Discussion Paper

Transitivizing elementary price indices for internet data using the cycle method

The views expressed in this paper are those of the author(s) and do not necessarily reflect the policies of Statistics Netherlands

2017 | 09

Leon Willenborg

Content

Summary

The present paper discusses a method to transitivize elementary price indices that have been calculated for internet data and that are not transitive. The paper is a continuation of Willenborg (2017a) that discusses a method to calculate the elementary price indices that are to be corrected with the cycle method, which is explained in the present paper. This method was proposed several years ago by the author. It can be shown to be a generalization of the well-known GEKS-method (see Willenborg, 2017b). Several scenarios are applied to update nontransitive price indices, some under the condition that previously computed values cannot be revised. These methods are illustrated by applying them to the same data, so that they can be directly compared with each other.

Keywords

Nontransitive price indices, transitivization, transitive closure, cycle method, multilateral index.

1. Introduction

In Willenborg (2017a) several methods are discussed that can be used to calculate elementary price indices for internet data. For the HWC project one such method, the group method, was used to calculate price indices for a particular web shop that sells clothing. This method uses an internal classification for clothing that was specifically developed for items on offer from clothing shops. The method is fairly simple to apply if the conditions are met that are supposed to hold to apply the method[1] (apart from the automatic coding of the clothing items) and yields results that look plausible. The method can undoubtedly be improved by refining the strata defined by the internal classification for clothing, which can be a bit broad, by smaller, more homogeneous substrata or subgroups. These are defined by using extra information available in the description of the various clothing items available at this web shop. The resulting subgroup method is central in Willenborg (2017a).

This subgroup method is employed to calculate elementary price indices at the subgroup level. This method, although easy to apply, yields elementary price indices that are not necessarily transitive. This may be undesirable. The advantage of transitive price indices is that they are free of chain drift. Chain drift may lead to chained indices that may differ markedly from corresponding direct indices.

In the present paper we discuss a general method to make such price indices transitive. For this process there did not seem to exist a verb, so we have coined one to denote it, namely ‘ transitivize’. The general method has been described before by the author (cf. Willenborg, 2010).

The transitivizing method discussed in the present paper is very general. It can be applied to any price index that is not transitive, not necessarily those described in Willenborg (2017a). So this method could be interpreted as a general method that can be used to ‘rectify’ or ‘repair’ a nontransitive price index and make another one that is transitive.

The transitivizing method is applied in the present report to certain elementary price indices for internet data, under various conditions. Some of these are about incremental methods that aim at producing nonrevisable price indices. That is, it is impossible to change previously calculated price indices. This is a realistic situation as publication policies of a statistical office usually imply that price indices should be published as soon as possible, and allowing no rectification of earlier published results (except for possible blunders, of course). This requires that the original

[1] In terms of processes, this is the *happy flow*. If the conditions for the happy flow are not met then one is dealing with the *exceptional flow*. A precondition for the application of the group method is that there are average prices available for both the base month and the reporting month. In this case the corresponding price index is easy to calculate. In case either of them is missing one has to resort to methods to find approximate average prices, for instance that of a larger group of clothing articles.

method for the transitivizing method is adapted to new conditions. As is shown in the present report, this is very well possible.

The remainder of the paper is organized as follows. In Section 2 the general method, described more extensively in Willenborg (2010), to adjust arbitrary price indices, is summarized. This method assumes that all data are available for adjustment. It is called the 'simultaneous method' in the present paper for that reason. In practice, however, the data typically become available in monthly portions. The goal is to publish price index numbers incrementally, adding new figures as soon as new data become available. These incremental methods use the simultaneous method described in Section 2 repeatedly and under certain restrictions due to the way price index information is compiled and how it is published. In Section 3 two such incremental methods are considered. Section 4 describes problems related to aggregating and adjusting price indices at various levels of aggregation. Section 5 deals with an efficient way of presenting the adjusted price indices, namely using spanning trees. In this section the results obtained in earlier examples are also compared. Section 6 concludes the main text with a summary and a discussion of the main results. It also contains some suggestions for future research. A list of references and four appendices complete the report. Three of these appendices provide details of examples discussed in the main text. The fourth appendix is a brief exposition of some basic concepts from graph theory, some of them flavoured by price index theory. These notions and the terminology introduced here are used throughout the present document.[2]

2. Simultaneous method

The subgroup method in Willenborg (2017a) is based on pairing 'like subgroups' and yields elementary indices, but these are not by definition transitive, as in case of the indices calculated using the group method. But these intransitive indices can be made transitive using the method proposed in Willenborg (2010), which is summarized in section 2.1. There also some notation and terminology is established for the remainder of the report. To fully understand what is going on, the reader is referred to this report. Below, only a summary of the main results is given.

2.1 Chain indices and path independence

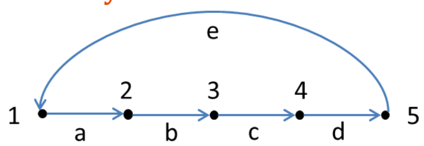
The aim of the method described in this section is to produce transitive price indices from intransitive ones. In terms of Balk (2008) they satisfy the circularity (or transitivity) test (test T1 in Section 3.4.1 in Balk, 2008), i.e. $p_{ij}p_{jk} = p_{ik}$ for all states i, j, k. Here, p_{ij} denotes a price index that compares period j to period i. If we choose

[2] The present document was reviewed by Sander Scholtus.

$i = j = k$ it follows that $p_{ii}p_{ii} = p_{ii}$ which yields $p_{ii} = 1$ for all i, as the option $p_{ii} = 0$ is ruled out. If we choose $k = i$ in the transitivity identity and use $p_{ii} = 1$ we obtain: $p_{ij}p_{ji} = 1$ for all states i, j. If the latter condition holds, the index is said to satisfy the time reversal test. If it satisfies $p_{ii} = 1$ for all states i, is said to satisfy (a weak form of) the identity test. If the transitivity identity is repeatedly applied, identities for larger cycles can be obtained.

The transitivity test yields constraints in multiplicative form. The tests can be summarized by saying that in a cycle of a Price Index Digraph (or PIDG)[3] the product of the price indices associated with its arcs (oriented in the same way) should be 1. Consider the example in Figure 2.1.1, representing 5 consecutive months in a year for some group of cloths. The chain indices have been calculated for the ordered pairs of months (1,2), (2,3), (3,4), (4,5) and (5,1). For each of these pairs the first entry indicates the base month and the second one the reporting month. These pairs of months correspond to the arcs labelled a, b, c, d and e respectively.

2.1.1 A cycle in a PIDG.



Let the indices associated with the arcs be denoted by x_a, x_b, x_c, x_d, x_e. Then transitivity requires that $x_a x_b x_c x_d x_e = 1$.

For our purposes it is more convenient to look at the logarithms of these indices, under the provision that they are all positive. Then the transitivity requirement translates into the additive requirement that $\log(x_a) + \log(x_b) + \log(x_c) + \log(x_d) + \log(x_e) = 0$, where $\log$ denotes the natural logarithm.[4] Note that this is a linear constraint in terms of the (natural) logarithms of the x's. It depends on the circumstances what type of constraint to choose, multiplicative or additive, as they are equivalent for all practical purposes. In our adjustment models it is more convenient to use linear constraints.

Suppose that a PIDG $G = (V, E)$ is given, with n points in V and m arcs in E. In particular we assume that the PIDG has been pruned of linear parts ('filaments').[5] Let y denote the 'vector of observations', which in our case corresponds to the natural logarithm of the calculated elementary price indices associated with the arcs of G. We shall call these values elementary log-price indices. Let W be a nonsingular, diagonal, nonnegative, $m \times m$ weight matrix, associated with each of the arcs of G. W controls which values can be perturbed more, and which should be perturbed less. Let C be a cycle matrix associated with G, of order $(m - n + 1) \times m$. $C = (c_{ij})$ is a (-1, 0, 1)-matrix, where i indices the elementary cycles and j the arcs. Suppose that

[3] See Appendix D for this and other related graph terminology.
[4] This is only for convenience. Any other base for the logarithm could be chosen instead of e.
[5] Please refer to Willenborg (2010) for details.

$j = (v, w)$, then $c_{ij} = 0$ if arc j is not part of cycle i, $c_{ij} = 1$ if $j = (v, w)$ is part of cycle i, and $c_{ij} = -1$ if the reverse of arc j, i.e. (w, v), is on cycle i.

Let $\hat{x}$ denote an adjustment of y that satisfies the cycle condition $C\hat{x} = 0$. We assume that $\hat{x}$ is obtained by minimising the expression $(x - y)'W^{-1}(x - y)$ under the condition $Cx = 0$. This can be achieved using the Lagrangian multiplier method. It yields an 'estimator' that is known as an RGLS-estimator[6]. In our case we have:

$$\hat{x} = y - WC'(CWC')^{-1}Cy = (I_m - WC'(CWC')^{-1}C)y = \Pi y \tag{2.1}$$

where I_m is the $m \times m$ identity matrix and $\Pi = I_m - WC'(CWC')^{-1}C$, which is an $m \times m$ matrix. The matrix CWC' is non-singular because C is of full row rank $m - n + 1$ and $W$is non-singular. The matrix Π is a projection matrix, for which $\Pi^2 = \Pi$ holds (idempotency). Its rank equals $Tr\ \Pi = m - (m - n + 1) = n - 1$, with $Tr\ \Pi = \sum_i \Pi_{ii}$, where Tr denotes the trace operator. Π satisfies $C\Pi = 0$, which implies $C\hat{x} = 0$, as required.

