environmental data as possible. These data cover a wide range of aggregation level, formats, and classification schemes. Some examples of how the input data must be translated are:

* Reclassification: e.g. translate a set of trade data from Central Product Classification (CPC) to Harmonized System (HS) production classification.
* Aggregation: e.g. aggregate data in a 400-sector classification down to a 60 sector classification
* Disaggregation: e.g. allocate a single total value to sectors according to their contribution to GDP

We created a data processing language to help with this task. This language assists with aggregation and disaggregation and facilitates addressing and populating areas of the tensor. A disaggregation command in the processing language could specify a total value of steel sector sales that should be allocated proportionally, or equally, amongst a number of metal manufacturing industries.

Tightly integrated with the language is a substantial library of correspondence matrices. Correspondence matrices contain a weighted mapping that reallocates values in a source vector into a different destination vector. A correspondence matrix maps a source vector of length N to a destination vector of length M using an N x M matrix. Each row contains a set of weights which sum to 1. The first element of the source vector is allocated to the destination vector using the weighs specified in the first row, the second element added to the destination vector using the weights of the second row, and so on. Correspondence matrices are an especially convenient tool for aggregating and disaggregating data and projecting data in one classification scheme into another scheme.

For each data source (each table from all the agencies supplying data, e.g. UN, national agencies, environmental agencies, and so on) we wrote a processing language script to populate the tensor using the contents of that source. The scripts specify which concordance matrix/matrices to use to translate the data into the classification used in the Eora table. The scripts also specify the destination address in the Eora tensor where the reclassified data should be placed.

The data processing language interpreter is called AISHA. AISHA is essentially a database server. It reads data processing scripts in batch and populates the tensor. Our implementation uses a distributed architecture. A number of workers process these scripts separately then the master process aggregates the results afterward. This would be the end of the story if data sources did not overlap and conflict. AISHA does not actually populate a tensor but rather populates an initial estimate and constraint matrix which are run through an optimization package to produce the final tensor.

# Reconciling Conflicting Constraints: Eora as a Solution of a Constrained Optimization Problem

The MRIO table should satisfy several constraints. One basic constraint is a balancing constraint: the total outputs of a sector must equal the total inputs of a sector. Published data provided by statistical agencies adhere to this condition. But when constructing a large MRIO like Eora data from a number of different sources have to be combined into one table. This newly combined table will most likely not fulfil this balancing constraint. Some elements of the table must be slightly adjusted in order to ensure that the balancing condition is fulfilled.

Another common problem during the construction of an MRIO is that of conflicting data. Consider the following situation: we have an IO table from the US government, detailing the sectorwise composition of GDP. We also have a figure for total US GDP, provided from the UN. The two total GDP figures are not equal. These two figures are constraints, and the realized final value should lie in between. To reconcile these conflicting constraints we determine the reliability of each datum and ask the optimizer to find a solution value which maximally satisfies these two reliability-weighted constraints.

To build Eora the problem of generating a table that adheres to all specified constraints (balancing, conflicting data and/or others) was taken in one step. Reconciling the data in order to fulfil any given external condition was achieved by interpreting this problem as a mathematical constrained optimization problem.

We begin the process of building Eora by constructing an initial version of the table using raw data from a number of different sources. Let’s call this version of MRIO the *Raw Data Table*. If all these different data sources coincided, the MRIO would be finished. But most likely the challenges described in the previous section will hold: the MRIO will violate our specified constraints.

## Reliability Information

In order to approach the problem we first have to introduce reliability data. The Maximum Entropy Principle introduced by Jaynes ([1957](#_ENREF_1)) asserts that an IO-transaction is a random variable with a best guess and an associated reliability. IO tables are assembled carefully by statistical bureaux but still the individual inter-sectoral transaction values are not 100% certain. Hence both elements of the MRIO and external conditions are subject to a certain, individual reliability. The balancing condition for example has to be exact, otherwise an MRIO cannot represent an economy appropriately (see Leontief (1896)). This is a *hard constraint*. Other hard constraints include upper and lower bounds, specifying that transactions may never be negative or that subsidies (recorded as negative values) may never be positive. Other constraints like the previous example of the US GDP might be less reliable. These are *soft constraints*. Clearly, if the same piece of information from two different sources state different facts, at least one of the two must be incorrect. In reality this will most likely hold for both. But one of the data sources might be more reliable than the other one. In this case for example, the data for the total US GDP provided by a national statistical agency could be far more reliable than what the UN reported. Hence, an external condition that is not 100% reliable does not have to be 100% fulfilled by the MRIO.

