Hans Ravn

The OptiFlow Model Structure

Note 2017:
The OptiFlow model was originally developed under the name OptiWaste in the now finished TopWaste project.

The OptiFlow model is further developed, applied and distributed from www.balmorel.com.

PRELIMINARY - The document is in transition from OptiWaste to OptiFlow terminology and content

Version 0.99, September 2017

This document revised 2017.09.09

PRELIMINARY - The document is in transition from OptiWaste to OptiFlow terminology and content

Contents

1 Introduction

The OptiFlow model was originally developed and distributed as part of the TopWaste project (2011 through 2015) under the name OptiWaste.

The present document contains two main parts. The first part is a gentle introduction to the main ideas and mechanisms of the OptiFlow model (Section 2). The second part is a technical documentation (Section 3).

1.1 Purpose of OptiFlow

The objectives of the OptiFlow model is to perform analyses that may provide better insight and decisions relative to improved use of waste for energy and/or material recycling integrating economic, environmental and resource scarcity considerations.

1.2 General aspects of OptiFlow

The OptiFlow model is made with representation of the following main aspects.

General:

- waste as resources for energy and materials
- economy and environment in relation to this
- combining aspects of waste life cycle analysis and energy system analysis

Geography:

- a (typically) national waste system in an international setting
- with linkages to an energy system
- national and international geographical information, including location and transport

Time:

- time aspects within the year to handle operations considerations
- time aspects over longer time spans (years) to handle information changes and investments over time

Technologies:

- for waste and energy handling and transformation
- with possibilities for representing existing technologies and investments in new ones

Implementation:

- optimization model
- may optionally be integrated with the energy system model Balmorel to have a more extensive energy system representation
- implemented in GAMS (General Algebraic Modeling System, www.gams.com)

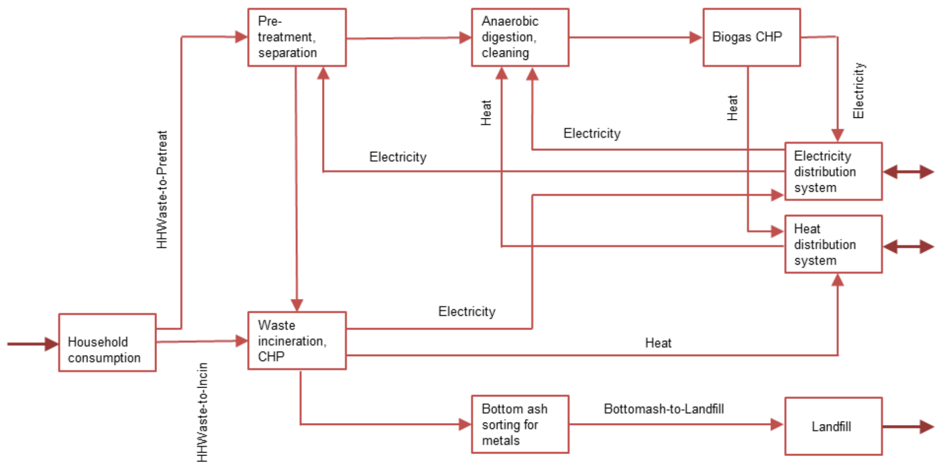

Figure 1: Example of a network.

1.3 Availability of OptiFlow

The GAMS code for OptiFlow will be available as open source after project termination.

2 A gentle non-technical introduction to OptiFlow

In this Section a gentle introduction to the main ideas and mechanisms of the OptiFlow model will be given. Technical details will be kept at a minimum. In particular, implementation details are not covered, details are given in Section 3. To keep the text simple, the scope of the text is narrower than covered by OptiFlow, cf. Section 2.10.

2.1 Network model - Processes, Flows and their relations

Network models in general

The OptiFlow is mainly formulated as a network model. Network models are important because a large number of problems may be formulated as network models and analyzed through suitable concepts, properties and algorithms. Network models come in various kinds, the OptiFlow is what is typically termed a generalized network model.

Background material on networks is indicated in Section 2.11 page 18.

Figure 1 shows an example of a network. It consists of a number of Nodes (rectangles) and Arcs (arrows). The Arcs in the example are directed as shown by the arrow heads, i.e., an Arc goes from one Node to another Node in a one-direction fashion. The Nodes, Arcs and their directed connections will be called the Topology of the network.

OptiFlow: Flows, Processes and how they relate

The topology of the waste and energy network is defined by Nodes and Arc as above above. In the OptiFlow terminology Processes and Flows (Proc and Flow for shorts) correspond to Nodes and Arcs. Therefore initially Processes, Flows and their basic relations are presented here.

PROC - Processes

The Processes in OptiFlow correspond to nodes in a traditional network model.

PROC is a set containing a number of elements, each element is a Proc.

A Proc represents some kind of relations between the Flow connected to the Proc, cf. below.

The Proc will together with the Flow, see next, constitute a Topology as illustrated in Figure 1. In such Topology the Proc are all unique, i.e., an element from set PROC can appear at most once (in every Area, cf. Section 2.4).

FLOW - Flows

The Flows in OptiFlow correspond to arcs in a traditional network model.

FLOW is a set of Flows. Flows are used to represent streams of material, energy, money or other types. A Flow has a certain Quality, i.e., a specific composition of material, energy, money etc., cf. Section 2.2 page 11.

An element Flow from set FLOW can appear zero or more times in a Topology as illustrated in Figure 1. This is in contrast to the Proc that are all unique (in every Area).

Hence, to uniquely refer to a Flow in the Topology it is necessary to specify the two Proc it connects i.e., the fromProc and the toProc (with Area information, cf. Section 2.4).

Hence, it is required to distinguish the following meanings of 'flow':

- FLOW: a set of elements
- Flow: an element in FLOW
- Flow: a specific Flow stream linking two specific Proc in a specific direction, cf. triplet below
- the numerical value of a specific Flow stream

Hopefully it will be clear from the context what is meant.

FLOWFROMTOPROC - Topology of Processes and Flows

The linking between the Flow and two Proc is given in FLOWFROMTOPROC, which also defines the direction of the Flow. An element in FLOW may in FLOWFROMTOPROC be defined to link more than one pair of Proc, as mentioned.

This relation therefore for any Flow specifies the two Proc that it relates, given as the triplet (fromProc, toProc, Flow) where fromProc, and toProc are two specific elements in the set PROC of Processes. The sequence of the two Processes dictates the direction of the Flow, namely from the first element in the triplet (fromProc) to the second element (toProc). See Figure 2. The three set PROC, FLOW and FLOWFROMTOPROC define the Topology of the network.

By definition the numerical value of any Flow stream between two Processes (an intermediate Flow stream) will be non-negative in the direction from fromProc to toProc.

Further elements in the definition of a specific Flow are indices for geography (Section 2.4 page 14) and time (Section 2.5 page 15) (hence not triplets but rather quintuples).

PROCINOUTFLOW - Relations of Flows at a Process

FLOWFROMTOPROC and PROCINOUTFLOW are the two most important elements relating PROC and FLOW in the OptiFlow model. While the three sets above define the Topology

Figure 2: Example relations for a Transform Proc, Eqn. (1) or (2).

of the network, PROCINOUTFLOW relates numerical values.

The parameter PROCINOUTFLOW specifies for a given Proc the relationship between inFlow and outFlow, where inFlow is one or possibly more elements from the set FLOW of Flows, and outFlow likewise is one or possibly more elements from the set FLOW of Flows. It therefore links elements in the triplet (Proc, inFlow, outFlow).

The following are examples of the main three forms that the linking may attain, representing respectively one-to-one, one-to-many and many-to-one relationships,

- Transform
- Variable Split (or VSplit for short)
- Variable Join (or VJoin for short)

Parameter PROCINOUTFLOW holds two kinds of information on any Flow relationship. One is qualitative and indicates which of the above three relation forms is used. The other is numerical and indicates the coefficient in the relation (indicated by $c^*_{*,*}$ in the following examples).

In the following some examples will be given. The set PROC is defined as $\{P^0, P^1, P^2, P^3, P^{41}, P^{42}, P^{43}, P^5, P^{61}, P^{62}, P^{63}, P^{71}, P^{72}, P^{73}, P^8\}$, from which some Proc are used in each example. Set FLOW is defined as $\{F_1, F_2, F_3, F_{41}, F_{42}, F_5, F_6, F_7\}$, where some of the Flow are used in each example.

Example: Transform

Consider the network snippet in Figure 2. The Transform around the Proc P^0 may be assumed to be described by the following equation which relates the two Flow F_1 and F_2. Here and in the following $c^*_{*,*}$ is a coefficient with fixed value.

$$F_2 = c^0_{1,2} F_1 \tag{1}$$

The Flow F_1 and F_2 in (1) have the same symbols as on Figure 2. They are understood to be specific Flow from set FLOW, more precisely the Flow F_1 and F_2. However, according to the above explanation, a specific Flow between two Proc is to be identified by a triplet (cf. page 5). Hence, F_1 is specified by the triplet (P^1,P^0,F_1). It will also in the following be named $F_1^{1,0}$ where the upper indexes correspond to the Proc indexes and their sequence reflects the (fromProc, toProc) sequence. Similarly, F_2 is given by the triplet (P^0,P^2,F_2) or the name $F_2^{0,2}$. This is reflected in the following version of the equation,

Note that in the Transform Proc (example equations (1) - (6)) the parameter values $c^0_{*,*}$ are multiplied on the inFlow.

$$F_2^{0,2} = c^0_{1,2} F_1^{1,0} \tag{2}$$

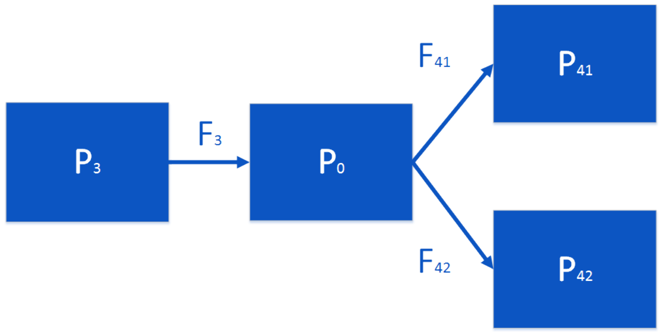

Figure 3: Example relations for a Transform Proc, Eqns. (3) - (4) or (5) - (6).