If we write $\Sigma = C'(CWC')^{-1}C$, we have $\Pi = I_m - W\Sigma$, with I_m, W, Σ symmetric matrices, so that $\Pi' = I_m - \Sigma W$. We have:

$$\begin{aligned}(\hat{x} - y)'W^{-1}(\hat{x} - y) &= (\Pi y - y)'W^{-1}(\Pi y - y) = y'\Sigma W \Sigma y \\ &= y'C'(CWC')^{-1}Cy\end{aligned} \tag{2.2}$$

which, statistically, can be interpreted as a variance.

The results above are in terms of log-price indices. Equation (2.1) shows that each adjusted index $\hat{x}_i$ is a linear function of the original log-price indices y_j.[7] So if we write $\hat{x}_i = \sum_{j=1}^{m} \alpha_{ij} y_j$, where $y_j = \ln(p_j)$ and $\hat{x}_i = \ln(\hat{p}_i)$, then

$$\hat{p}_i = \prod_{j=1}^{n} p_j^{\alpha_{ij}}. \tag{2.3}$$

So (2.3) presents the general form of the adjusted indices $\hat{p}_i$, in terms of the original indices p_j. It is understood that if p_j is the price index associated with arc j, p_j^{-1} is associated with the reversed arc. The adjusted price indices have the property that if one takes the product of all the adjusted indices belonging to the arcs on any cycle in the PIDG, the result equals 1. From this follows that if one wants to calculate the price index between a base point in the PIDG and a reference point, one can take any path in the PIDG and multiply the indices corresponding to the arcs on the path. The result is independent of the path chosen.

Also note that the form of the adjusted indices in (2.3) is that of Cobb-Douglas type (see Balk, 2008, p. 97). The rough elementary indices are of this generalized Cobb-

[6] RGLS=Restricted Generalized Least Squares. In our case we are in fact not dealing with an estimator at all, because there are no observations with errors, yielding random variables. Our setting is deterministic, not stochastic. But in this setting exactly the same result is obtained as in the statistical setting.
[7] Note that the indices i, j are in fact pairs of indices.

Douglas type. It is a nice feature to work within a single class of index formulas (i.e. the class of generalized Cobb-Douglas type indices). If the adjusted price indices in (2.3) were to be described in terms of the rough elementary indices, they would be of this form too (with different exponents, of course). This coherence in form is very attractive.

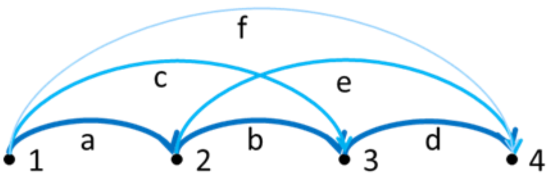
Note that the results are free of any distributional assumptions: there is no randomness involved. For our problem we do not need it, for it is not statistical: we are not dealing with random variables. We therefore do not have to look at the (co)variance of $\hat{x}$, and the like. We only need an interpolation technique, to calculate adjusted results. Having said that, it does not harm to think statistically either. For instance, when choosing a weight matrix (in Section 2.3), statistical arguments are used to come up with a sensible choice for such a matrix.

Spanning trees

The adjusted price indices are transitive. That is how they were constructed. It is convenient to use a spanning tree to store some of the price indices, and generate the remaining ones in the full PIDG by transitive closure. See Appendix D for background information on spanning trees and transitive closure.

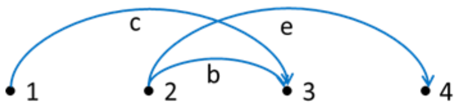
In the PIDGs we consider, the month-on-month digraph (MoM digraph, for short) is the most special one. In view of the weights that we use for the optimal adjustments, it is a minimal spanning tree, in the sense that the sum of the weights associated with the arcs is minimum among the spanning trees associated with the PIDG. This spanning tree is special because its arcs only connect neighbouring months. And these are most closely connected, 'time wise' and for instance in terms of overlap of articles in subgroups. The overlap is likely to be maximal for neighbouring months. In Figure 2.1.2 an example of a PIDG is given, in fact one that is used in several examples in the current report. The arcs are coloured and drawn with line widths of different thickness. The darker the colour and the thicker the line width of an arc the more important that arc is in the PIDG. The arcs that are dark blue are those of the optimum tree.

2.1.2 PIDG, an optimum spanning tree (arcs in dark blue) and remaining arcs.



Perhaps, the reader is of the opinion that the optimum spanning tree in Figure 2.1.2 is in fact the only spanning tree in this PIDG, as it is so natural. But it is not. In Figure 2.1.3 another spanning tree is shown. It is of no particular use at all, except to show that a spanning tree for a digraph in general or PIDG in particular need not to be unique.

2.1.3 Another spanning tree for the PIDG in Figure 2.1.2.

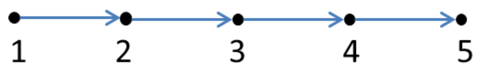


This method via the minimum spanning tree offers a natural way to produce a transitive price index graph. As soon as we have specified the values associated with each of the arcs of the minimum spanning tree, the values for the remaining arcs can be deduced, using transitive closure. This includes arcs pointing from a month to an earlier one. Through exponentiation, one calculates the corresponding price indices.

2.2 Applying the adjustment method

In case one would only calculate the month-on-month price indices, MoM price indices for short, there is no need (nor even possibility) to apply the adjustment method. There are no values to compare. Of course one can choose to do this and one is sure to obtain transitive price indices, by applying transitive closure. That is basically the idea of Hill's method. See Balk (2008, section 7.6). The linear digraph just described can be viewed as a spanning tree, even an optimal one, under natural conditions. In Figure 2.2.1 such a tree is shown, for a 5-month period.

2.2.1 Mom-digraph, indicating that only prices in adjacent months are compared.



But this method, based on MoM price indices, discards comparisons of prices in a subgroup that are several months apart. If we do take these into account we are likely to find that the resulting PIDG is not transitive. The adjustment method described in the previous section does produce transitive (log-)price indices, however. The (log-)price indices corresponding to adjacent months are adjusted only slightly, as a result of using this extra information from months that are further apart.

The gist of the method is to calculate, for a given set of articles, the elementary price indices p_{ij} for some extra months i, j some distance apart, but possibly not too far.[8] One can calculate the elementary price indices at the subgroup level as explained in Willenborg (2017a). These price indices will be used to correct the rough price indices $p^r_{i,i+1}$, where the superscript denotes the 'roughness' of the indices.[9]

As we have seen (see Section 2.1) for the adjusted price indices it holds that $p_{ii} = 1$ for all i, and $p_{ij}p_{ji} = 1$ for all i, j. So if we look at the P-matrix we only need to con-

[8] What is a good limit to this distance is a point that needs further investigation. The more apart months are, the smaller the number of articles they have in common, on average.

[9] Technically this could also be done at the item level. But this would create problems when trying to deal with the dynamics of the article population. The subgroup method deals with this problem automatically.

centrate on one of its triangles, the lower triangle ($i > j$) or the upper triangle ($j > i$). The values in the other half can then be simply calculated from the half considered.

To calculate the p_{ij}'s we can start at the subgroup level by calculating rough elementary indices. These are aggregated to the group level, using a method from Section 5 in Willenborg (2017a). The aggregation should be done only for those pairs of months that one wants to use for adjusting the MoM price indices.[10]

In order to adjust the original price indices at the group level, weights should be used. It does not seem to be reasonable that all arcs in the PIDG should count equally 'heavy' in the adjustment procedure. It is better to differentiate between the amount of adjustment allowed for the various arcs involved. See Section 2.3 for more on this topic.

As explained in Section 2.1 the adjustments need only be applied to arcs that are part of cycles in the PIDG. All other arcs or linear pieces ('filaments') can be discarded. Once the adjustments are made, transitive closure may be invoked to calculate the price indices associated with each of the arcs in the complete PIDG. But it has to be checked in practice that the PIDG with rough price indices is indeed connected.[11] In the present report we assume that it is.

2.3 The weight matrix

In Willenborg (2010) where the method explained in Section 2.1 originates from, it is suggested how that weight matrix could be chosen. This is actually a suggestion taken from Balk (2008) who has borrowed it in turn from two papers by Hill; it is referred to as the Paasche-Laspeyres spread. But as indicated in Willenborg (2010, section 4.3) alternative choices of the weight matrix W are possible. In the present paper we shall not use the Paasche-Laspeyres spread. We will be guided by the distance $|j - i|$ for a price index p_{ij}.

As to the class of feasible weight matrices ($M = W^{-1}$) we restrict our attention at the outset to diagonal matrices[12], with strictly positive elements on the main diagonal: $W_{ee} > 0, M_{ee} > 0$. This means that every arc e in the PIDG considered gets a weight. If the off-diagonal elements of W would be nonzero, then we would have associated weights to pairs of arcs, which is difficult to understand what it means.

An easy way to find weights for the various arcs is the following. Let Ω denote a suitable class of weight matrices, as of yet unknown. We will fence it in in several steps. It should become clear that such an Ω is tailor-made to a specific problem. As specified before (Section 2.1), a weight matrix should be square, non-singular, non-negative and diagonal. We follow the method suggested above, but now using some

[10] This is not only true for the adjustment method described in Section 5. It is also true for the incremental method described in Sections 5.4 and 5.5.

[11] A graph is connected if each pair of its vertices can be joined by a path.

[12] M is diagonal if and only if W is.

parameters. In Table 2.3.1 the rows and columns are indexed with the periods (months). They correspond with the points in the corresponding PIDG. There are several parameters λ_i, for which we assume to hold $1 \leq \lambda_1 \leq \lambda_2 \leq \cdots \leq \Lambda$, where Λ is a suitable upper bound chosen for the problem at hand. So the distance between months depends only on the numbers of months separating them. Also, the idea is that the value associated with an arc connecting more distant months may be perturbed more than months closer to each other. In Table 2.3.1 $\lambda_1 = 1$ is chosen. This value acts as a gauge value, providing a reference for the values of the remaining λ's. In practice finding useful weights requires some tinkering with real data.