This concept holds for the elements in the MRIO. Each transaction value in the MRIO is subject to certain reliability. That means that every element in the Raw Data Table can be adjusted (within a certain range determined by its reliability) and still represent the real transaction value reasonably well.

Reliability data is usually not published by statistical agencies. When this happens the reliability information can be estimated. We use a variety of heuristics to estimate the reliability of various datasets. Large transaction values are typically well-measured and thus more reliable than small values ([Lenzen et al. 2010](#_ENREF_2)). We assign a higher reliability to national statistical agency data than to UN data, and higher reliability to UN economic data than to trade data. The published Eora results include the information about our reliability estimates of every dataset. Since the final MRIO table is a composite of these data, we also provide reliability figures for the final table showing how reliable each value in the table is, based on the combined reliability of the various input data informing that result.

## The Concept of Constrained Optimization

The basic idea of constrained optimization is the following:

Obtain the Final MRIO by reconciling the elements of Raw Data Table according to their reliability in such a way that:

1. All hard external conditions are fulfilled exactly
2. All soft external conditions are well satisfied, the degree of fulfilment being determined by their reliability
3. The original Raw Data Table is minimally disturbed

Or in short: find a Final MRIO that fulfils all external conditions while minimally disturbing the Raw Data Table.

## Mathematical Interpretation of an MRIO

In order to apply constrained optimisation to the reconciliation problem we have to see the MRIO from a mathematical perspective. Consider a small MRIO

In mathematical terminology, the Raw Data Table is called *initial estimate.* The term *initial estimate* sometimes causes confusion as the Raw Data Table was sourced from officially published dataset; hence it is not an estimate at all. But from a mathematical point of view, the Final MRIO is the solution of an optimization problem and the Raw Data Table is the initial estimate of what the solution will be.

In order to fulfil the balancing condition, the sum over all elements of the first row of the table must equal the sum over all elements of the first column of the table. The same must hold for the second row and second column. The equations for the balancing constraints (or balancing condition) for the table **T** are given by

A trivial manipulation of these two equations yields

The diagonal elements and appear with altering signs in the equations, hence they cancel each other out. The final equations for the balancing constraints are thus

|  |  |
| --- | --- |
| (1) |  |

All variables and values of these two equations have a corresponding reliability. We know from the definition of IO that the balancing constraints must be fulfilled. Hence, the 0-values on the right hand side of the equations cannot be violated. That means that the 0-values have standard deviations of. Balancing constraints are hard constraints and hard constraints have a standard deviation .

An example for a soft constraint could be the following. Consider a data source is available that provides information that the total final demand of the MRIO *M* is equal to a value *a*. The corresponding equation for this information is

In this case, we can be almost certain that the value *a* is not totally reliable. Hence the standard deviation for the value *a* would not be equal to zero, i.e. . This equation can be violated by the Final MRIO that is to be computed. The acceptable amount of the violation is determined by the standard deviation of the value *a*.

## Summarizing the Balancing Constraints in a Single Matrix Equation

Now that the constraints are already available as equations the first step to a mathematical interpretation has been taken. In order to more easily work with the MRIO as an optimization problem we vectorize the transactions matrix **T** and call the resulting vector :

becomes .

This allows us to reformulate the first balancing constraint

as a vector-by-vector equation:

Hence the balancing constraint equation has the form

|  |  |
| --- | --- |
| (2) |  |

The vector *g* is the so-called *coefficients vector* holding the corresponding coefficients for the elements of the vector *p* to represent the balancing equation (1).