Example: Transform

Transform around the Proc P^0 in the network snippet in Figure 3 may be assumed to be described by the following equation

$$F_{41}^{0,41} = c_{3,41}^{0} F_{3}^{3,0} \quad (3)$$

$$F_{42}^{0,42} = c_{3,42}^{0} F_{3}^{3,0} \quad (4)$$

Similar comments as in the above example apply with respect to the Flow $F_{41}^{0,41}$, $F_{42}^{0,42}$ and $F_{3}^{3,0}$ being instances of elements Flow from set FLOW.

Example: Transform

The above Figure 3 might also be used for illustrating another relation. Assume the relevant equations are

$$F_{41}^{0,41} = c_{3,41}^{0} F_{3}^{3,0} \quad (5)$$

$$F_{42}^{0,42} = c_{42,41}^{0} F_{41}^{0,41} \quad (6)$$

While in (3) - (4) and Figure 3 both outFlow relate to the single inFlow, in (5) - (6) one outFlow ($F_{41}^{0,41}$) relates to the inFlow while the other outFlow ($F_{42}^{0,42}$) relates to the second outFlow ($F_{41}^{0,41}$). It is also possible to relate two inFlow.

Example: Variable Split

Assuming that the network snippet in Figure 4 describes a VSplit Proc, the associated equation is

$$\begin{aligned} F_{5}^{5,0} &= c_{5,61}^{0} F_{6}^{0,61} \\ &+ c_{5,62}^{0} F_{6}^{0,62} \\ &+ c_{5,63}^{0} F_{6}^{0,63} \end{aligned} \quad (7)$$

or more generally

$$F_{5}^{5,0} = \sum_{j} c_{5,j}^{0} F_{6}^{0,j} \quad (8)$$

In this example Flow $F_{6}^{0,61}$, $F_{6}^{0,62}$ and $F_{6}^{0,63}$ represent triplets (cf. page 5) $(P^0, P^{61}, F_{6}^{0,61})$, $(P^0, P^{62}, F_{6}^{0,62})$ and $(P^0, P^{63}, F_{6}^{0,63})$, i.e., involving the same Flow element F_6 from set FLOW.

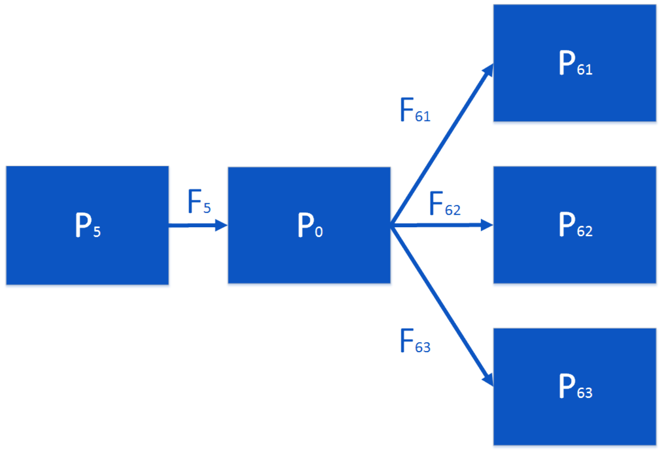

Figure 4: Example relations for a VSplit Proc, Eqns. (7) - (8)

In many cases all $c^0_{5,*}$ in equation (7) will have the value 1; recalling (page 5) that the numerical value of any intermediate Flow stream is non-negative, this situation is seen to represent the case where one inFlow, $F^{5,0}_5$, is divided among three outFlow, $F^{0,61}_6$, $F^{0,62}_6$ and $F^{0,63}_6$. An example where the $c^0_{5,*}$ are not 1 is when P^0 represents a division of energy streams. If $F^{5,0}_5$ is in units GJ/s (or GW) while $F^{0,61}_6$, $F^{0,62}_6$ and $F^{0,63}_6$ are in units MW then all $c^0_{5,*} = 1000$.

Note that in (7) and (8) the parameter values $c^0_{5,*}$ are multiplied on the outFlow, in contrast to the situation in the other types of relations (Transform and VJoin Proc), where $c^0_{*,*}$ are multiplied on the inFlow.

Example: Variable Join

The network in Figure 5 may illustrate summation of waste Flows from different areas, or summation of energy Flows from e.g. the electrical network and incineration based electricity generation. The example may illustrate e.g. mixture of household waste from various districts in a city. The VJoin Proc type applies only if the inFlows that are joined are of the same Quality. The equation is

$$\begin{aligned} F^{0,8}_8 &= c^0_{7,8}F^{71,0}_7 \\ &+ c^0_{72,8}F^{72,0}_7 \\ &+ c^0_{73,8}F^{73,0}_7 \end{aligned} \tag{9}$$

or more generally

$$F^{0,8}_8 = \sum_j c^0_{j,8}F^{j,0}_7 \tag{10}$$

In most cases all $c^0_{*,8}$ in equation (9) will have the value 1; this situation is seen to represent the case where three Flows, $F^{71,0}_7$, $F^{72,0}_7$ and $F^{73,0}_7$, are summarized directly to one Flow, $F^{0,8}_8$. As in the above example there are also cases where $c^0_{*,8}$ are not 1. An example could, as above, be when there is a change of units, e.g., if the inFlows are in units kg/hour, while the outFlow is in unit tonne/hour, then all have the value 0.001. If further there is assumed a 10% loss in a mixing Proc, the common value will be 0.0009.

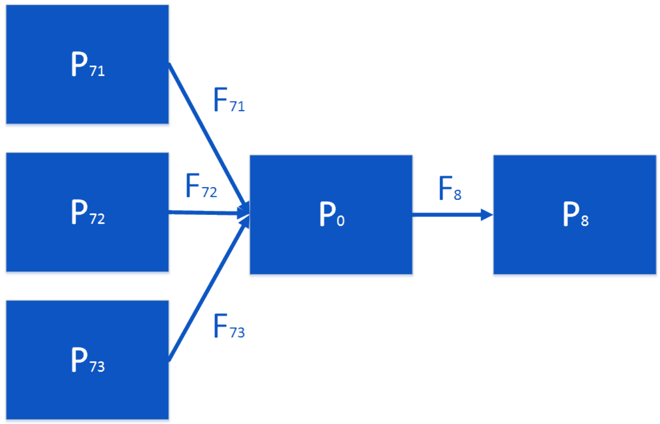

Figure 5: Example relations for a VJoin Proc, Eqns. (9) - (10)

Note that in (9) and (10) the parameter values $c^0_{*,8}$ are multiplied on the inFlow, as with the Transform Proc and unlike the VSplit Proc.

Flow Bundles and Proc Sharing

Suppose that the Proc P^0 in Figure 2 and P^0 in Figure 3 are somehow closely related in a specific network application, for instance by being located on the same property and operated by the same personnel; or maybe they are even parts of the same physically integrated activity. It might therefore seem natural to indicate this in the graphs in figures 2 and 3.

This is illustrated in Figure 6. The associated Transform equations are the same as (2) - (4).

The Flows appearing in any of the equations (2) and (3) - (4) are related through these equations. Let such a set of related Flows be called a Flow Bundle relative to Proc P^0. For instance, the Flow Bundle related to Figure 2 and equation (2) is $\{F^{1,0}_1,\ F^{0,2}_2\}$, and the Flow Bundle related to Figure 3 and equations (3) - (4) is $\{F^{3,0}_3,\ F^{0,41}_{41},\ F^{0,42}_{42}\}$.

The combination of Proc described is called Proc Sharing. Such Sharing applies not only to Transform Proc but to VSplit and VJoin as well, and also to mixtures of these three types. Note that actually a Proc with Bundling can therefore not necessarily be classified as being of one the three types, Transform, VJoin and VSplit; rather, this classification applies to Flow Bundles.

Proc Sharing (of the first two examples (2) and (3) - (4)):

$$F^{0,2}_2 = c^0_{1,2} F^{1,0}_1 \quad (11)$$
$$F^{0,41}_{41} = c^0_{3,41} F^{3,0}_3 \quad (12)$$
$$F^{0,42}_{42} = c^0_{3,42} F^{3,0}_3 \quad (13)$$

In order that the sharing of Proc shall work correctly, it must not be possible to mix up the Flow Bundles. Necessary and sufficient conditions for this are that no inFlow (element from FLOW) appears as inFlow in more than one of the Flow Bundles, and that no outFlow (element from FLOW) appears as outFlow in more than one of the Flow Bundles.

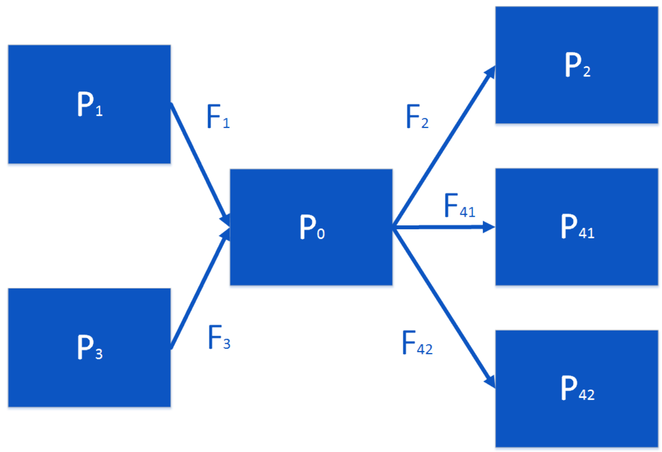

Figure 6: Flow Bundles and Proc Sharing.

Source, Sink, Buffer, Exim and other specific Proc types

The set PROC holds all Proc. However, set PROC may be subdivided into a number of subsets according to some general properties of the Proc in PROC. This classification is independent of the one used on page 5 (Transform, VSplit, VJoin). The meaning of the subsets may be illustrated with reference to Figure 1 page 4.