From this table we can infer the weight for arc (i, j). To give some examples, take arc (3,5) and we find the weight λ_2, for arc (2,8) we find weight λ_6. We can even go further and require, for instance, that $\lambda_i \geq \lambda_{i-j} + \lambda_j$ for any j such that $1 \leq j < i$.

2.3.1 Input for a weight matrix.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**…**
1	-	1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	…
2	1	-	1	λ_2	λ_3	λ_4	λ_5	λ_6	…
3	λ_2	1	-	1	λ_2	λ_3	λ_4	λ_5	…
4	λ_3	λ_2	1	-	1	λ_2	λ_3	λ_4	…
5	λ_4	λ_3	λ_2	1	-	1	λ_2	λ_3	…
6	λ_5	λ_4	λ_3	λ_2	1	-	1	λ_2	…
7	λ_6	λ_5	λ_4	λ_3	λ_2	1	-	1	…
8	λ_7	λ_6	λ_5	λ_4	λ_3	λ_2	1	-	…
…	…	…	…	…	…	…	…	…	…

For a problem at hand, and in particular the arcs that are present, a weight $W_{\bar{\lambda}}$ matrix should be assembled from Table 2.3.1. Here $\bar{\lambda}$ is the vector of λ's that appear in the assembled matrix $W_{\bar{\lambda}}$. See e.g. the choice of a weight matrix in the example on the simultaneous method in Section 2.4.

2.4 Examples

Example: Problem size

This is a continuation of the example in Section 5.3 on the use of the subgroup method in Willenborg (2017a). This example illustrates in a concrete case how rough indices are calculated. In the example there are twelve months and a group of articles consisting of 6 subgroups. It is easy to see how an upper bound of the computational complexity of the application of the subgroup method is in terms of the size of the problem, determined by the length of the period (counted as the number of months), and the number of subgroups. More precise estimates can be made when it is known for which subgroup-month combinations prices are known. Also, what the maximum separation is for two months to be compared, determines the computational complexity of an instance of the subgroup problem.

The present example is used to illustrate the size of the problems one can expect when transitivizing at the group level. As will be clear, the resulting matrices are of a manageable size.

Referring back to the example in Section 5.3 on the use of the subgroup method in Willenborg (2017a), note that the maximum number of arcs at the subgroup level is at most equal to

number of subgroups x $\binom{number\ of\ months}{2} = 6 \times \binom{12}{2} = 396.$

So in general if there are N_{sg} subgroups there are at most $66N_{sg}$ arcs at the subgroup level.

For the aggregated price graph, with 12 states (months), we have as the maximum number of arcs to consider $\binom{number\ of\ months}{2} = \binom{12}{2} = 66$ arcs.

To calculate transitive price indices with the method in the present section should be quite feasible, as we have to work with square matrices of order 66, and a cycle matrix not exceeding order 55 x 66 (= (number of arcs – number of points +1) x numbers of arcs).

The maximum number applies to the case where for each month and for all subgroups prices are available, and are being used. This need not be the case, as one can choose to calculate only rough indices for pairs of months that are not too far apart. So a bound on this clearly limits the number of indices to be calculated at both the subgroup and group level.

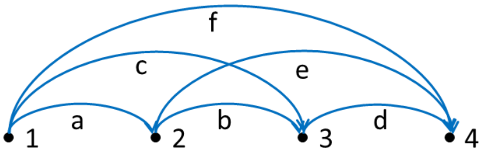
The results apply when the so-called simultaneous model is applied, using all data for an entire period (see the example on the simultaneous method below). In case an incremental problem has to be solved the computation may be smaller (see the example on the incremental method without revisions in Section 3.2) or bigger (see the example on the incremental method with revisions in Section 3.1) in total, after the period is over. However, with an incremental method calculations have to be made every month. ■

Example: Simultaneous approach

We now illustrate the method introduced in the present section in terms of a small numerical example. We only present the results here. The details of the calculations can be found in Appendix A. The basic data of this example will be used to illustrate various methods to transitivize price indices (Sections 3.1 and 3.2). It should be stressed that the result can only be achieved after all data of the full period considered (4 months in this example) have been collected.

We consider a group of articles that is subdivided into 7 subgroups. We have a period consisting of 4 subsequent months. From the subgroups we calculate the starting values for each ordered pair of months, i.e. $(1,2), (2,3), (1,3), (3,4), (2,4), (1,4)$. We denote the arcs by letters: $a = (1,2)$, $b = (2,3)$, $c = (1,3)$, $d = (3,4)$, $e = (2,4)$ and $f = (1,4)$. The PIDG is displayed in Figure 2.4.1.

2.4.1 A PIDG for a four-month period.



It should be stressed that this example is intended to illustrate the method in a concrete, but artificial case. The example shows how transitivized results can be obtained fairly easily. Computations have been performed in Excel. Details can be found in Appendix A.

The starting values are the logarithms of the rough price indices. They form the vector y. We have[13]

$$y = \begin{pmatrix} y_a \\ y_b \\ y_c \\ y_d \\ y_e \\ y_f \end{pmatrix} = \begin{pmatrix} 0.252147344 \\ 0.149176924 \\ 0.130465037 \\ 0.055268825 \\ 0.238375362 \\ 0.126974666 \end{pmatrix}$$

A cycle matrix for the PIDG in Figure 2.4.1 is

$$C = \begin{pmatrix} 1 & 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & -1 & 0 \\ 1 & 1 & 0 & 1 & 0 & -1 \end{pmatrix}.$$

The columns correspond to the arcs a, b, c, d, e, f, respectively. The weight matrix that we use is motivated by the goal to disturb the values of the arcs a, b and d not too much, those associated with arcs c and e a bit more, and arc f the most. For the calculations we have used

$$W = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 & 0 & 20 \end{pmatrix}.$$

Using formula 7.1 we obtain as the adjusted estimate for y:

[13] In this and other numerical examples we use high precision to represent the numerical values, just to show in much detail the effect of the calculations. In practice one would use this high precision only for calculations, but not for display and publication.

$$\hat{x} = \begin{pmatrix} \hat{x}_a \\ \hat{x}_b \\ \hat{x}_c \\ \hat{x}_d \\ \hat{x}_e \\ \hat{x}_f \end{pmatrix} = \begin{pmatrix} 0.218098687 \\ 0.121884527 \\ 0.339983213 \\ 0.048928244 \\ 0.170812771 \\ 0.388911457 \end{pmatrix}.$$

Note that the transitivity condition holds: $\hat{x}_c = \hat{x}_a + \hat{x}_b$, $\hat{x}_e = \hat{x}_b + \hat{x}_d$, $\hat{x}_f = \hat{x}_a + \hat{x}_e$. Note also that this is not the case for the corresponding original values, i.e. the components of y. ■

3. Incremental methods

In the approach described in Section 2 the assumption is that the data are available for the entire period and that adjustments of the rough elementary indices can be made in this entire period. However, in practice an incremental method should be applied. As soon as data about a new month become available they are used immediately to update the price index information. We consider the following two methods in the present section:

1. **Incremental method with revisions**. There are two calculations, one 'behind the screen' and one 'in front of the screen'. The latter concerns the officially published price indices where the restrictions about nonrevisability of previously published indices hold. The former concerns auxiliary computations that can be carried out without these restrictions. This means that 'behind the screen' the simultaneous method can be applied, each month using old and now data. All values associated with the arcs in the PIDG at that time can be changed in the calculations behind the screen. The result of these calculations is an adjusted value of the price index associated with the current month and the month before. This yields an updated spanning tree used 'in front of the screen'.
2. **Incremental method without revisions**. In this case we only have an 'in front the screen' situation. The method updates PIDGs step-by-step. Price indices estimated in this way in previous months cannot be altered. Only the values associated with the newly added arcs (pointing from previous months to the newest month) can be changed.

The essence of both methods is that the values associated with the optimal spanning tree used 'in front of the screen' are calculated, step-by-step. This is a spanning tree as in Figure 2.2.1, depicted for a five month period. It is optimal in case the arcs are weighted with elements from the weight matrix that is used for the adjustment calculations. This tree consists of subsequent months 1, 2, 3, … and the arcs (1,2), (2,3), … corresponding to MoM price indices.

3.1 Incremental method with revisions

This method is an easy adaptation of the method of Section 2. It requires that at any new month m a similar, but smaller, problem as in Section 2 is solved, namely for the first m months. We only have to consider the period $1, \ldots, m$, and corresponding sub-matrices from the cycle and weight matrix and from the y- and $\hat{x}$-vectors.

Although the method also recalculates previously adjusted values (log-price indices), only one value of the newly calculated adjustments is used, namely that of the newly added arc on the optimal spanning tree, i.e. arc $(m-1, m)$. So, in effect, step-by-step, we calculate the values of the optimal spanning tree.

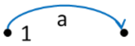
We illustrate the method in the following example.

Example: incremental method with revisions

We use the same data on the subgroups as in the example on the simultaneous method in Section 2.4, but we use them a bit differently. In the present example we only give the results. Details on the calculations can be found in Appendix B. We describe how the spanning tree for MoM price indices is built, step-by-step.

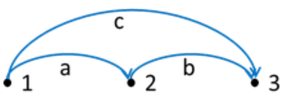
We start with the addition of month 2. Figure 2.4.1 represents the situation. Then the log-price index can be calculated from the subgroups. There is nothing to compare it to, so this is the value for the first extension. In our case we have $\hat{x}_a = 0.252147344$. This value is used for publication and cannot be revised later. So we have $\hat{x}_a^* = 0.252147344$, where the asterisk denotes fixation of the value, that is for publication purposes. Figure 3.1.1 depicts the situation.