For every constraint equation there exists a vector *g* such can expressed in the form of (2). The *g*-vectors that for each constraint can now be summarized in a constraint matrix **G**. Each row of **G** represents one constraint. For the three constraint examples previously used in this section the coefficients matrix **G** is

|  |  |
| --- | --- |
| (3) |  |

The values that are on the right hand side of the corresponding constraints equations are summarized in the vector *c*.

With the concepts of Equation (3) we can write the complete set of external constraints that an MRIO has to adhere to elegantly as

The corresponding reliability data for the vectorized MRIO *p* and the right hand side values *c* can be stored in two separate vectors and that have the same sizes as the vectors whose reliability information they hold, namely *p* and *c*.

## Formulating the Constrained Optimization Problem

Let us call the Raw Data Table and the Final MRIO (that is yet to be calculated) . Let be the vectorization of and let be the vectorization of .

Recall the concept of the optimization problem that was stated before:

*Find a Final MRIO that fulfils all external conditions while minimally disturbing the Raw Data Table.*

With the concepts that were developed so far, the mathematical expression of the statement “*MRIO that fulfils all external conditions”* is simply given by

This system of equation is usually underdetermined, which means that there are more unknowns (in this case amount of elements in the MRIO) than constraints. Hence, there is usually more than one solution that adheres to the constraints (i.e. the Final MRIO is not uniquely defined by the constraints).

What remains to be interpreted mathematically is what it actually means to “*consider the Raw Data Table as well as possible”.*

As and are vectorized representations of the Raw Data Table and the Final Table, the change applied to the Raw Data Table in order to adhere to the given constraints can be “measured” by a so-called objective function . Since the system of constraints equation allows more than one solution, the optimizer is designed to find that solution that minimizes the objective function subject to the constraints. The optimization problem is therefore given by

For the Eora model, the objective function was based on Byron’s (1978) approach which uses a quadratic Lagrange function of the form where denotes a diagonal matrix with the -values of the vector in the main diagonal. First order conditions then have to be applied in order to find the Lagrange-multiplier and then Final MRIO.

In order to calculate the Lagrange-multiplier, a matrix inversion has to be calculated which would have proven to be calculation-intensive and possibly numerically unstable for a very large problem like the Eora problem. In order to avoid the explicit calculation of the Lagrange multipliers and matrix inversion, Van der Ploeg (1982) elegantly reformulates Byron’s approach using the following ideas:

1. Sort the rows of matrix and the right-hand side vector such that all rows for hard constraints are at the top, followed by the rows belonging to soft constraints. Let and denote the block of constraint lines that belong to the hard constraints and the soft constraints respectively. Then and take the form

and

Since soft constraints are not completely reliable, they may be violated to some extent. These violations are taken care for by introducing a disturbance for each soft constraint (note that there are no disturbances introduced for hard constraints as they have to be adhered to exactly). Let be the vector of all disturbances, then the system of soft constraints becomes

1. By defining

the system of equation for the soft constraints can then be re-written as

where denotes the identity matrix of appropriate dimensions. For the hard constraints we simply obtain

where denotes an all-zero matrix of appropriate dimensions. Hence, despite the introduction of disturbances for the soft constraints, the equations for the hard constraints haven’t changed.

By summarizing soft and hard constraints again into one equation, we obtain

The diagonal matrix of standard deviations for the vector then becomes

Where is defined as before and is the diagonal matrix of -values of . Since hard constraints do not obtain a disturbance variable does not contain -values that are equal to zero.

With these concepts, the optimisation problem can be rewritten as

with

The advantage of van der Ploeg’s approach is that the reliability information of the right-hand side vector was shifted to the iterate which is now called . The disadvantage is that the iterate (formerly , now ) grows by as many variables as there are soft constraints and hence, the problem becomes significantly bigger. However, the amount of constraints remains the same. The solution of this problem is the Final MRIO and the vector of disturbances of the soft constraints. The Final MRIO adheres to all constraints and considers the reliability of the raw data and the constraints during the calculation process.