- Source - The Proc "Household consumption" has two outFlow but no inFlow. A Proc with only outFlow is called a Source Proc. The set of Source Proc is called PROCSOURCE. A Source Proc has an associated non-negative Flow (indicated in Figure 1). This Source Flow equals the sum of the outFlow.
- Sink - The Proc "Landfil" has an inFlow but no outFlow. A Proc with only inFlow is called a Sink Proc. The set of Sink Proc is called PROCSINK. A Sink Proc has an associated non-negative Flow (indicated in the Figure). This Sink Flow equals the sum of the inFlow.
- Buffer - The Proc "Electricity distribution system" and Proc "Heat distribution system" have both inFlow and outFlow. Such Proc are called Buffer Proc. The set of Buffer Proc is called PROCBUFFER. A Buffer Proc has an associated Flow, unrestricted in sign (not shown in the Figure). This Buffer Flow equals the sum of the inFlow minus the sum of the outFlow.
- Exim - There may in each Area (a geographical entity, Section 2.4 page 14) be one Proc which is the export/import gate for that Area, cf. Figure 8 page 14.
- Time - The time resolution within the year need not have equal importance for all parts of the modeled system. Hence, a dual time resolution may be applied (cf. Section 2.5 page 15 and PROC_T).
- Storages are implemented, see Section 2.6 page 16 for the set PROCSTORAGE.

Some further comments

Since the numerical value of any Flow stream between two intermediate Proc will be non-negative (as pointed out page 5) this forces both outFlows from Proc "Household consumption" on Figure 1 to be zero. To permit positive Flow values to leave a Source Proc a special non-negative Flow, FlowSource, is introduced to account for the balance. Hence, assuming that the Proc is of type VSplit and that both coefficients on the right hand side are 1 and using an informal notation referring to Figure 1,

$$\begin{aligned} \text{HHconsumptionSourceFlow} &= \text{HHWaste-to-Pretreat} \\ &+ \text{HHWaste-to-Incin} \end{aligned} \tag{14}$$

Here, the Flow "HHconsumptionSourceFlow" is the Source Flow.

For similar reason a special non-negative Flow, Sink Flow, is introduced for Sink Proc. Thus e.g. for Proc "Landfill" in Figure 1 introduce the Sink Flow "LandfillSinkFlow" and in informal notation the equation

$$\text{LandfillSinkFlow} = \text{Bottomash-to-Landfill} \tag{15}$$

Finally, for Buffer Proc introduce a special Flow, Buffer Flow. For Buffer Flows the sign of the numerical value is not in general known. This is in contrast to intermediate Flow, Source FLow and Sink Flow that are all non-negative. For the example in Figure 1 it is not known if the net balance in Proc "Electricity distribution system" is negative, positive or zero. For this Proc the balance equation in informal notation for the Buffer Flow is

$$\begin{aligned} \text{ElecDistBufferFlow} &= \text{El-to-AnaerobDigestion} \\ &+ \text{El-to-PretreatSeparation} \\ &- \text{BiogasCHP-to-El} \\ &- \text{WasteincinerationCHP-to-El} \end{aligned} \tag{16}$$

(17)

The sign convention here is chosen such that the Buffer Flow is positive if the net outflow is positive and negative if the net inflow is positive. I.e., the Buffer Flow represents the net outflow quantity. If a Source Proc is reclassified as a Buffer Proc the sign of the net Flow from the Buffer Proc is non-negative (as for the Sink Proc), and if a Sink Proc is reclassified as a Buffer Proc the sign of the net Flow from the Buffer Proc changes (i.e., it is the opposite as for the Sink Proc).

It may be noted that for the model to work as intended the Source and Sink Proc types are not needed, they may be replaced by Buffer Proc. However, for understanding, communication and debugging reasons it is recommended to use the Source and Sink Proc types whenever possible.

Network model - generalized

OptiFlow is a *generalized* network model. The "generalized" aspects of the model are that the coefficients in relations like (2) - (10) may differ from 1 and -1. This has implications for properties of the solution (in particular with respect to integrality of solution values) and its interpretation.

Background material on networks is indicated in Section 2.11 page 18.

Caveat

If you fail to grasp the above basics in relation to FLOW, PROC, FLOWFROMTOPROC and PROCINOUTFLOW you may encounter endless pain with the OptiFlow model.

2.2 Flow Quality

The term Quality was used above in a rather intuitive sense. Here it will be made more explicit.

TRACE (right) .		Energy	C-fossil	C-biogen	Ash	Plastic	Ferr. metal	Non-ferr.	Elec-tric.	Heat	Money	GHG emiss.
FLOW (below)	Unit	GJ	ton C	ton C	ton	ton	ton	ton	MWh	MWh	€	ton
Residual HHW	ton	x	x	x	x	xx	x	x				
Organic fract.	ton	x	x	x	x	x						
Ferrous Scrap	ton						x				x	x
Electricity	MWh	3.6							1		x	x
Heat	MWh	3.6								1	x	x

Table 1: Illustration of Qualities in relation to Flows. The left column gives the name of a Flow, and the next column shows its unit. It is understood that 1 unit of that Flow is considered. The remaining columns give the content of the Traces in the Flow. The 'x' indicates where non-zero numbers might be placed.

Thus, let there be defined a set TRACE. The elements in this set represent physical and other quantities related to the Flow in the network. The elements in the TRACE set are what we may be interested in keeping track of, reporting about, etc.

Assume for specificity that TRACE contains the following elements: energy, C-fossil, C-nonfosil, ferrous metal, Money and other elements specified in the top row of Table 1.

Further let there be given a parameter FLOWTRACE that links elements in FLOW with elements in TRACE. It might look as shown in Table 1. The number in a cell (indicated by 'x') specifies the content of Trace in a Flow under given conditions, e.g. with reference to Figure 1.

2.3 On linearity

One important property of the above is that the relations are linear, as seen from (2) - (13). This is important because any non-linearity will in general introduce more complications in the modelling, more efforts needed in solution of the model, and more difficulties on the interpretation of the results. It is assumed in the core parts of OptiFlow that all relations are linear.

Be aware that by 'linearity' is meant that the problem may be formulated as a traditional linear programming (LP) problem as illustrated in (23) - (25).

In some other conceivable modelling of waste and energy systems nonlinearity may e.g. origin from mixing flows of different Qualities, even if all other relations are linear.

For example when three types of waste are mixed in variable proportions, and the types differ with respect to energy content per tonne. To illustrate, let $F_7^{71,0}$ represent the first type, $F_7^{72,0}$ the second type and $F_7^{73,0}$ the third type, and assume their energy contents per tonne to be $e_7^{71,0}$, $e_7^{72,0}$ and $e_7^{73,0}$, respectively. This might be as illustrated in Figure 5.

Relevant equations are (18), representing the mass balance of flows (in tonne/h or similar), and (19) representing the energy balance of flows, viz., summation of energy content inflows to output $e_8^{0,8}F_8^{0,8}$ (GJ/h or similar).

$$F_7^{71,0} + F_7^{72,0} + F_7^{73,0} = F_8^{0,8} \tag{18}$$

$$e_7^{71,0}F_7^{71,0} + e_7^{72,0}F_7^{72,0} + e_7^{73,0}F_7^{73,0} = e_8F_8^{0,8} \tag{19}$$

This implies that the energy content (in GJ/tonne or similar) of $F_8^{0,8}$ is:

$$e_8^{0,8} = \frac{e_7^{71,0}F_7^{71,0} + e_7^{72,0}F_7^{72,0} + e_7^{73,0}F_7^{73,0}}{F_7^{71,0} + F_7^{72,0} + F_7^{73,0}} \tag{20}$$

The last equation represents the energy content $e_8^{0,8}$ of the output of the mixed waste (GJ/tonne). This is not known, but to be determined from elements of (19). The expression for this is (20), which is non-linear.

However, this is not how the described mixture of waste should be done in OptiFlow. In this model, linearity is maintained by ensuring that Proc P^{71}, P^{72} and P^{73} in Figure 7 each separate

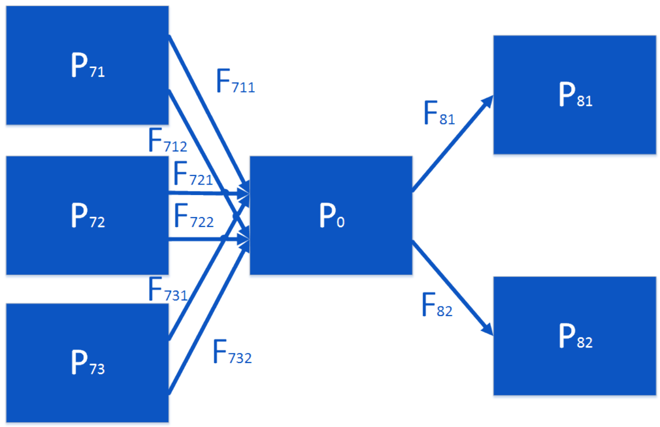

Figure 7: OptiWaste maintains linearity of the VJoin Proc by appropriate formulation.

their output in two Flows (with dimensions tonne/h and GJ/h, respectively) going to P^0. I.e., the two Flow Bundles share Proc P^0 (page 9). Similarly the outFlows (with dimensions tonne/h and GJ/h, respectively) from P^0 seen on Figure 7 maintain the two kinds of inFlow separate. With this representation, using a combination of Transform and VJoin Proc, the linearity is maintained.

The associated liniear OptiWaste equations are:

$$
\begin{aligned}
F_{81}^{0,81} &= c_{711,81}^{0} F_{711}^{71,0} \\
&+ c_{721,81}^{0} F_{721}^{72,0} \\
&+ c_{731,81}^{0} F_{731}^{73,0}
\end{aligned} \tag{21}
$$

$$
\begin{aligned}
F_{82}^{0,82} &= c_{712,82}^{0} F_{712}^{71,0} \\
&+ c_{722,82}^{0} F_{722}^{72,0} \\
&+ c_{732,82}^{0} F_{732}^{73,0}
\end{aligned} \tag{22}
$$

The reason for the failure of the representation in Figure 5 is that the three inFlows to P^0 are of different Qualities (Section 2.2) and therefore violate the assumption for the VJoin Proc.

The linearity property will be maintained under the enhancements to the core model that are presented further on in this Section and in most of the more technical documentation in Section 3.