3.1.1 First step in which the value for arc a is established.



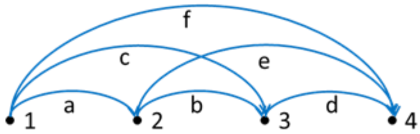
Next, we add month 3. This yields two new arcs, b and c. The situation is depicted in Figure 3.1.2. For all three arcs we have starting values computed from the subgroup data. We have $y_a = 0.252147344$, $y_b = 0.149176924, y_c = 0.130465037$. This PIDG has one cycle. Note that $y_c \neq y_a + y_b$. After adjusting, the values are as follows: $\hat{x}_a = 0.229575741$, $\hat{x}_b = 0.126605322$, $\hat{x}_c = 0.356181063$. Note that indeed $\hat{x}_c = \hat{x}_a + \hat{x}_b$. From this step only the value of x_b is retained and fixed, for publication purposes, so we write $\hat{x}_b^* = 0.126605322$, where the asterisk denotes fixation of the value. Also the value for x_c is fixed, so we can write $\hat{x}_c^* = \hat{x}_a^* + \hat{x}_b^*$.

3.1.2 Second step in which the adjusted value for arc b is calculated.



In the final step in our example, month 4 is added. This is shown in Figure 3.1.3. Three new arcs have been added to the existing situation in Figure 3.1.2, namely arcs d, e, f. We are now in the same situation as in the example on the simultaneous method in Section 2.4 and will obtain the same results for the adjusted values for a calculation ‘behind the screen’. However, this time we will only retain the one for arc d, i.e. $\hat{x}_d^* = 0.048928244$. From this the results for $\hat{x}_e^*$ and $\hat{x}_f^*$ follow. They are also fixed for use 'before the screen', that is as published results. The asterisks indicate this. The price indices are obtained by exponentiating $\hat{x}_a^*, \hat{x}_b^*, \ldots.$, that is, by $e^{\hat{x}_a^*}, e^{\hat{x}_b^*}, \ldots$.

3.1.3 Third step in which the adjusted value of d is calculated.



So in three steps we have calculated the adjusted values for the spanning tree with arcs a, b and d: $\hat{x}_a^* = 0.252147344$, $\hat{x}_b^* = 0.126605322$ and $\hat{x}_d^* = 0.048928244$. ■

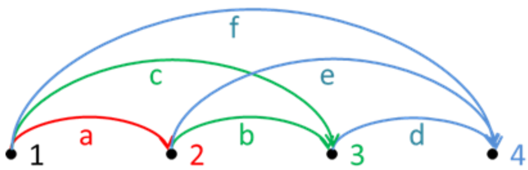
3.2 Incremental method without revisions

Contrary to the method in the previous section, the method in the present section never revises a single value that has been calculated in a previous step; only new values can be adjusted.

Example: Incremental method without revisions

We use the same initial data as in the example on the simultaneous method in Section 2.4. In Figure 3.2.1 we have depicted the PIDG that we will build. The vertices and the arrows are differently coloured, to indicate that they will become 'active', that is, will be subjected to adjustments, at different moments in time, namely every time when the prices of a new month become available. We have drawn only one digraph, instead of several, as in the example on the incremental method with revisions in Section 3.1, to stress that the PIDG is in fact constructed step-by-step, by adding new results and keeping previous results.

3.2.1 Indices for four months, incrementally added.



We discuss the method, providing only the main results. Details can be found in Appendix C. We have used an approximate method, but the approximations are pretty good. Exact solution methods are a bit more involved, as they require separate optimization problems to be solved to optimality.

First month 2 is added. Only month 2 and 1 can be compared. For month 2 only one arc, namely arc a (the red arc), and corresponding rough elementary index is added. This index is also the final one, as there is nothing to compare it with. We have (see

Appendix B): $\hat{x}_a^* = y_a = 0.252147344$. The asterisk indicates, as before, that this value is fixed for all subsequent calculations.

Next, month 3 added. There are two new rough elementary indices that are added, associated with the arcs b and c (the green arcs), i.e. y_b and y_c. They are associated with rough elementary indices due to the comparison of month 2 and 3 (arc b) and month 1 and 3 (arc c). The rough elementary indices associated with these arcs, i.e. $y_b = 0.149176924$ and $y_c = 0.130465037$ are the only values that can be adjusted at this step. The index associated with arc a is fixed in the calculations. There is only one cycle in the PIDG at this point. As adjusted values we have $\hat{x}_b^* = 0.124553358$ and $\hat{x}_c^* = 0.376700702$. Note that the first of these values differs not so much from the original value y_b, but the second one does deviate a lot more from its original value, y_c. This is because we consider the value of the arc b to be more reliable than that of arc c, so this is less adjusted.

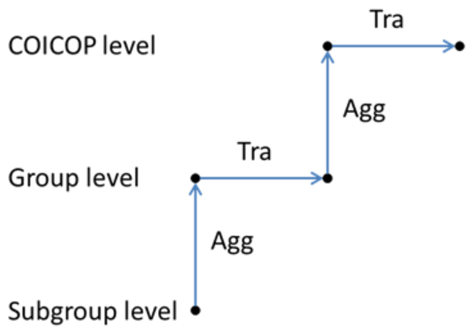
Finally, month 4 is added. Three new arcs, namely d, e and f and the corresponding values $y_d = 0.055268825$, $y_e = 0.238375362$ and $y_f = 0.126974666$ are added, all rough price indices calculated with the subgroup method. For the extended PIDG we have two elementary cycles (see the example on the simultaneous method in Section 2.4), namely bde^{-1} and $abdf^{-1}$ where e^{-1}, f^{-1} denote the reversed arcs of e, f, respectively. These newly added values to the new PIDG are the only ones that can be adjusted in this step. The previously adjusted log-price indices associated with arcs a, b and c are fixed. We now find the adjusted values: $\hat{x}_d^* = 0.047099759$, $\hat{x}_e^* = 0.171653117$ and $\hat{x}_f^* = 0.423800461$. They complete the PIDG that we have been building in three steps, without revising any previous results. Note the revisions that have been made. The one for arc f is substantial, as it is a ‘long arc’, connecting periods that are three months apart. ■

4. Aggregation and adjusting

The method described so far suggests that the transitivizing should be applied at the group level. However, if one is only interested in transitive indices at the aggregated, i.e. COICOP, level, there is no need to produce transitive indices at the group level as well. One can aggregate, in principle, from the subgroup level upwards, and transitivize at the top level only. This is computationally more efficient. An interesting point for investigation is to see how the results differ at the top (COICOP) level, when transitivizing at the top level only, and at the top level and at the group level.

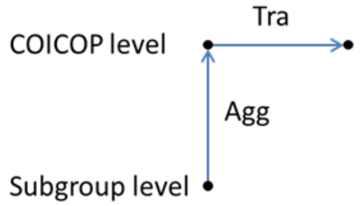
Figure 4.1.1 shows the two-step procedure of first aggregating from subgroup to group level, transitivizing at group level, aggregating the transitive indices to COICOP level (using proxy weights derived from turnover) and then transitivizing this result.

4.1.1 Aggregating and transitivizing via an intermediate level.



In Figure 4.1.2 the intermediate transitivizing step at the group level is discarded. There is an aggregating step from the subgroup to the group level (without weights) and these rough indices are then aggregated to the COICOP level (using weights). The resulting rough indices at the COICOP level are then made transitive, using the same kind of methods as described before for the group level.

4.1.2 Aggregating and transitivizing without an intermediate step.



5. Building a spanning tree

In the present section we have collected various methods to compute the values for the minimum spanning tree. We assume to be operating under two conditions. The first is that a spanning tree is built stepwise, extending it every month. The second is that previously published price indices cannot be changed.

We take the spanning tree as the one corresponding to month-on month price indices. Most of the examples have been presented before in the present report. To avoid an abstract and general discussion we use the same input data as considered in the example on the simultaneous method in Section 2.4, the example on the incremental method with revisions in Section 3.1 and the example on the incremental method without revisions in Section 3.2. The method is more general and is also applicable to different time periods, say of one year. It can be applied to fixed year and rolling year approaches.

For graph related concepts like PIDG, spanning tree and transitive closure, see Appendix D.

5.1 Goal of the method

Before we consider the various methods that are possible, we first describe the general goal of the method of the present section. The idea is to build, step-by-step, a spanning tree. This in turn is used to generate a complete PIDG, which is by definition transitive. When building a spanning tree in this way, one only needs to worry, at each step, about the last arc that is added.

With such a spanning tree available, its transitive closure can be calculated. It determines the PIDG uniquely. It is also a compact, in fact minimal, representation of a transitive PIDG, with transitive closure as the method to expand the tree to the full PIDG.

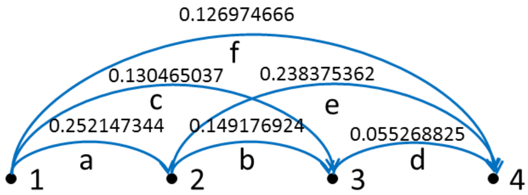
The fact that previously published price indices cannot be changed does not prevent us from making calculations - behind the curtain, so to speak - without this restriction. This allows us to compare results produced with and without restrictions concerning revisability of earlier results. And we can use the unrestricted results to compute a new addition to the spanning tree.

In fact, our assumption of transitivity includes what is usually called time reversibility in the price index literature.

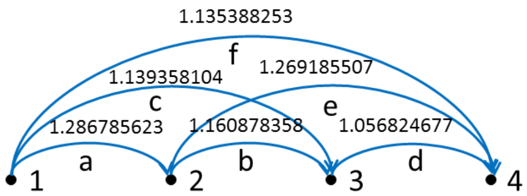
5.2 Rough indices

Rough indices were introduced in Willenborg (2017a). They provide the input for the adjustment methods discussed before, that is, the simultaneous method and the incremental methods. In Figure 5.2.1 we have collected the (natural) logarithms of the rough price indices that have been obtained by aggregating from the subgroups in the example on the simultaneous method in Section 2.4. The corresponding price indices are presented in Figure 5.2.2.