Often, certain values of the MRIO have to stay within certain boundaries. Transaction values of the basic price table for example have to be positive values. Values in the … sheet can only be negative values. Hence, each element can be subject to an upper or lower bound, i.e. . By allowing positive or negative infinity as a feasible value for the upper or lower bound, a bound equation can be formulated for each element . The upper bound and lower bound values can be summarized in vectors of equal size to the size of . The boundary conditions for the whole MRIO can then be summarized as

Adding these boundary conditions to the optimisation problem, we obtain

|  |  |
| --- | --- |
| (4) |  |

The Eora model was generated based on the optimisation problem given by (4).

For the Eora project the total MRIO held roughly elements which were subject to constraints. The total amount of data that was considered during the optimization process was approximately 40GB. During the calculation process, more than 250 GB of RAM were used by the algorithm. As commercial algorithms weren’t able to solve a problem of this size, a custom parallelized optimization algorithm was developed. The frequency of iterations, the heavy communication load, and the fact that every worker requires **G** (the 30GB constraints matrix) together recommended the use of a large shared memory multiprocessor system. To build Eora for one year the optimizer runs for several hours on a commercial 24 core system with 288GB of memory. Solving the constrained optimization problem is tractable but pushes the limits of current computing hardware.



Figure 7 – Schematic of how the optimizer reconciles conflicting constraints. Data point D1 has a high confidence and data point D2, purporting to report the same value, has a low confidence. The optimal solution S minimizes the violation of the conflicting constraints and thus lies closer to the higher reliability constraint D1.

Because the raw data table and conflicting data are specified as soft constraints the optimizer is also able to generate an estimate of the standard deviation of each element in the result table. This estimate of the standard deviation is based on the degree conflict between the raw data table, conflicting constraints, and the degree to which this data point was adjusted to satisfy balancing constraints. Figure 7 illustrates how the optimizer reconciles two conflicting data points both purporting to report the same value. The optimizer would report the estimated standard deviation of the solution S as larger than the standard deviation of D1 but smaller than the standard deviation of D2. Eora is the first large MRIO to provide results along with estimate of the confidence of those results. Eora provides users with transaction-level reliability estimates and comprehensive reporting on which constraints (from the various input data sources) are best and least respected in the final table.

# The Leontief Inverse

Now we have a complete balanced, harmonized MRIO table. Nearly all environmental applications will proceed by calculating the Leontief inverse of the MRIO. The Leontief inverse answers the question: how much total input, both directly in the product and indirectly required for its production, is in a given unit of output? This calculation is the foundation of determining consumer-responsibility footprints, as it makes it possible to calculate the *total* environmental impact, including all impacts upstream in the supply chain, required to produce one unit of output.

The Leontief inverse is calculated as follows. Normalizing the transactions matrix **T** by gross output converts each element into a coefficient such that each column sums to one, i.e. each coefficient represents the relative contribution of each input per unit of output. This matrix is called **A**, the technical coefficients matrix or direct requirements matrix. A car, for example, contains steel directly (chassis, engine, etc.), but it also requires steel indirectly as an input to its production. The factory, car carrier, and showroom all use some steel as well, some fraction of which is needed for the provision of that car. This total requirement matrix L (in contrast to the direct requirement matrix) is the sum of an infinite series starting with the identity matrix I:

**L = I** + **A** + **A**2 + **A**3 + … = .

Leontief realized this is simply the matrix inversion

**L** = (**I** – **A**)-1

Each element of the matrix Leontief inverse matrix **L** thus reports the *total* quantity input required to produce one unit of output. The inputs can be weighted by their environmental load, using the satellite indicators, in order to find the environmentally-weighted total input required for each unit product produced.

Structural Path Analysis (SPA) can be used to selectively perform this series expansion on individual elements and trace out supply chains. SPA is commonly used to search the top-ranked (largest flow) supply chains or chains starting or ending at sectors of interest. The idea of SPA is not difficult but implementation is an art. SPA algorithms must essentially search a 15,000 \* 15,000 = 225,000,000-branch tree whose leaf node values asymptotically approach zero with depth. In the worst case a single input could visit every single sector in the world before being finally consumed. But overall, evaluating 10-15 links fully captures 99% of all supply chains. Still, intelligent heuristics for pruning and sorting are mandatory. The Leontief inverse and SPA are used complementarily. Footprints calculated using the Leontief inverse report the total footprint of products and sectors and SPA algorithms search for the individual supply chains involved.