Obviously, not all aspects of the waste and energy systems may be modeled in a linear way. Although sometimes a reformulation may turn apparently inherently non-linear relations into linear ones without loss of precision as the above (21) - (22) demonstrates, there are of course other situations where this is not possible. In such cases there is a trade-off between the precision of the modeling and the attractiveness of linear modeling - i.e., a usual challenge of modeling.

Hence, although the core parts of the OptiFlow are linear, there are certain extensions to the model that are not; see Section 2.10 page 17 for an overview.

As seen, the network framework of OptiFlow permits a quite flexible and powerful way of modeling a waste and energy system.

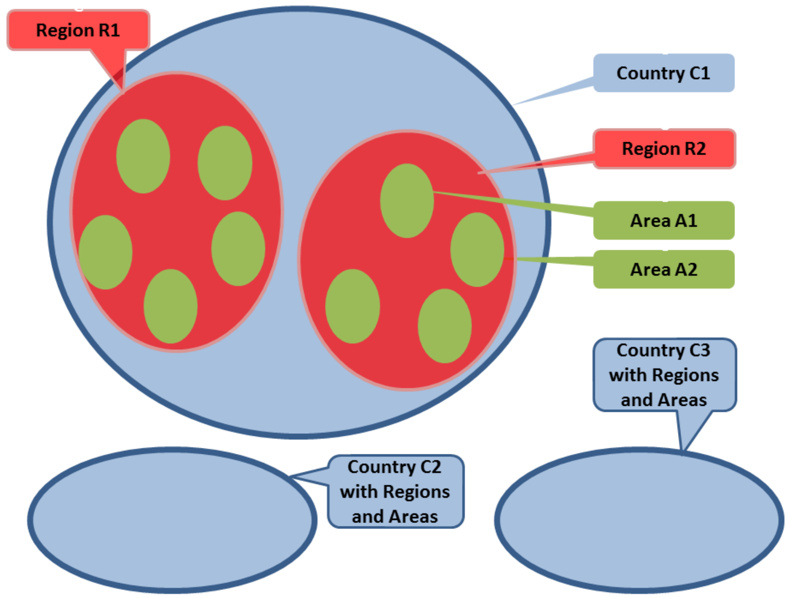

Figure 8: Geography with Countries, Regions and Areas. Proc within Areas are not shown, nor are Flow relations between the geographical entities (they are indicated in Figure 9).

2.4 Geography

Area, Region, Country

The model permits specification of geographically distinct entities. The main types of geographical entities are Areas, Regions, and Countries. These entities are in relation to the data structure specified by the sets AAA (Areas), RRR (Regions) and CCC (Countries), and for the subsets relating these sets (such as specification of the Areas that are in a specific Region).

To ensure generality in geographical specification, all geographical entities, including the elements in AAA, RRR and CCC, are specified in set CCCRRRAAA.

Each Country is constituted of one or more Regions while each region contains zero or more Areas. Any Area must be included in exactly one Region, and any region must be included in exactly one Country, see also Figure 8. Thus, there is a three-level hierarchical relations between Countries, Areas and Regions.

The Areas are the lower level building blocks with respect to the geographical dimension. Thus, all Proc are described at the level of Areas.

The Regions are for representation of electricity entities, which are relevant when more details are included for electricity transmission.

Transport

With a specification of geography follows the possibility to transport materials around. To permit transport between a pair of Areas each of the two areas must have a single specific Proc, the Exim Proc. The Exim Proc is the entry/exit point for the Area, all Flow that may be

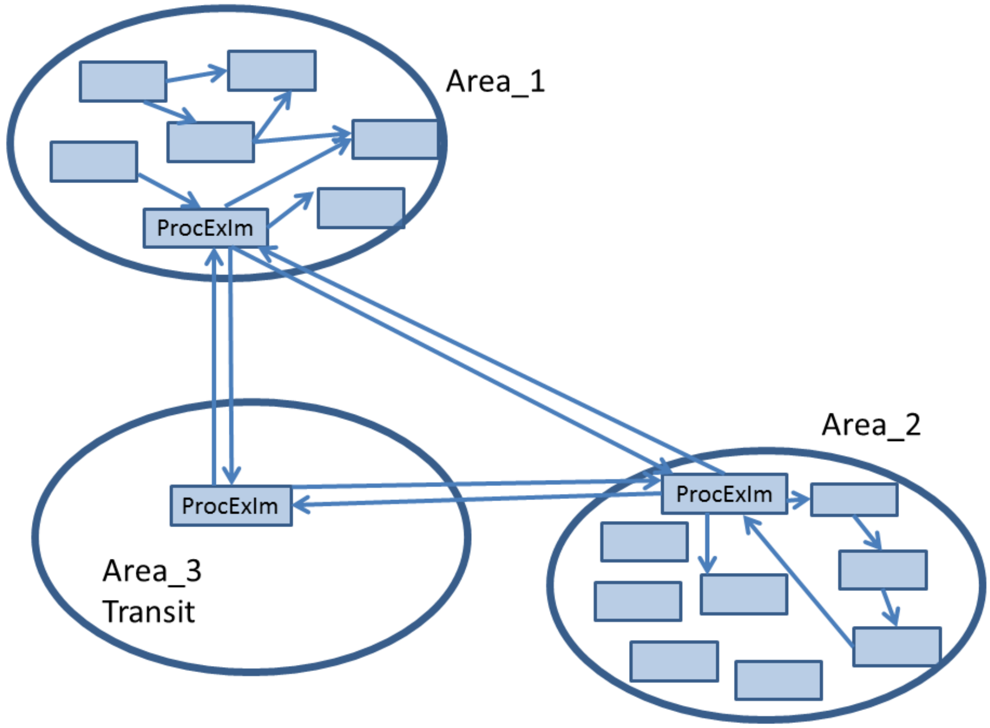

Figure 9: Transport between Areas and the role of ProcExIm as port (or gate) of the Areas. Note the Transit Area.

imported to or exported from an Area must pass through this Proc, cf. Figure 9. The set PROCEXIM, a subset of set PROC, holds all Exim Proc.

Transport distances may be specified for all pairs of Exim Proc. Each transport activity can have associated cost, emission and other consequences, depending linearly on transport distance and transported amount.

2.5 Time

Time is given in OptiFlow by Years, Seasons and Terms, held in sets YYY, SSS and TTT, respectively, with three-level hierarchical relations. Thus, the Seasons SSS represent sub-divisions of any Year and the Terms TTT represent sub-divisions of any Season. Subsets Y, S and S of YYY, SSS and TTT, respectively, are subsets to be used for simulation. For example with 52 elements in SSS (thus, each element represents one week) and 168 elements in TTT it is possible to model a subdivision of the Year into 52*168 or 8736 time segments (each of approximate duration 1.003 hours for a non-leap year). If S is in this example chosen to have only 4 elements while T has 168 elements, this may be interpreted as if the year is represented by four weeks, each week with hourly resolution.

The time resolution within the year need not have equal importance for all parts of the modeled system. Hence, a dual time resolution may be applied. Set PROC_T (a subset of PROC) is used for specifying which Proc that relate to a fine time resolution. Those elements in PROC that are not in PROC_T relate to a less fine time resolution, more specifically, only one element from T is used, while for elements in PROC_T all elements in T are used.

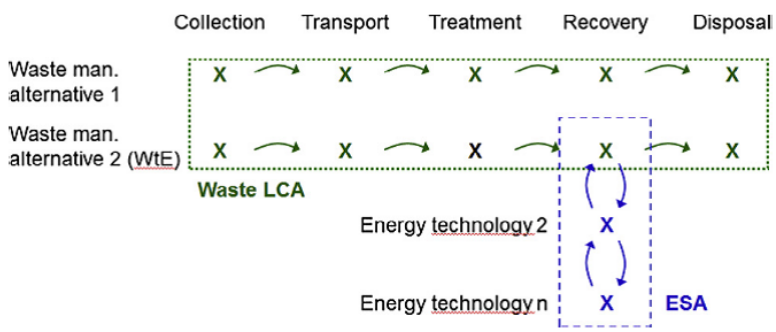

Figure 10: System boundaries of energy system analysis (ESA) and waste management life cycle assessment (LCA). (Waste Management 38, 2015, 486-495)

2.6 Storages

Storages may be modeled with capacities. There are two kinds of Storages, with Seasonal balance and with Annual balance (intra-seasonal and inter-seasonal, respectively), defined by sets PROCSTORAGE and PROCSTORAGE_Y that are subsets of PROC. A storage may be applied to any kind of Flow (e.g., household waste, a sorted fraction, or energy).

The balance equation (be it Seasonal or Annual) ensures that over the relevant time span (Season or Year, respectively) the total inflow to a Storage equals the total outflow.

2.7 Endogenous Investments

Any Proc may be specified to have a certain capacity, expressed as the maximal size of Flow entering or leaving it. Such capacity may be increased by the optimization. The capacity increase is characterized by costs, life time, and other relevant information, which is taken into account in the optimization. Section 3.5 page 23.

2.8 Objective function

In OptiFlow the optimal solution is determined, i.e., the solution that satisfies all the equations and in addition optimizes (maximizes or minimizes, whichever is relevant) the value of the objective function.

There are a number of several objective functions to choose from. The obvious one is a linear weighting of some of the Flows by attaching coefficient to each of these Flows. The relevant Flows are specified in the set FLOWINDIC which is a subset of set FLOW. Section 3.6 page 24 and equation (32) page 28 give more detail.

2.9 Integration of waste and energy in a single model

This section sketches waste management life cycle assessment (LCA) and energy system analysis (ESA) in a common formulation as optimization problems, an approach which is applied in the OptiFlow model.

In a typical ESA primal energy carriers like coal, gas, oil, wood, wind, solar radiation and others undergo transformations in energy technologies to end up in a small number of main energy carriers for use in households and industry are handled, in many case electricity and heat. The emphasis is often on the choices of energy carriers, their transformations, and the associated cost, emissions, and other aspects.

In a typical waste management LCA waste from households and industry is accounted through a number of handling operations and transformations (generation, collection, transport, sorting, transformations, recycling, reuse, disposal). The emphasis is often on the whole chain from waste generation (or even the generation of the goods that are consumed with some waste consequences) over transformations to disposal or reuse. Figure 10 illustrates the two types of analysis and their potential combination.