5.2.1 Log-price indices for the method that aggregates from the subgroups without making adjustments.



5.2.2 Price indices corresponding to the log-price indices in Figure 5.2.1.



The (log-)price indices are not transitive, as we know from the examples given before. For instance $y_c = 0.130465037 \neq 0.252147344 + 0.149176924 = y_a + y_b$. Because the (log-)price indices are not transitive it is not possible to generate them from a spanning tree, such as the one with arcs a, b, d.

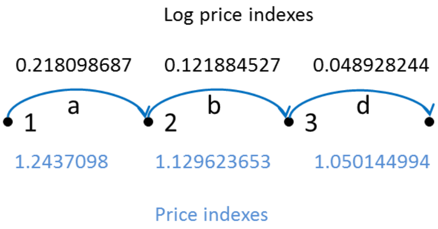
It should be noted that the rough indices associated with the arcs a, b, d alone can be used to build a spanning tree. By taking its transitive closure we can calculate the corresponding full PIDG. Although this is a transitive set of price indices, the information in the spanning tree is only a fraction of the data of that in the full PIDG. The information in other arcs (pairs of months) is not used to correct the information on the price indices associated with the arcs, a, b and d. The methods that we discuss in the sequel of the present section do use this extra information to correct the values associated with the arcs a, b and d.

5.3 Simultaneous method

This method (explained in the example on the simultaneous method in Section 2.4) can only be applied if all the data for an entire period are available. The input values are the aggregates of the elementary indices, for all arcs in the PIDG one wants to apply. The method then produces in one go adjustments for all values in the PIDG. This method in itself is fine, but it is not in line with the desire to publish results as soon as possible, that is when prices of a new month become available. However, the simultaneous method can be applied in case of the incremental method with revisions and ‘behind the screen’. The method can in fact be applied to periods of any length. It can be applied in our incremental setting at the end of each month.

In Figure 5.3.1 the calculated log-price indices for the simultaneous method used in the example on the simultaneous method in Section 2.4 are shown, as well as the price indices calculated from these.

5.3.1 Minimum spanning tree with price indices and log-price indices for the simultaneous method.



The log-price indices for the other arcs can be generated from these by transitive closure. We have: $\hat{x}_c = \hat{x}_a + \hat{x}_b$, $\hat{x}_e = \hat{x}_b + \hat{x}_d$ and $\hat{x}_f = \hat{x}_a + \hat{x}_b + \hat{x}_d$.

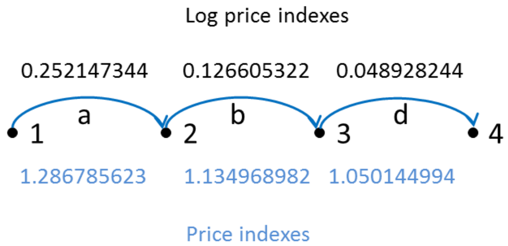
5.4 Building a nonrevisable spanning tree, I

The method discussed here is that used in the example on an incremental method with revisions in Section 3.1. The simultaneous method of the previous section can be used to build a spanning tree step-by-step, where previously calculated values in the spanning tree are not revised. This can be done as follows. Suppose we have calculated a spanning tree for the period starting at month 1 and ending at month m. At the end of a new month $m+1$ we use the simultaneous method to all logarithms of the rough price indices we have collected so far and we get transitivized versions. We then use only the adjusted value associated with the MoM-pair $(m, m+1)$ to extend the current spanning tree, so that it now applies to the period from month 1 to month $m+1$. The fact that the arcs in the adjusted simultaneous PIDG applying to this period probably had different values for the arcs $(1,2), (2,3), \dots, (m-1, m)$ than the spanning tree for the period 1 to m, is irrelevant. In fact we can associate with the new arc $(m-1, m)$ any value. It will never be in conflict with the values associated with previous MoM-pairs. Also the newly added value does not interfere with previously calculated values for the successive spanning trees, either directly calculated or indirectly calculated (through transitive closure).

This method is easy to apply, and attractive for its simplicity. It is recommended that the full PIDG calculated without restrictions ('behind the screen') should not be far off from the full PIDG calculated from the one derived from the appropriate spanning tree. A drawback of this method is that it may be quite different. In that case, the method itself does not offer a way to improve the situation. For consolation: other methods exist that offer the possibility to make corrections.

In Figure 5.4.1 the results, log-price indices and price indices, are shown if this method is applied to our small example. The values for the remaining arcs can be calculated from those for arcs a, b and d by transitive closure.

5.4.1 Minimum spanning tree with price indices and the log-price indices for the incremental method with revisions.



5.5 Building a nonrevisable spanning tree, II

The method used in this section is discussed in the example on the incremental method without revisions in Section 3.2. This method builds, incrementally, a PIDG. Price indices that have been calculated before cannot be altered later. There are no 'behind the screen' calculations. In this sense the method is different from the method discussed in the previous section. So this method can only modify the price indices associated with the most recently added arcs. If month $m+1$ is the most recent month, only the values of the arcs that end in $m+1$ are added, that is, the price indices associated with the arcs $(1, m+1), \dots, (m, m+1)$ are the only ones that can be adjusted.

5.5.1 Spanning tree with price indices and log-price indices for the incremental method without revisions.

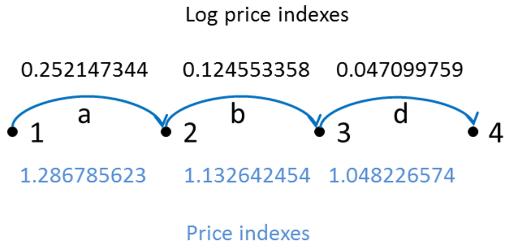


Figure 5.5.1 only shows the values of the optimum spanning tree, as a generator of the entire PIDG by using transitive closure.

5.6 Comparison of the methods

Of all the methods presented above, the one in Section 5.2 is the easiest to apply. One only needs to calculate the rough price indices at the group level, for the arcs connecting month i and month $i+1$. But this method does not exploit the additional information available in the data. For this reason it is not the most attractive approach, although it is the simplest one.

In contrast with this method, the simultaneous method (discussed in an example in Section 2.4) is the one that makes the most use of the available data (in a way), and would for that reason qualify as the best one possible of the methods discussed. The only drawback is that it is not an incremental method. But it can be used ‘behind the screen’ to build a nonrevisable spanning tree. Two such methods have been elaborated in the present report, namely in Sections 3.1 and 3.2.

In Table 5.6.1 the log-price indices for the simultaneous method (in Section 2.4), the 'incwirev' method (in Section 3.1) and the 'incworev' method (in Section 3.2) have been collected, for easy comparison. Added are the results of a method, called 'rough_tc' method, that uses the rough data for the arcs a, b and d to calculate the values for the arcs c, e and f by applying transitive closure.

5.6.1 Log-price indices for the various methods.

		method				
		rough	rough_tc	simultaneous	incwirev	incworev
arc	a	0.252147344	0.252147344	0.218098687	0.252147344	0.252147344
	b	0.149176924	0.149176924	0.121884527	0.126605322	0.124553358
	c	0.130465037	0.401324268	0.339983213	0.356181063	0.376700702
	d	0.055268825	0.055268825	0.048928244	0.048928244	0.047099759
	e	0.238375362	0.204445749	0.170812771	0.175533566	0.171653117
	f	0.126974666	0.456593093	0.388911457	0.405109307	0.423800461

In Table 5.6.2 the price indices, the original one, using the rough method, and the adjusted ones have been collected. The values are the exp-values of the corresponding figures in Table 5.6.1, but this time rounded to two decimal places.

5.6.2 Rounded price indices for the various.

		method				
		rough	rough_tc	simult.	incwirev	incworev
arc	a	1.29	1.29	1.24	1.29	1.29
	b	1.16	1.16	1.13	1.13	1.13
	c	1.14	1.50	1.40	1.43	1.46
	d	1.06	1.06	1.05	1.05	1.05
	e	1.27	1.23	1.19	1.19	1.19
	f	1.14	1.58	1.48	1.50	1.53

If we look at the data in Table 5.6.2, some phenomena should be noted. All methods yield the same values for arc a, except the simultaneous method and the incwirev method. These methods yield smaller values than the rough data. Also the price indices for the rough data are not transitive. If we look at the value for c for the rough data, it is (much) lower than the value computed from arcs a and b by transitive closure. For all methods, except ‘rough’, it is enough to give the values for the arcs a, b and d. The values associated with the remaining arcs can be computed by transitive closure. The results for the 'simultaneous' method and the two incremental methods are comparable. Those of the 'simultaneous' method tend to be a bit more 'dampened'. Those of the 'incworev' method tend to be more extreme. Those of the 'incwirev' are in between these methods.

But one should be careful with generalizing these findings, as the results only concern a single, small example. But in this example the 'simultaneous', 'incwirec' and 'incworev' produce results that are intuitive and that look promising.

In practice it would be advisable to use several of the methods simultaneously, to be able to compare deviations that may grow between revisable and nonrevisable price indices. For publication purposes, the incremental method with revisions is the most attractive one. But the values for the optimum tree the simultaneous method produces and the one that was built with the incremental method with revisions offer interesting comparisons. Also the method that calculates the values for the optimum spanning tree, without any corrections (‘rough_tc’), offers interesting material for comparison. So these three methods should be used to make comparisons. The incremental method without revisions could be discarded. It is both less interesting than any of the two other adjustment methods and it also requires extra effort to calculate its results. So applying this method seems to be a waste of time. It is included here to show another option, that at first sight seemed reasonable but on second sight does not offer any advantages.