# Applications of the Eora MRIO

The Eora MRIO provides a time series of MRIO tables with matching environmental and social satellite accounts for the entire world economy, at a high level of sector and country detail. 187 countries are represented at resolutions of 25-500 sectors, depending on raw data availability, tracing over 5 billion supply chains. The time series covers 1990-2010. The Eora MRIO presents a completely harmonised and balanced world MRIO table, incorporating millions of raw data points from major sources such as the UN System of National Accounts (SNA), UN COMTRADE, Eurostat, IDE/JETRO, and many national input-output tables. Every MRIO element comes with an accompanying estimate of reliability.

The Eora MRIO table tracks nearly 2,000 sustainability indicator line items covering:

* energy
* GHG emissions
* air pollutants
* ecological footprint (hectares)
* human appropriated net primary productivity (grams of carbon)
* water use (litres)

The Eora MRIO has also been used for studies on the footprint of biodiversity, linking 30,000 species threat records from the IUCN Red List to production industries and final consumers, and to study conflict mineral (coltan and rare earth metals) supply chains originating in Africa and Asia.

The power of IO analysis to distinguish and link producers, supply chains, and consumers makes it useful for developing sustainability policies. No other analytical methodology can quantify the links between producers and consumers and expose supply chains as comprehensively. This data-rich resource can be used to inform sustainability policies for producers, traders, and consumers.

Most environmental legislation is currently designed to control the footprint of production. The footprint of production can be constrained with regulation requiring cleaner production, protection and conservation measures, better enforcement of existing legislation, and by buyers demanding high environmental standards from suppliers.

Trade flows in environmentally deleterious products can be constrained. For example the Convention on International Trade in Endangered Species of Wild Fauna and Flora (CITES) broadly restricts any trade in endangered species and derived products. Proposed carbon taxes function similarly, effectively restricting trade of an undesirable good. Sustainability certificates such as those issued by the Forest Stewardship Council and Marine Stewardship Council serve to constrain trade in illegal and irresponsibly sourced forest and marine products. Currently, high-risk tradeflows are discovered and investigated opportunistically not systematically. IO analysis systematically and comprehensively quantifies supply chain paths. This is useful for identifying and ranking at-risk supply chains.

To constrain consumer demand, sustainability labels, such as dolphin-safe tuna, organic produce, and fair trade coffee, can be used to shift consumer demand away from high-impact products toward more responsibly sourced products. Corporate and government buyers can use footprint data to help build sustainable procurement policies ensuring those organizations consume products with a lower environmental footprint. One major motivation of environmental footprinting is to provide more information helping consumers understand their ecological footprint. The Eora MRIO table uses sophisticated computational techniques to provide data-rich answers that inform sustainability policy.

**References:**

Byron 1978: Byron, R. (1978). The estimation of large social account matrices. Journal of the Royal Statistical Society Series A, 141(3):359-367.

Jaynes, E. T. 1957. Information theory and statistical mechanics. *Physical Review* 106: 620-630.

Lenzen, M., R. Wood, and T. Wiedman. 2010. Uncertainty analysis for Multi-Region Input-Output models - a case study of the UK's carbon footprint. *Economic Systems Research* 22: 43-63.

Leontief (1986): Leontief, W. (1986). Input-Output Economics. Oxford University Press, New York, NY, USA.

Miller, R. E. and P. D. Blair. 2009. *Input-Output Analysis: Foundations and Extensions 2nd Ed*. Cambridge, UK: Cambridge University Press.

Van der Ploeg 1982: van der Ploeg, F. (1982). Reliability and the adjustment of sequences of large economic accounting matrices. Journal of the Royal Statistical Society Series A, 145(2):169-194.