By combining the two perspectives into one model, aspects from both of these methodologies can be represented and balanced. When expressed as linear programming (LP) problems, LCA and ESA can commonly be formulated as:

$$\max Z = c'x \quad (23)$$
$$Ax \leq b \quad (24)$$
$$x \geq 0 \quad (25)$$

Here, c and b are column vectors of coefficients (input data), A is a matrix of coefficients, x is column a vector of variables and denotes transpose. The first line is the objective function which maximizes the value Z. This is done by choosing the optimal value of the x vector subject to the equations and inequalities in the second and third lines. The model in (23) - (25) is widely used in ESA, e.g. in Balmorel (Ravn & al. 2001). In such an application, the inequalities may represent e.g. capacity constraints on production, storage and transmission units, fuel use restrictions, and emission limits on CO_2 equivalents. Due to (24) the system might maybe more suitable be called *piecewise* linear than linear; in any case it is convex.

An LCA can more specifically be formulated as the following, which is a sub version of (23) - (25):

$$\max Z = c'x \quad (26)$$
$$Ax = b \quad (27)$$

In the LCA application, a single equation in $Ax = b$ (i.e., one row) will then specify relations between, e.g. the flow of waste entering a sorting Proc and the various fractions leaving the sorting Proc. Other equations can specify emissions and energy and resource consumption, associated with the Proc.

The shown LCA optimization problem has no inequalities. This reflects that in LCA, typically a number of discrete fixed alternatives are simulated and compared with the goal of identifying the most attractive option among them. Such comparative analyses can for instance cover energy recovery versus recycling of a given waste fraction. As such, the best solutions are not identified as part of an optimization but based on the interpretation of the simulated results of the LCA.

As outlined above, LCAs and ESAs can however be formulated on the same form as linear optimization problems. Thus, when integrating the life cycle impacts of several possible alternatives into a single optimization model, the optimal alternative or combination of alternatives can be identified through optimization. This modelling approach is fully compatible with the approach often used for ESA and, thus, qualified for integration with an energy system model.

2.10 Further remarks and extensions

Additional remarks on LCA, ESA and OptiFlow:

- The LCA traditionally comes in two versions, attributional and consequential. Consequential LCA is concerned with how (small, marginal) changes in waste generation will affect environmental and other conditions. In the context of the OptiFlow model the consequential LCA is used.

- The LCA is often represented as a linear system - not piecewise linear in the sense of the LP model (Sections 2.3 page 12, 2.9 page 16). Additionally OptiFlow may be applied with certain non-linear feature, see below.
- LCA typically works with a functional unit, in the context of waste management this could e.g. be one tonne of municipal solid waste (MSW). In the context of the OptiFlow model the full waste (and other) amounts are used. One consequence is that economy of scale may be naturally handled in OptiFlow (Section 2.10 page 17).
- LCA handles many Qualities of Flows, as does OptiFlow (Section 2.2 page 11)
- LCA does not traditionally deal with economic aspects of waste and waste treatment systems, this is possible in OptiFlow due to the generality of the modeling with respect to Qualities (Section 2.2 page 11)
- In LCA the time differentiation or dynamics within the year is often ignored, while aspects related to the longer time frame (e.g., 100 years) may be central (Section 2.5 page 15).
- In LCA geography is often aggregated (Section 2.4 page 14).

Certain functionalities are provided as extensions to the core model. An extension is typically implemented as an "addon" in the GAMS code, cf. Section 3.15 page 35. Some examples are

- Economy of scale (implying Mixed Integer Programming, MIP, and hence non-linearity)
- Unit commitment (implying Mixed Integer Programming, MIP, and hence non-linearity)
- Integration with Balmorel, cf. Section 2.9 page 16. Detailed documentation of the Balmorel model may be found at www.balmorel.com
- Multi-criteria decision analysis MCDA - deriving the Pareto optimal frontier
- Quadratic elements (implying Quadratically Constrained Problems, QCP, and hence non-linearity)

2.11 Further inspiration

More technical details on the OptiFlow model are found in Section 3 below. Some background material of various kinds is indicated below.

Background material on linear programming may be found in e.g. Frederick S. Hillier, Gerald J. Lieberman: Introduction to Operations Research, McGraw-Hill, 2001; David G. Luenberger: Linear and Nonlinear Programming, Addison-Wesley, 1989. - Standard topics with LP are i.a. optimization, primal and dual models and solutions, and marginal values.

Background material on networks and convexity may be found in e.g. M. S. Bazaraa, J. J. Jarvis and H. D. Sherali: Linear Programming and Network Flows, John Wiley & Sons (2010); Ravindra K. Ahuja, Thomas L. Magnanti and James B. Orlin: Network Flows: Theory, Algorithms and Applications, Prentice-Hall (1993); George L. Nemhauser and Laurence A. Wolsey: Integer and Combinatorial Optimization, Wiley1988.

Background material on integer models may be found in e.g. Laurence A. Wolsey: Integer Programming, Wiley 1998.

3 Further documentation of OptiFlow

This Section gives more details on the OptiFlow model and its implementation. The emphasis is on describing what the OptiFlow model can do and how the User can make it do it; less emphasis is on the internal working of the model.

For subjects that are not mentioned here, please refer to the gentle introduction in Section 2 from page 4, or to the complete list of identifiers in Section 4 page 36.

Finally, the GAMS code itself provides the ultimately updated and detailed documentation.

3.1 On GAMS

The model is implemented in the GAMS modelling language. GAMS is the acronym for General Algebraic Modeling System. The system is suitable for formulation, documentation and solution of large mathematical models.

GAMS is a commercial product, to apply OptiFlow the User therefore needs the GAMS system installed with an appropriate license. To use GAMS also an appropriate solver is needed (optimization program), this may be licensed or open source.

It is assumed that the reader is sufficiently familiar with the GAMS language. A Tutorial, a User's Guide, an Expanded GAMS Guide (McCarl), and other relevant information about the GAMS modeling system may be found in the GAMS system or at www.gams.com.

GAMS uses a number of keywords that will be used in the present document, e.g., sets, parameters, variables, equations, models and more, consult some of the above references if needed.

3.2 Geography and Time

Geography

The main elements related to geography were introduced in Section 2.4 page 14.

The consistency of the three-level hierarchical structure is ensured by sets CCCRRR(CCC,RRR) "Regions in countries" and RRRAAA(RRR,AAA) "Areas in regions". The set CCCRRRAAA holds all the geographic elements i.e., all elements from CCC, RRR and AAA.

One of the key elements in relation to geography is that the geographical extension of the data input may be larger (i.e., having data with respect to more geographical entities) than is actually used in a simulation. E.g., the input data may have data on Denmark, England, Poland and Sweden (hence, set CCC "Set of all countries" contains the names of these four Countries) but a simulation will be performed for Denmark and Sweden only if the set C(CCC) "Countries in the simulation" contains the names of only these two Countries.

Moreover, the code ensures that not only are the two Countries selected for the simulation, also the associated Regions and Areas are consistently selected. This is obtained using subsets IR(RRR) "Regions in simulation", IA(AAA) "Areas in the simulation" and ICA(CCC,AAA) "Relation of areas to countries in the simulation".

Further subsets AAAIMPORT and AAATRANSIT related to transport are treated in Section 3.4 page 23.

Time

The main elements related to geography were introduced in Section 2.5 page 15. As appears, time has, like geography, a three-level hierarchical structure applying sets YYY, SSS and TTT to represent Years, Seasons and Terms, reprectively. Like for geography, it is possible to distinguish between what is in the input data (using YYY, SSS and TTT) and what is used in the simulation (using subsets Y(YYY), S(SSS) and T(TTT)).

Since the number of time slices (or time segments) (S,T) may be chosen as needed, the duration of a time slice (S,T) needs adaptation to ensure consistency. This is done automatically such that the total duration of all the (S,T) is one year (8760 hours). With a subdivision of the Year into e.g. 52*168 or 8736 time slices each will have a duration of 1.003 hours for a non-leap year. A subdivision into e.g. 52*336 or 17472 time slices each will have a duration of approximately 30 minutes.

Note that there is in fact a fourth level in the time structure. Thus, PROC_T(PROC) is the subset of PROC that operates on full time resolution T, while elements not in PROC_T are operating on only a single elements within T. Such distinction with respect to domain (i.e., indexes in declaration) detail are usually indicated by postfix _ followed by the most detailed level; for time this could e.g. also be _Y. Other similar ideas are applied elsewhere with relation to time or geography, as seen in e.g. Section 3.6 page 24 and Section 3.3 page 21.

3.3 Processes, Flows and how they relate

Simple sets and subsets - PROC and FLOW

PROC and subsets

The set of all Proc is given in SET PROC "All processes".

The set PROC is subdivided into a number of mutually exclusive subsets, viz.,
PROCSOURCE(PROC), PROCSINK(PROC), PROCBUFFER(PROC), PROCEXIM(PROC), PROCSTORAGE(PROC), PROCSTORAGE_Y(PROC), PROC_T(PROC), IPROCINTERIOR(PROC). PROCINDIC(PROC)

Most of these set were described in Section 2.1 page 4. The set IPROCINTERIOR holds the set of PROC that are not part of any of the other subsets (except PROCINDIC, cf. Section 3.6 page 24) of PROC.

The notation like "PROCSTORAGE(PROC)" indicates that the set PROCSTORAGE is a subset of the set PROC. Here PROC serves as the domain; the domain is the collection of indexes used in the declaration, many subsets have several sets in their domain. Simple sets are sets without domain, like PROC, FLOW, YYY, SSS, TTT or CCCRRRAAA.

FLOW and subsets

The set of all Flow is given in SET FLOW "All flows".

A subset of FLOW is FLOWINDIC(FLOW) that is used with the objective function, cf. Section 3.6 page 24

Proc and Flow linkage

As explained in Section 2, in particular from page 5, the following sets and parameters are essential for the topology of the network.