6. Summary and discussion

Transitive price indices have the attractive property that they are free of chain drift. So for a transitive index there is no possibility of a, possibly big, discrepancy between chained indices and the corresponding direct indices. In the opinion of the author transitivity should be a requirement for any decent price index, not just an option. There are many transitive price indices. If a (proto) price index does not have that property, one can transitivize it, and use the resulting price index instead. The present paper discusses one such method to transitivize nontransitive price indices, namely the cycle method. This method can be viewed as a generalization of the GEKS method.

The present report is based on three main ideas:

- The subgroup method is used to find rough elementary price indices at the subgroup level, that can in turn be aggregated to price indices at the group level. (This is explained in Willenborg, 2017a).
- The rough indices are transitivized. Certain constraints apply. Various models can cope with this. The indices should be calculated incrementally. Published indices cannot be revised.
- Spanning trees are used in two ways: in the transitivization of price indices and to represent the transitivized indices succinctly. A spanning tree with indices associated with its arcs can be expanded into a complete PIDG, using transitive closure. This PIDG is by definition transitive.

It would be possible – and in fact an easy way to avoid inconsistencies – to calculate only MoM chain indices, using the aggregated price indices at the group level. In this way one simply does not make manifest that there may be ‘intransitivities’ when price indices for more distant pairs of months are calculated by aggregation. It yields a spanning tree, and the transitive closure can be taken. The resulting PIDG is transitive. This is an easy way to obtain transitive price indices, but also one that does not use all the information that is available. If one decides to use also the price indices calculated for months that are not adjacent, one is likely to find that the resulting PIDG is intransitive. But a method can be found to make these rough indices transitive. Application of this method forces these indices also to comply to closely related tests for price indices, such as the identity test and the time reversal test.

The methods described here have not been applied before (to my knowledge). Although the method was already described in Willenborg (2010), which essentially was an adaptation to price indices of a general adjustment method described in Willenborg (1993). In the present report only some small numerical examples have been given specifically to demonstrate the method using some numerical results. But the method has not otherwise been tested. The various methods (and some others) should be applied using real data.

There are various parameters in the methods discussed in the present paper that can be modified. It starts already with the calculation of the rough price indices. As Willenborg (2017a) indicates there are various ways to make price comparisons, ignoring the overlap in articles or not, when comparing the average prices of a group of articles in a pair of months. Also one can investigate for what arcs in the PIDG rough indices should be used as input for transitivization. Is it sensible to compare prices for a group of articles for months that are quite far apart? Or should one only consider months that are closer, such that there is a considerable overlap in articles. Should one use the same periods for all products, or should they depend on the type of product? And how, exactly? Should seasonal products have a longer period than nonseasonal ones? Another interesting question is how the weight matrices should be chosen.

A particular question applies to Section 4. This question is about the level at which the transitivization should take place, and what consequences it has when it is applied at various levels of aggregation. Should it be applied only at the COICOP level, or can it also be applied at an intermediate level? It is interesting to investigate how do the results differ? A question to explore is whether the transitivization can be done at different aggregation levels simultaneously, and not sequentially, as suggested in Section 4. It would be interesting to investigate this question with Paasche, Laspeyres, Fisher or Törnqvist index numbers, as none of them is transitive.

In fact the idea underlying the transitivization (‘path independence’) can also be applied to aggregation. Here it would mean that an index can be calculated in such a way that it does not matter how it was aggregated from elementary indices upwards. This is related to the idea underlying the consistency-in-aggregation test (cf. Balk, 2008, p.109).

References

Balk, B. M. (2008). Price and quantity index numbers, Cambridge University Press.

Gibbons, A. (1985). Algorithmic graph theory, Cambridge University Press.

Willenborg, L. (1993). An adjustment method based on graph homology. Report, Statistics Netherlands, Voorburg.

Willenborg, L. (2010). Chain indexes and path independence. Report, CBS, The Hague.

Willenborg, L. (2017a). Elementary price indices for internet data. Discussion paper, CBS, The Hague.

Willenborg, L. (2017b). From GEKS to cycle method. Discussion paper, CBS, The Hague.

Appendix A. Details of an example about the simultaneous method

The current appendix has the details of the example on the simultaneous method in Section 2.4.

We start with a table with average monthly prices for a four months period, which presents the (artificial) case of a group of articles with seven subgroups and for a four month period. In this case it is not indicated how there average monthly prices have been calculated from the prices of the individual articles. That is of no importance for our example.

Average monthly prices for seven subgroups and four months.

		month			
		1	**2**	**3**	**4**
subgroup	**1**	3	5	4	4
	2	10	-	12	-
	3	20	25	-	-
	4	44	45	44	46
	5	100	-	110	105
	6	44	-	48	-
	7	-	4	8	10

From the average prices in this table we can calculate all possible price ratios, or rough elementary price indices, for each subgroup. The results are presented in the next table. Each ordered pair of months corresponds to an arc in the PIDG in Figure 2.4.1.

Rough elementary price indices for all ordered pairs of months (arcs) in the four month period.

		arc					
		a=(1,2)	b=(2,3)	c=(1,3)	d=(3,4)	e=(2,4)	f=(1,4)
subgroup	1	1.666666667	0.8	1.333333333	1	0.8	1.333333333
	2	-	-	1.2	-	-	-
	3	1.25	-	-	-	-	-
	4	1.022727273	0.977777778	1	1.045454545	1.022222222	1.045454545
	5	-	-	1.1	0.954545455	-	1.05
	6	-	-	1.090909091	-	-	-
	7	-	2	-	1.25	2.5	-

As we want to use geometric means to average the price indices in this table, we calculate the natural logarithm of the figures in this table and average them. The results are collected in the next table. It should be noted that our adjustments are applied to the natural logarithms of the rough elementary price indices at the group level, to which we will refer to as the logarithms of the rough price indices. These are given in the second last row; they form the y'-vector (the transpose of the y-vector). The last row labelled 'exp' contains the exponents (e^{y_i}) of the components of the y-vector; these values are the logarithms of the rough price indices. Note that the components of y are not transitive. For instance, the value associated with arc c (=0.130465037) is not equal to that associated with arcs a and b added together (=0.252147344 + 0.149176924= 0.401324268). In fact the deviations (as in the case of arcs c, a and b) can be quite substantial.

Natural logarithms of the figures in the previous table, and some other quantities.

		arc					
		a=(1,2)	**b=(2,3)**	**c=(1,3)**	**d=(1,4)**	**e=(2,4)**	**f=(1,4)**
subgroup	**1**	0.510825624	-0.223143551	0.287682072	0	-0.223143551	0.287682072
	2	-	-	0.182321557	-	-	-
	3	0.223143551	-	-	-	-	-
	4	0.022472856	-0.022472856	0	0.044451763	0.021978907	0.044451763
	5	-	-	0.09531018	-0.046520016	-	0.048790164
	6	-	-	0.087011377	-	-	-
	7	-	0.693147181	-	0.223143551	0.916290732	-
	sum	0.756442031	0.447530773	0.652325186	0.221075298	0.715126087	0.380923999
	n	3	3	5	4	3	3
	sum/n (y)	0.252147344	0.149176924	0.130465037	0.055268825	0.238375362	0.126974666
	exp	1.286785623	1.160878359	1.139358104	1.056824677	1.269185508	1.135388254

For the price index graph of this problem (see Figure 2.4.1) we have a cycle matrix with three elementary cycles. If we take the spanning tree as the tree consisting of the vertices 1, 2, 3 and 4 and arcs a, b and d, we obtain the cycle matrix in the next table. This spanning tree is the more natural one to take, as it contains the arcs that are most important. In terms of the weights associated with the arcs (see below) it is also a (and very often the only) minimum weight spanning tree, where the sum of the weights associated with its arcs is minimal among all spanning trees for the PIDG we are considering.

Data for the cycle matrix, and how they are obtained by successively adding arcs not part of the spanning tree (c, e, f), to the spanning tree (with arcs a, b, d).

		arc					
action	**elem. cycle**	**a**	**b**	**c**	**d**	**e**	**f**
add c	**C1**	1	1	-1	0	0	0
add e	**C2**	0	1	0	1	-1	0
add f	**C3**	1	1	0	1	0	-1

The next table contains the weight matrix that we will use. Note that only the main diagonal is nonzero. The weights chosen in this table are such that the longer the arc, the greater the associated weight, implying that the easier it is to adjust the corresponding log-price index, that is, a component of y. There has been no experimentation in making a reasoned choice for these weights. This, in fact, is a research topic in its own right, as was remarked already in the main text.

The weight matrix.

		arc					
		a	**b**	**c**	**d**	**e**	**f**
arc	**a**	1	0	0	0	0	0
	b	0	1	0	0	0	0
	c	0	0	10	0	0	0
	d	0	0	0	1	0	0
	e	0	0	0	0	10	0
	f	0	0	0	0	0	20

Now we have all the ingredients for the calculation of $\hat{x}$, the adjusted y. The result is presented in the next table, containing both y and $\hat{x}$.

The original log-price index vector and its adjusted counterpart using the simultaneous method.

		Log-price indices	
		y	$\hat{x}$
arc	a	0.252147344	0.218098687
	b	0.149176924	0.121884527
	c	0.130465037	0.339983213
	d	0.055268825	0.048928244
	e	0.238375362	0.170812771
	f	0.126974666	0.388911457

Note that the components of $\hat{x}$ are transitive (except for possible rounding errors): the value associated with arc c equals those associated with a and b added. Likewise, the value associated with e equals that of those associated with b and d added. The value associated with f equals that associated with a and e added, and that associated with c and d added, as well as that associated with a, b and d, added.

The adjusted price indices can be calculated from $\hat{x}$, by exponentiating each component. The results are displayed in the following table.