SET FLOWFROMTOPROC(AAA,IPROCFROM,IPROCTO,FLOW) "FLOW (index 4) from PROC (index 2) to PROC (index 3)". The sets IPROCFROM and IPROCTO are sets that are identical to PROC, they are said to be aliased with PROC. Such aliases are introduced for readability and because they may be needed. The 'I' in front of the names of the aliases indicates that the identifiers are internal, i.e., not meant to be handled by the User , cf. Section 3.13 page 32. It also indicates that the identifier does not get its values from a data file but by assignment in the code, cf. Section 3.11 page 31.

PARAMETER PROCINOUTFLOW(AAA,PROC,IFLOWIN, IFLOWOUT, IPROCINOUTRELATION) "Relationship at PROC (index 2) between FLOWIN (index 3) and FLOWOUT (index 4); interior PROC only (∼)". The symbol ∼ is an indication of the unit in which the values are given, cf. Section 3.12 page 31.

SET IPROCINOUTRELATION ”Possible relationships between inFlow and outFlow for a Proc”
/
IONEONE ”Transform relation”
IONEONEREC ”Transform relation using reciprocal value”
IONEMANY ”VSplit relation”
IMANYONE ”VJoin relation”
/;

The relations to the Transform, VJoin and VSplit types in Section 2.1 page 5 are obvious. The IONEONEREC is essentially the same as the IONEONE relation with the difference that the constant (the numercal value of the relation) is given with the reciprocal value as this may in some cases seem more natural. Additional internal sets and parameters serve as convenient helper (see Section 3.13 page 32 on ’internal’). Thus, e.g. IPROCINOUTFLOW is merely a resumé of the topology relations in PROCINOUTFLOW.

A sign convention is used on the numerical value of the parameter to indicate if the relation of between Flows linking to a Proc is between two inFlows or two outFlows, or between one inFlow and one outFlow. The sign of a parameter value is therefore a logical information, not a numerical.

To get summations in equations right (viz., not repeating equations superfluously) some bookkeeping is needed as explained in relation $P^{PFFvso2}$ and $P^{PFFvsi2}$ on page 26. The following two internal subsets are at the core of handling this.

PARAMETER IPRIOONEMANY(AAA,PROC,IFLOWIN,IFLOWOUT) ”Indicator for use with VSplit: 0: not outsum, 1: outsum primary, 2: outsumt secondary; only primary used for equation indexes (∼)”

PARAMETER IPRIOMANYONE(AAA,PROC,IFLOWIN,IFLOWOUT) ”Indicator for use with VJoin: 0: not insum, 1: insum primary, 2: insum secondary; only primary used for equation indexes (∼)”

Further convenient internal sets are

SET INEGPROCININFLOW(AAA,PROC,IFLOWIN,IFLOWIN2) ”Set with two incoming FLOW to a PROC (negative value of PROCINOUTFLOW)”

SET INEGPROCOUTOUTFLOW(AAA,PROC,IFLOWOUT,IFLOWOUT2) ”Set with two outcoming FLOW to a PROC (negative value of PROCINOUTFLOW)”

SET ITRANSFROMTO(IAAAE,IAAAI) ”Set of Areas that are connected by transport (not necessarily symmetric)” (Section 3.4 page 23)

SET ILEAVEPROC(AAA,PROC,FLOW) ”For each PROC: the set of Flows that originate from this Proc (based on FLOWFROMTOPROC)”

SET IENTERPROC(AAA,PROC,FLOW) ”For each PROC: the set of Flows that enter this Proc (based on FLOWFROMTOPROC)”

SET IAPROCKAPNEW(Y,AAA,PROC) ”Area, Proc where technology may be invested based on APKN and some implicit constraints” (Section 3.5 page 23)

Bounds on Flows

Flows may be bounded according to need. It is possible to set lower bounds, upper bounds or both (setting both to the same value is in GAMS the same as fixing it).

These three possibilities are specified using internal set ILOUPFXSET, given as

SET ILOUPFXSET ”Set representing lower, upper and fixed values ”
/
ILOUPFX_LO ”Lower limit”
ILOUPFX_UP ”Upper limit”
ILOUPFX_FX ”Fixed value”
/;

Bounds on individual interior FLOW are given in FLOWBOUND,
PARAMETER FLOWBOUND(YYY,AAA,IPROCFROM,IPROCTO,FLOW,ILOUPFXSET) ”Bounds on individual interior FLOW (U/h)”.

Bounds on all FLOW related to a PROC are given in SOSIBUBOUND (’sosibu’ is short for Source, Sink and Buffer). Several versions are given, differing with respect to the number of indexes in the time dimension.

PARAMETER SOSIBUBOUND(AAA,PROC,FLOW,ILOUPFXSET) ”Bounds on Source, Sink and Buffer Process Flows - Annual values (*)”

PARAMETER SOSIBUBOUND_AS(AAA,PROC,FLOW,SSS,ILOUPFXSET) ”Bounds on Source, Sink and Buffer Process Flows (*)”

PARAMETER SOSIBUBOUND_AST(AAA,PROC,FLOW,SSS,TTT,ILOUPFXSET) ”Bounds on Source, Sink and Buffer Process Flows (*)”

PARAMETER SOSIBUFLOW_VAR_T(AAA,PROC,FLOW,SSS,TTT) ”Variation in some Source, Sink or Buffer Flows over the year (∼)”

Bounds on relations between inFlow and outFlow are specified through

PARAMETER FLOWBOUNDSHAREIN(AAA,IPROCFROM,IPROCTO,FLOW,ILOUPFXSET) ”Bounds on shares of sum of inflow to PROC (share)”

PARAMETER FLOWSHAREOUT2IN(AAA,IPROCFROM,IPROCTO,FLOW,ILOUPFXSET) ”Bounds on shares (of inFlow) of variable splits of outflow from PROC (share)”

PARAMETER FLOWSHAREOUT2OUT(AAA,PROC,IFLOWIN,IFLOWOUT,IFLOWOUT2,ILOUPFXSET) ”Fixed relation (index 4 divided by 5) between two split FLOWs leaving PROC (share)”

As seen, there are input data identifiers that differ only with respect to the domain (indexes) such that there are differing levels of detail with respect to geography and time. If there are such two or more input data files the values to be used are found by overwriting values with less detail with values with more detail. Hence, a flexibility is obtained with respect to the resulting level of geography and time values.

Properties of Proc

A Proc may be given some properties (or data). The types of properties are specified in SET PROCDATASET ”Process data types”
/
PROCINVCOST ’Investment cost (MMoney/(U/h))’
PROCKAPVARIABLE ’Capacity is a variable to be found for each year (1/0) ’
PROCFROMYEAR ’Process available from this year (int) ’
PROCLIFETIME ’Economic lifetime, years (int)’
/;

The values are given in PARAMETER PROCDATA(PROC,PROCDATASET) ”Process data (*)”.

Application to investments is described in Section 3.5 page 23.

3.4 Transport, Export and Import

Transport is related to pairs of Areas, cf. Section 3.2 page 19.

PARAMETER TRANSDIST(PROC,IAAAE,IAAAI) ”Distance between two Areas (not necessarily symmetric) (km)” where IAAAE and are alised with AAA, Section 2.4 page 14.

PARAMETER TRANSFLOWMAX(YYY,IAAAE,IAAAI,PROC,FLOW) ”Transport quantity maximum value (ton/h)”.

Further subsets are AAATRANSIT(AAA) ”Areas that are for waste transit only” and AAAIMPORT(AAA) ”Areas that are for waste import only”.

Transport comes with costs of various kinds. The types of costs are specified in
SET TRANSCOSTDATASET ”Transport cost element data and mapping to FLOWINDIC”
/
TCDMONEYKMTON ”Transport cost related to distance and amount (Money/(ton*km))”
TCDMONEYTON ”Handling cost related to transported amount (Money/ton)”
TCDGHGEMISSION ”CO2 equiv emission related to distance and amount (ton/(ton*km))”
/;

The values are the given in
PARAMETER TRANSCOST(YYY,TRANSCOSTDATASET,TRANSDISTWEIGHT, PROC,FLOWINDIC) ”Transport ’cost’ in terms of INDICS (*)”.

PARAMETER TRANSFUEL(YYY,TRANSCOSTDATASET,PROC,FLOW) ”Transport fuel”.

3.5 Proc Capacity and Investments

In the OptiFlow model any Proc has an associated capacity. Capacity is implemented as a bound on a Flow that enters or leaves the Proc.

Capacity may be exogenous (i.e., specified explicitly by the User in a data file) or endogenous (found by the model according to the input data and the objective function).

Capacity cannot be given for entering Flow with IMANYONE (i.e., not for VSplit Proc) in PROCINOUTFLOW nor for leaving Flow with IONEMANY (i.e., not for VJoin Proc).

The following tow set specify the type of information and for any Proc on which Flow the capacity is given.

SET IFLOWINOUT ”Specifies if the capacity of Proc is set as a bound on entering/leaving Flow from/to Proc (for storage use with _STO)”
/
IFLOWINOUT_IN ”The bound is on the Flow to Proc
IFLOWINOUT_OUT ”The bound is on the Flow from Proc
IFLOWINOUT_STO ”The bound is on the Storage
/;

SET PROCKAPDATA(PROC,FLOW,IFLOWINOUT) ”Process data: capacity relative to FLOW and in/out direction ()”.

The Proc capacity that is exogenously given may now be entered in PARAMETER PROCKAPFX(YYY,AAA,PROC,FLOW,IFLOWINOUT) ”Capacity of Process ((U/h), (U) for storage)”.

The endogenous capacity to be found by the model is not necessarily for any Area and Proc combination, but only for this given in SET AAAPROCKAPNEW(AAA,PROC) ”Areas for possible location of new Proc capacity”.

The endogenous capacity has some Proc specific properties like cost or life time, cf. Section 3.3 page 22.

3.6 Objective function

In an optimization model the objective function is the function that is to be maximized or minimized, subject to constraints. Thus, it presents the evaluation criterion for the results obtained with the given User supplied data. Often some of the constraints have a substantial role is this, e.g. a constraint limiting the emission of CO2 equivalents.

The objective function used in the OptiFlow model may therefore be seen as formulated using a number of sets, parameters, variables and equations.

The objective function (or 'equation' as is is in the GAMS terminology) is QOBJW "Objective function value assuming weighted objectives values technique (U)".

In the following more detail is provided on QOBJW.

Indicator, Proc and Flow

The term indicator is used to denote the items that are part of the evaluation of the attractiveness of a solution. In optimization terms the indicators appear in the objective function.

The evaluation is on Flows. The Flows that may be part of the evaluation (i.e, part of the objective function) are specified as FLOWINDIC, a subset of set FLOW. Similarly, the Proc that are to be relevant as part of this is specified as PROCINDIC, a subset of set PROC; only Proc that are Source, Sink or Buffer are intended be part of set PROCINDIC.

SET PROCINDIC(PROC) "The indicator PROC with relations to FLOW that we are interested in evaluating".

SET FLOWINDIC(FLOW) "The indicator flows that we are interested in evaluating".

Relations between PROC, FLOW and FLOWINDIC

The numerical relations between Flow and Indicators are specified in a number of SOSIBU2INDIC parameters ('sosibu' is short for Source, Sink, Buffer). Several versions are given, differing with respect to the number of indexes in time and geography dimensions.

PARAMETER SOSIBU2INDIC(YYY,PROC,FLOW,FLOWINDIC) "Coefficients for transformation of VSOURCE, VSINK and VBUFFER flows to FLOWINDIC values (*)"

PARAMETER SOSIBU2INDIC_AST(AAA,PROC,FLOW,FLOWINDIC,SSS,TTT) "Coefficients for transformation of VSOURCE, VSINK and VBUFFER flows to FLOWINDIC values (*)"

PARAMETER SOSIBU2INDIC_AS(AAA,PROC,FLOW,FLOWINDIC) "Coefficients for transformation of VSOURCE, VSINK and VBUFFER flows to FLOWINDIC values (*)"

PARAMETER SOSIBU2INDIC_RST(RRR,PROC,FLOW,FLOWINDIC,SSS,TTT) "Coefficients for transformation of VSOURCE, VSINK and VBUFFER flows to FLOWINDIC values (*)"

Constraints related to the objective function

The next set specifies the type of constraints and weights that can be applied.

SET IINDICLIMGOALSET "Limits and desirable values of INDIC"
/
INDICMIN "Lower limit for Indic value (-INF means unlimited downwards) (*)"
INDICMAX "Upper limit for Indic value (+INF means unlimited upwards) (*)"

INDICWEIGHT ”Indic relative weight (*)”
/;

The above is used in PARAMETER INDICLIMGOAL(FLOWINDIC,IINDICLIMGOALSET) ”Limits, desirable quantities and weight of Indics (*)” (input data, despite the initial ’I’, cf. Section 3.13 page 32).

The following variable VFLOWINDICVALUE is used to hold the sum of the Indic values; it is defined in equation QINDICQUANT. VFLOWINDICVALUE is used in the objective function QOBJW (using the ”INDICWEIGHT”) and in equations QINDICINDICMIN and QINDICINDICMAX (using ”INDICMIN” and ”INDICMAX”, respectively):

FREE VARIABLE VFLOWINDICVALUE(Y,FLOWINDIC) ”Quantities of indicators (time weighted) (U)”

EQUATION QINDICQUANT(Y,FLOWINDIC) ”Total quantity of INDIC (U)”

EQUATION QINDICINDICMIN(Y,FLOWINDIC) ”Minimal annual quantity of INDIC (U)”

EQUATION QINDICINDICMAX(Y,FLOWINDIC) ”Maximal annual quantity (sum-of-years) of INDIC (U)”

See also equation (32) page 28.

3.7 Variables

This Section will provide a more formal description of the endogenous parts of the model. The emphasis is on the key variables and equations, including the objective function, for the waste network flow part of the model. For units see Section 3.12 page 31. The description of the non-waste energy part of the model is omitted as this is more in line with well known energy modeling principles, see e.g. Ravn et. al (2001).

The notation here is not the same as used in Section 2 or in most later sections; however, it is convenient as a math-oriented notation, differing from the GAMS-oriented notation mainly used elsewhere in the present document. Also note that the tables (2) - (4) are meant to give an idea about the functionality, while some GAMS coding details may differ. A list of all GAMS identifiers is given in Table 4 page 39; all variable names start with ’V’.

The main symbols of the model are listed in Table 2 and Table 3.

Each variable listed in Table 2 is actually a vector, except V^{Idc}. A specific element in such vector is specified by sub-indices, e.g. $V_{p,p',f}$ is the element in V representing Flow f from Process p to Process p'. Similar notational convention applies for parameters.

Most Flows are positive (or zero), since they typically represent some physical entity. The exception is the net flow from Buffer Processes that may be positive, zero or negative. This is due to the use of Buffer Processes also for representing some net balances. For instance, a Buffer Process may represent an electricity hub that receives electricity generated by a waste incineration plant and delivers electricity to a waste sorting plant. Without further information it is not possible to know the sign of the net Flow from the Buffer Process. The Buffer Flow variable is therefore declared as free. The indicator Flow V^{Idc} is also declared as a free variable.

The sets P^S, P^I, P^B and P^{Int} are non-overlapping and together constitute set P, i.e., $P^S \cap P^I$, $P^S \cap P^B$, $P^S \cap P^{Int}$, $P^I \cap P^B$, $P^I \cap P^{Int}$, $P^B \cap P^{Int}$, and $P^S \cup P^I \cup P^B \cup P^{Int} = P$. The relation between set P^{Idc} of indicator nodes and the other subsets of P is user specified. Typically, P^{Idc} will be the set P^B of buffer nodes but it may be a subset thereof.

The set of leaving flows that are summarized, P^{PFFvso}, is derived from R^{PFF} which has a field for specification of whether for any entry (p, f, f') the relation is to be a fixed relation between a single incoming flow f and a single leaving flow f', or whether the relation is the splitting of

Table 2: Symbols used for the OptiWaste model description (math-oriented notation). The first part describes input data and the second part describes variables. (Waste Management 38, 2015, 486-495)

Symbol	Type	Description	Indexes	Comment
P	Set	Set of Processes	-	
P^S	Set	Set of Source Processes	P	Subset of P
P^I	Set	Set of Sink Processes	P	Subset of P
P^B	Set	Set of Buffer Processes	P	Subset of P
P^W	Set	Set of waste generation Processes	P	Subset of P
P^{Idc}	Set	Set of Processes for indicators	P	Subset of P
F	Set	All Flows	-	
F^{Idc}	Set	Indicator Flows	F	Subset of F
F^W	Set	Waste generation Flows	F	Subset of F
R^{PPF}	Set	Flow is from Process to Process	P, P, F	
R^{PFF}	Parameter	Relation of Flows into and from Process	P, F, F	
R^{CIdc}	Parameter	Coefficients of Flows in objective function	F	
V^{Idc}	Variable	Objective function value	-	Free
V	Variable	Flow between two processes	P, P, F	Positive
V^O	Variable	Net Flow from Source Process	P^S, F	Positive
V^I	Variable	Net Flow from Sink Process	P^I, F	Negative
V^B	Variable	Net Flow from Buffer Process	$P^B F$	Free
V^{Idc}	Variable	Indicator Flow	P	Positive

Table 3: Symbols used for the OptiWaste model description (math-oriented notation). Derived sets. (Waste Management 38, 2015, 486-495)

Symbol	Type	Description	Indexes	Comment
P^{Int}	Set	Set of Interior nodes	P	Subset of P
P^{PFFvso}	Set	Set of leaving flows that are summarised	P, F, F	
$P^{PFFvso2}$	Set	In set P^{PFFvso} but not first element	P, F, F	Subset of P^{PFFvso}
P^{PFFvsi}	Set	Set of entering flows that are summarised	P, F, F	
$P^{PFFvsi2}$	Set	In set P^{PFFvsi} but not first element	P, F, F	Subset of P^{PFFvsi}

a single incoming flow f on two or more leaving flows. The elements in P^{PFFvso} are arbitrarily split in two subsets with all but one element in the subset $P^{PFFvso2}$. Similar ideas are behind the sets P^{PFFvsi} and $P^{PFFvsi2}$ in relation to a node with two or more incoming flows that are to be summarized to a single leaving flow. The purpose of these subsets is to ensure that in case of such splitting or summation there will be generated only one instance of the equation involving such (p, f, f') (the instance that holds the summation of a splitting relation) rather than one instance for each (p, f, f').

3.8 Equations

Note: The notation related to (28) - (32) (similar to those in Waste Management 38, 2015, 486-495) is not the same as used in Section 2 or in later sections; however, it is convenient as a math-oriented notation, differing from the GAMS-oriented notation used elsewhere in the present document. Also note that the equations (28) - (32) are meant to give an idea about the functionality, while some GAMS coding details may differ. For units see Section 3.12 page 31. A list of all identifiers are given in Table 4 page 39; all equation names start with 'Q'.

The key equations are those that describe the balances of the flows entering and leaving a process. Some are listed in Table 5. For physical entities these typically describe mass balances, i.e., that the sum of incoming mass equals the sum of outgoing mass. However, also more special relations may be present, like specification of cost associated with operating the real process that a specific P represents. Table 5 shows that the same type of balance may be represented

Table 4: Relation between symbols in the two above tables and in the GAMS code.

Symbol (above)	GAMS Symbol	GAMS indexes	Type
P	PROC	-	Set
P^S	PROCSOURCE	(PROC)	Set
P^I	PROCSINK	(PROC)	Set
P^B	PROCBUFFER	(PROC)	Set
P^{Idc}	PROCINDIC	(PROC)	Set
F	FLOW	-	Set
F^{Idc}	FLOWINDIC	(FLOW)	Set
R^{PPF}	FLOWFROMTOPROC	(AAA,IPROCFROM,IPROCTO,FLOW)	Set
R^{PFF}	PROCINOUTFLOW	(AAA,PROC,IFLOWIN,IFLOWOUT,.. .. IPROCINOUTRELATION)	Par
R^{CIdc}			Par
V^{Idc}	VOBJ		Var
V	VFLOW	(Y,AAA,IPROCFROM,IPROCTO,FLOW,.. ..,S,T)	Var
V^O	VFLOWSOURCE	(Y,AAA,PROCSOURCE,FLOW,S,T)	Var
V^I	VFLOWSINK	(Y,AAA,PROCSINK,FLOW,S,T)	Var
V^B	VFLOWBUFFER	(Y,AAA,PROCBUFFER,FLOW,S,T)	Var
V^{Idc}	VFLOWINDICVALUE	(Y,FLOWINDIC)	Var
P^{Int}	IPROCINTERIOR	(PROC)	Set
P^{PFFvso}	ILEAVEPROC	(AAA,PROC,FLOW)	Set
$P^{PFFvso2}$	IPRIOVARIOUTSUM	(AAA,PROC,IFLOWIN,IFLOWOUT)	Set
P^{PFFvsi}	IENTERPROC	(AAA,PROC,FLOW)	Set
$P^{PFFvsi2}$	IPRIOVARIINSUM	(AAA,PROC,IFLOWIN,IFLOWOUT)	Set

by two equations (distinguished by suffixes like "_T" or "_S"), to accommodate the dual time applied, cf. Section 2.5 page 15.