The transitivized price index at the group level compared with the original rough price indices.

		$\hat{x}$	$e^{\hat{x}}$	rough indices
arc	a	0.218098687	1.243709799	1.286785623
	b	0.121884527	1.129623653	1.160878358
	c	0.339983213	1.404924006	1.139358104
	d	0.048928244	1.050144994	1.056824677
	e	0.170812771	1.186268624	1.269185507
	f	0.388911457	1.475373912	1.135388253

In this case the indices are transitive under multiplication (except for possible rounding errors). For instance, the adjusted index associated with arc c is equal to the indices associated with arcs a and b multiplied. Likewise for other arcs in the PIDG at hand.

This implies that we can use the adjusted values associated with the arcs of the spanning tree (with arcs a, b and d) to generate the values associated with the arcs c, e and f.

Comparing the original values with the adjusted ones, associated with the arcs a, b, d, we see that the adjustment resulted in a decrease of the original indices, and for the values associated with arcs c and f a big increase.

Appendix B. Details of an example on an incremental method with revisions

In the present appendix detailed information can be found on the calculations for the example about the use of the incremental method with revisions in Section 3.1. We use the same starting data as in Appendix A.

We start with adding month 2, and arc a. In this case we need to aggregate the log-price index from the subgroup level. The result was calculated in the example on the simultaneous method in Section 2.4 and yields $x_a^* = 0.252147344$. The asterisk denotes that this value is fixed for the optimum spanning tree for this method.

Now we look at the case where month 3 is added, which implies that we have to consider the PIDG with three arcs, viz a, b, c (cf. Figure 2.4.1). The other cases, i.e. adding month 2 and adding month 4 have been dealt with before, namely in Appendix A. The results are given in the table below.

The y-vector and how it is calculated for three vertices and three arcs.

		arc		
		a=(1,2)	b=(2,3)	c=(1,3)
subgroup	1	0.510825624	-0.223143551	0.287682072
	2	-	-	0.182321557
	3	0.223143551	-	-
	4	0.022472856	-0.022472856	0
	5	-	-	0.09531018
	6	-	-	0.087011377
	7	-	0.693147181	-
	sum	0.756442031	0.447530773	0.652325186
	n	3	3	5
	sum/n (y-vector)	0.252147344	0.149176924	0.130465037
	exp	1.286785623	1.160878359	1.139358104

The PIDG we now consider has one cycle. A cycle matrix is:

$$C = \begin{pmatrix} 1 & 1 & -1 \end{pmatrix}.$$

As a weight matrix we take

$$W = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 10 \end{pmatrix},$$

where the rows and columns are supposed to be labelled by arcs a, b, c, respectively. It is a submatrix of the weight matrix for the simultaneous problem (cf. Appendix A).

By applying formula 2.1 we obtain the Π-matrix

$$\Pi = \begin{pmatrix} 0.916666667 & -0.083333333 & 0.083333333 \\ -0.083333333 & 0.916666667 & 0.083333333 \\ 0.833333333 & 0.833333333 & 0.166666667 \end{pmatrix},$$

so that

$$\hat{x} = \begin{pmatrix} \hat{x}_a \\ \hat{x}_b \\ \hat{x}_c \end{pmatrix} = \Pi y = \begin{pmatrix} 0.229575741 \\ 0.126605322 \\ 0.356181063 \end{pmatrix}.$$

Note that $\hat{x}_c = \hat{x}_a + \hat{x}_b$.

From this step we retain the value for arc b in the spanning tree that we are building. We have: $\hat{x}_b^* = 0.126605322$.

Next, month 4 is added. We are now in the same situation as in Appendix A, and we can use the solution obtained there. We can look up the value for arc d, i.e. $\hat{x}_d^* = 0.048928244$.

Appendix C. Details of an example on an incremental method without revisions

The example considered here on an incremental method without revisions, is discussed in Section 3.2. The basic data for this method are the same as in Appendices A and B, to which we will refer in order to avoid duplication.

The incremental method without revisions requires that no previously calculated values can ever be changed. We first show how these calculations would proceed if we keep certain values fixed. We do not use numerical values; we just go through the various steps. This method can certainly be implemented. But it requires extra effort. We wanted to stay within the framework of the other examples, so we used a good approximation. This was accomplished by taking a suitable weight matrix W.

'Exact' calculations

In step 1, y_a is the rough elementary log-price index calculated from the subgroup price indices. Since this value cannot be compared with any other value, it is the final one, denoted as x_a, in the incremental method. To indicate that it is fixed value we add an asterisk as a superscript and write $x_a^* = y_a$.

In step 2, two arcs are added, b and c, and two associated rough elementary log-price indices, viz y_b and y_c. There is one requirement that has to be satisfied by the adjusted counterparts, viz x_b and x_c:

$$x_a^* = x_c - x_b.$$

The adjustments in this steps yield fixed values x_b^* and x_c^*.

In step 3, three arcs are added, d, e and f, corresponding to rough elementary log-price indices y_d, y_e, y_f. The two constraints satisfied by the adjusted log-price indices x_d, x_e, x_f are:

$$x_b^* = x_e - x_d\ ,$$
$$x_a^* + x_b^* = x_f - x_d\ .$$

This yields the fixed adjusted log-price indices x_d^*, x_e^*, x_f^*. If the process would continue to add new arcs and corresponding rough elementary log-price indices, we would generate new linear constraints due to the cycle conditions.

In each of these steps conditional optimization problems would have to be solved: a quadratic function has to be minimized under the condition that a set of linear constraints of the type just discussed, have to be obeyed.

Approximate calculations

Month 2, 3 and 4 are successively added, as in the incremental method with revisions. But now once adjusted values will never be changed. If a new month is added, only the values associated with this month (as reporting month) can be modified to make the extended PIDG transitive.

We start the process with adding month 2. This yields arc a, and the value calculated directly from the subgroup data. This value is the same as calculated for example 3.2.1 (Appendix B contains the details). We have $x_a^* = 0.252147344$, where the asterisk indicates, as in Appendix B, that the value is fixed. This time not only for publication purposes, but also for subsequent calculations in the example.

In the next step month 3 is added, introducing two arcs, b and c, with associated values $y_b = 0.149176924$ and $y_c = 0.130465037$. These are the only values that can be changed in this step. This problem could be solved with a somewhat different method. However, we wish to find a solution to this problem staying within its framework. So we used a weight matrix which gives a very high penalty to changing the value at arc a, established in the previous step. In fact, the weight matrix we use is

$$W = \begin{pmatrix} 10^{-14} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 10 \end{pmatrix}$$

with a very small value for the value of arc a. The values of arcs b and c are allowed to change, and that of arc c the most. Of course, the cycle matrix in this case is the same as in the example on the incremental method without revisions in Section 3.2, namely

$$C = (1 \quad 1 \quad -1).$$

Employing formula 2.1 we obtain for the adjusted values for arcs b and c: $x_b^* = 0.124553358$ and $x_c^* = 0.376700702$. As before, the asterisks denote fixed values, but this time for both publication and subsequent calculations.

In the final step, month 4 and arcs d, e and f are added. Their initial values are $y_d = 0.047099759$, $y_e = 0.171653117$ and $y_f = 0.423800461$, as calculated from the subgroups (logarithms of the rough price indices). These are the only values that can changed. In this case there are three elementary cycles and a cycle matrix is as previously obtained, in the example on the simultaneous method in Section 2.4 and in the example on the incremental method with revisions in Section 3.1 (see the fourth table in Appendix A) :

$$C = \begin{pmatrix} 1 & 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & -1 & 0 \\ 1 & 1 & 0 & 1 & 0 & -1 \end{pmatrix}.$$

As a weight matrix we have used:

$$W = \begin{pmatrix} 10^{-14} & 0 & 0 & 0 & 0 & 0 \\ 0 & 10^{-14} & 0 & 0 & 0 & 0 \\ 0 & 0 & 10^{-14} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 & 0 & 20 \end{pmatrix}.$$

The values 10^{-14} are used to keep the values for the arcs a, b and c (practically) fixed. The values for arcs d, e and f are allowed to change, that of e more than that of d, and that of f more than that of e. Although the method is close to being useless (because of the large differences of order of magnitude of the entries in the matrix W) the computations yield a valid adjustment, and in particular adjusted values for the arcs d, e, and f: $\hat{x}_d^* = 0.062732954$, $\hat{x}_e^* = 0.190251427$ and $\hat{x}_f^* = 0.442398771$. Although there is no need to indicate that these values are fixed we have done so to stress their immutability in case the adjustment process should be continued.

In the following table the results obtained in the present section have been collected. For easy comparison we have also included the original rough price indices.

Adjusted price indices for the incremental, approximate method without revisions.

		$\hat{x}$	$e^{\hat{x}}$	rough indices
arc	**a**	0.252147344	1.286785623	1.286785623
	b	0.124553358	1.132642454	1.160878359
	c	0.376700702	1.457468026	1.177132383
	d	0.047099759	1.048226574	1.076474948
	e	0.171653117	1.18726592	1.269185508
	f	0.423800461	1.527756716	1.135388254

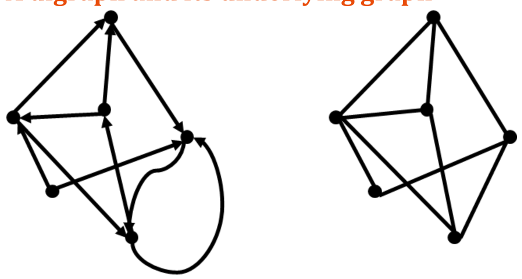
As one may expect, the adjusted price indices differ more from the rough indices than in case of the method in Appendix B, simply because there are less possibilities to compensate for bigger discrepancies. Note that the adjustments for the arcs a, b and d are smallest, as was the purpose of the method.

It should be borne in mind that the results of the cycle method depend on the choice of the weight matrix.

Appendix D. Some concepts from graph theory

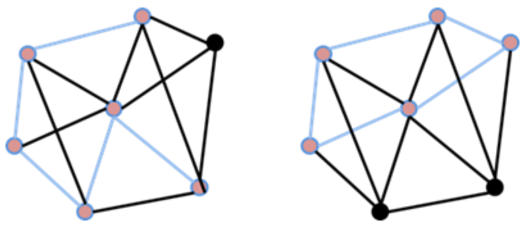
Price index information is conveniently presented using some basic concepts from graph theory. A graph is an abstract object consisting of dots and lines connecting them. The dots typically represent entities and the lines connecting them some sort of relationship that exists between these objects. These lines are usually called edges. There are also directed graphs or digraphs, for short. They consist also of dots, and also of lines connecting some of them, but this time the lines have a direction. They are usually called arcs, arrows or directed edges. A digraph is also used to express objects and relations between them. But this time the relations are asymmetric. Relations like 'bigger than', 'parent of', etc. In the context of the present paper we use both types of graphs. Digraphs are used to represent price index information. The (sub)group-month combinations pertaining to clothing articles are represented by the dots (also referred to as points or vertices). We need a digraph and not a graph to represent this price index information, because there is an asymmetry between the base month and the comparison month. We call such a digraph a price index digraph, abbreviated as PIDG. Sometimes, we want to discard the direction of the arrows, and are just satisfied to know that there is a link between two entities, in our case months. In that case we replace the arcs in a PIDG by edges and the resulting graph is called a price index graph, PIG for short. It is the graph underlying a PIDG: it has the same set of points, and $\{a, b\}$ is an edge in the PIG if (a, b) or (b, a) is an arc in the PIDG.

A digraph and its underlying graph



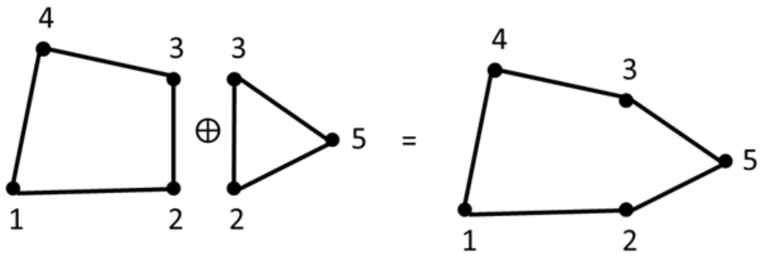
Given a PIDG (or a digraph, more generally) in order to apply the method in the present report we have to determine a full set of elementary cycles. A cycle in a digraph, or rather in the underlying graph, is a closed path in the graph $G = (V, E)$, where V denotes the set of points and E the set of arcs (or edges) of G. A path is a sequence of points $(v_1, \dots, v_k)$ with $v_i \in V$, such that $\{v_i, v_{i+1}\} \in E$ for $i = 1, \dots, k-1$. Then k is called the length of the path. In case $v_1 = v_k$the path is closed, and is called a cycle. A connected graph without cycles is called a tree.

A path (left) and a cycle (right), defined by the blue edges.



Elementary cycles in a graph are like a basis in a finite dimensional vector space: with them every cycle in the graph can be generated, by an addition operation (but not in $\mathbb{R}$ or $\mathbb{C}$, but in F_2, the finite field, with only two elements, namely 0 and 1). In a finite dimensional vector space, such as $\mathbb{R}^n$ each vector can be written as a linear combination of basis vectors, which are vectors in a basis. Likewise each graph can be written as sum (the ring sum) of elementary cycles. As in vector spaces, a basis in a cycle space is not necessarily unique. The ring sum is illustrated in the following figure.

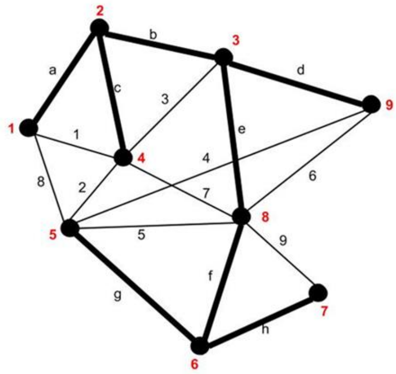
Adding two cycles using the ring sum



A set of elementary cycles for a graph G can be generated with the help of a spanning tree of G. This is subgraph of G, in this case with an arc set that is a subset of E, which is a tree, that is, contains no cycles (it is a tree!), and is complete, meaning that its set of points is V, equal to that of G. A spanning tree for G is fairly easy to compute (using a so-called greedy algorithm). See e.g. Gibbons (1985) for various algorithms involving spanning trees.

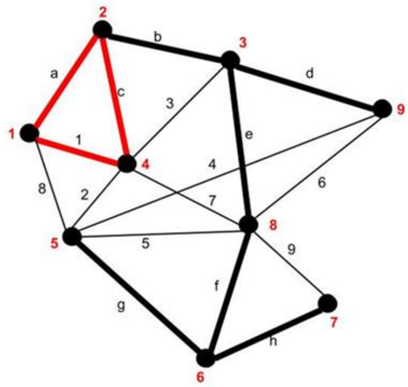
The next figure is an example of a graph together with a spanning tree. The edges in the spanning tree are bold, and they are labelled with letters from a to h. The edges not part of the spanning tree are labelled from 1 to 9. Each such edge corresponds to an elementary cycle.

Graph with a spanning tree

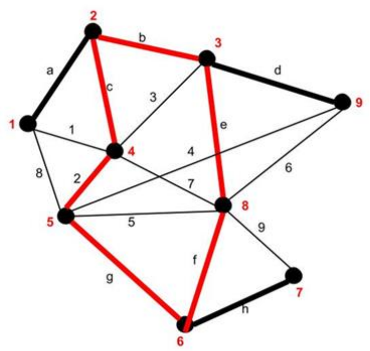


Two examples of such elementary cycles, corresponding to edges 1 and 2, are presented in the following two figures. Together with 7 other such cycles they form a basis of the cycle space of the graph in the next figure.

Elementary cycle associated with edge 1 of the spanning tree in the previous figure.

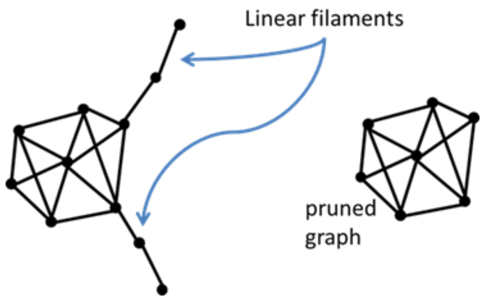


Elementary cycle associated with edge 2 of the spanning tree in the spanning tree in the fourth figure in the present appendix.



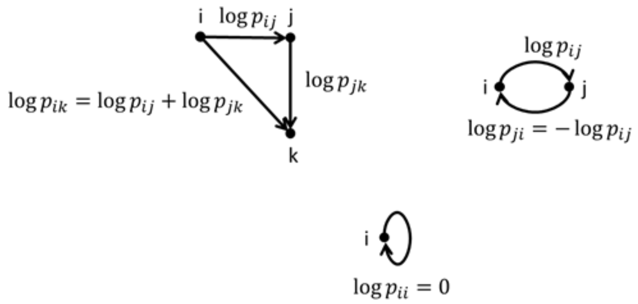
Once we have a calculated a spanning tree S for G, we can calculate a basis of elementary cycles as follows. If G and S are the same, there are no cycles and the basis of elementary cycles is empty. If G contains edges that are not in S, every such edge generates exactly one elementary cycle when it is added to S, eliminating ('pruning') any linear filaments, that is, all arcs (or edges) that are not part of a cycle. The next figure illustrates this.

Pruning of a graph with linear filaments



Spanning trees are also handy and compact objects to store the information of a transitive PIDG. For such a PIDG holds that if (i, j) and (j, k) are arcs with associated price indices p_{ij} and p_{jk} so for the arc (i, k) a price index exists equal to $p_{ij} = p_{ik}p_{kj}$. Here we have used the multiplicative form for price indices. By taking (natural logarithms) we get the additive form for log-price indices: $\log p_{ij} = \log p_{ik} + \log p_{kj}$. Furthermore it is assumed that $p_{ii} = 1$ holds (or in additive form $\log p_{ii} = 0$), for all $i \in V$ and $p_{ij}p_{ji} = 1$ for all $i, j \in V$ (in additive form $\log p_{ij} + \log p_{ji} = 0$). The mechanism employed to calculate price indices associated with 'implied arcs' can also be used to generate price index values for such arcs. The process to calculate such values for all possible implied arcs is called transitive closure.

Transitivity, reversibility and identity graphically represented



A particular and important spanning tree for a PIDG is the one corresponding with the MoM price indices (see for instance Figure 2.2.1).

Explanation of symbols

Empty cell	Figure not applicable
.	Figure is unknown, insufficiently reliable or confidential
*	Provisional figure
**	Revised provisional figure
2014–2015	2014 to 2015 inclusive
2014/2015	Average for 2014 to 2015 inclusive
2014/’15	Crop year, financial year, school year, etc., beginning in 2014 and ending in 2015
2012/’13–2014/’15	Crop year, financial year, etc., 2012/’13 to 2014/’15 inclusive

Due to rounding, some totals may not correspond to the sum of the separate figures.

Colofon

Publisher
Statistics Netherlands
Henri Faasdreef 312, 2492 JP The Hague
www.cbs.nl

Prepress
Statistics Netherlands, Studio BCO

Design
Edenspiekermann

Information
Telephone +31 88 570 70 70, fax +31 70 337 59 94
Via contactform: www.cbsl.nl/information