These process balance equations are of four kinds, relative to elements of P that are of type Source, Sink, Buffer or Interior, respectively.

Table 5: Some main equations in OptiFlow.

Symbol	Indexes	Unit
QNODEBALANCE_S	(Y,AAA,PROC,IFLOWIN,IFLOWOUT,S,ITWITHPROC)	U/h
QNODEBALANCE_T	(Y,AAA,PROC,IFLOWIN,IFLOWOUT,S,T)	U/h
QSOURCENODEBALANCE	(Y,AAA,PROCSOURCE,FLOW,S,T)	U/h
QSINKNODEBALANCE	(Y,AAA,PROCSINK,FLOW,S,T)	U/h
QBUFFERNODEBALANCE_S	(Y,AAA,PROCBUFFER,FLOW,S,T)	U/h
QBUFFERNODEBALANCE_T	(Y,AAA,PROCBUFFER,FLOW,S,T)	U/h
QSTORAGEBALANCE	(Y,AAA,PROCSTORAGE,FLOW,S,T)	U
QSTORAGEBALANCE_Y	(Y,AAA,PROCSTORAGE_Y,FLOW,S)	U

For the Source type Process the equation (QSOURCENODEBALANCE in the GAMS code) is for any given Process $p \in P^S$ and any Flow $f \in F$ leaving p and entering some other p'. This equation is indexed over (p, f). Thus,

$$V^O_{p,f} \quad = \sum_{p' | R^{PPF}_{p,p',f}} V_{p,p',f} \tag{28}$$

For the Sink type Process the equation (QSINKNODEBALANCE in the GAMS code) is for any given Process $p \in P^I$ and any Flow $f \in F$ entering p coming from some other p'. This equation is indexed over (p, f). Thus,

$$V^I_{p,f} \quad = \quad - \sum_{p'|R^{PPF}_{p'p,,f}} V_{p',p,f} \tag{29}$$

For the Buffer type Process the equation (QBUFFERNODEBALANCE in the GAMS code) is for any given Process $p \in P^B$ and any Flow $f \in F$ entering p coming from some other p'. This equation is indexed over (p, f). The sign convention is chosen such that if a Source or a Sink Process is reclassified as a Buffer Process the sign of the net Flow from the Process remains unchanged. Thus,

$$V^B_{p,f} \quad = \quad \sum_{p'|R^{PPF}_{p,p',f}} V_{p,p',f} - \sum_{p'|R^{PPF}_{p'p,,f}} V_{p',p,f} \tag{30}$$

For an Internal Process the equation (QNODEBALANCE in the GAMS code) is more involved (see Section 3.13 page 32 on 'internal'). This equation is indexed over (p, f, f'). Thus, for any given Process $p \in P$ and any given Flow $f \in F$ into p and any given Flow $f' \in P$ leaving p, such that $(p, f, f') \notin P^{PFFvso2}_{p,f,f'}$ and $(p, f, f') \notin P^{PFFvsi2}_{p,f,f'}$

$$\begin{aligned}
& \textstyle\sum_{p_1|(R^{PPF}_{p,p_1,f} \text{ and } p_1 \notin P^{PFFvso}_{p_1,f,f} \text{ and } p_1 \notin P^{PFFvsi}_{p_1,f,f})} V_{p_1,p,f} * R^{PFF}_{p_1,f,f'} \\
+ & \qquad \textstyle\sum_{p'|(R^{PPF}p,p',f \text{ and } p_1 \in P^{PFFvsi2}_{p_1,f,f})} V_{p',p,f} * R^{PFF}_{p,f,f'} \\
= & \\
+ & \textstyle\sum_{p_1|(R^{PPF}_{p,p_1,f} \text{ and } p_1 \notin P^{PFFvsi}_{p_1,f,f} \text{ and } p_1 \notin P^{PFFvso}_{p_1,f,f})} V_{p_1,p,f} * R^{PFF}_{p_1,f,f'} \\
+ & \qquad \textstyle\sum_{p'|(R^{PPF}p,p',f \text{ and } p_1 \in P^{PFFvso2}_{p_1,f,f})} V_{p',p,f} * R^{PFF}_{p,f,f'}
\end{aligned} \tag{31}$$

Instances of any of the above three types of equations are generated only for those combinations of process and Flows that are relevant according to the input data. Thus, for instance there are no instances that involves combinations of Flows and Processes that are not linked through input data R^{PPF}.

The objective function (QOBJW in the GAMS code) is given as

$$V^{Idc} \quad = \quad \sum_{(p,f)|(p \in P^{Idc} \text{and } f \in F^{Idc})} P^{CIdc}_f * V^{Idc}_{p,f} \tag{32}$$

By choice of coefficients in the input parameter R^{CIdc} the relative attractiveness of the net Flows from P^{Idc} is specified. This objective function is actually one of several possible ones, it is characterized by being based on a weighting of the Flows. See further Section 3.6 page 24.

Further equations related to an objective function are QOBJQ and QOBJWQ, cf. page 24. They are meant for providing a more broad view of the range of solutions as a function of various variations.

3.9 Model and Solve

Model

In the GAMS language the word Model has the specific meaning of a collection of previously declared Equations. Hence, it is possible to declare more Equations than what are actually used in a specific model, and to specify several models from previously declared equations.

To specify the solution of the model the Solve statement is used, e.g.

SOLVE OPTIWASTEW Maximizing VOBJW USING LP

In this, VOBJW is the variable that holds the objective function value, cf. Section 2.8 page 16, and it is to be maximized (the alternative would be to specify Minimizing, however, this is not recommended here in OptiFlow, because there are then quite a number of consequences that would require attention). The model type is here specified to be LP (linear programming), relevant alternatives are e.g. MIP (Mixed Integer Programming) and QCP (Quadratically Constrained Programming). Such model types are standard within the optimization tradition, and an abundant literature is available. GAMS relies on separately developed solvers for solving the model.

Solve

The Solve statement triggers a number of actions by GAMS. First the model code and input data are verified for correctness and consistency. If errors are found the process stops with error messages. Otherwise, the model is transferred to the solver, and the solution process is started. When finished, GAMS reports on the solution status.

Solution values

If the solution was successful the values of the variables and equations will reported. Generally, each variable and equation will be reported to have two values at the optimal point: the level and the marginal value.

For a variable the level is the numerical value it attains, e.g. 1234.56 ton/h. For an equation the level is the numerical value it attains (evaluating the expression of the equation using the optimal values of the relevant variables).

The marginal value is an indication of the answer to the question, How much will optimal value change if we make a change of one unit of one of the right hand sides of an equation? If in particular the objective function is expressed in monetary terms the marginal value may be interpreted as a price (with dimension e.g. Money/ton or Money/MW).

Alternative expressions for level and marginal values are primal and dual values.

3.10 Files, Folders and Projects

The model is distributed over a number of (ascii) files in a folder hierarchy. In this section we give an overview.

The top level in the hierarchy is the Project folder. A Project folder (called e.g. MyOptiWasteProject1) holds all files and folders needed for making an analysis with OptiFlow.

At the second level is the base folder, herein you should find the folders indicated:

- ☐ MyOptiWasteProject1
  - ☐ base
    - ☐ addons - see further Section 3.15
    - ☐ data - all input data needed for running the OptiFlow model is located here, Section 3.11
    - ☐ model - here is found the main file OptiWaste.gms and additionally option files (optiwasteopt.opt and optiwastegams.opt) for selection of various details in model, solution and execution; also GAMS generated standard output (in particular the list file OptiWaste.lst) will be located here
    - ☐ logerror - the input data is screened for obvious errors, any observations will be reported here

- ☐ output - the User may (in the option file) select various types of output to be reported here

Some of the folders contain sub-folders, many of them with code to supplement the main code in OptiWaste.gms.

Projects and Cases

While the base folder is the place for holding all needed OptiFlow code it may also be seen as the folder that holds input data and associated output for a specific Case, the Base Case. Alternative Cases, differing with respect to input data, may be formulated within the Project. Hence, with one Base case and three alternative Cases, the first folder level below the Project folder may look like

- ☐ MyOptiWasteProject1
  - ☐ base
  - ☐ co2low
  - ☐ lowcost
  - ☐ recycle

Any non-base case has the following subfolders:

- ☐ data
- ☐ model
- ☐ logerror
- ☐ output (the output subfolder has additional subfolders)

Most non-base subfolders should hold no code. The exception is the model folder which should hold the two files balopt.opt and balgams.opt. Note in particular that a non-base Case should not hold the OptiFlow.gms file. A non-base Case should hold only the data files that differ in content from those of the base Case. The output from execution will be placed in appropriate folders of the non-base Case.

It is assumed that GAMS is run through the GAMS IDE (www.gams.com) and that a GAMS Project is created with the project file (extension: gpr) located in the base/model folder.

Running the **base** case implies that the option files optiwasteopt.opt and optiwastegams.opt are read, and dependent on chosen options the relevant code and data files will be included. All data files are located in the base/data folder. Output files go to subfolders of the base Case.

A **non-base** Case is more involved. Consider for specificity a Case called case3 (this is then also the name of the Case folder). The following happens:

- The option files balopt.opt and balgams.opt are read from case3/model. As minimum the balopt.opt should differ from that of the base Case by having option caseid set as $setglobal CASEID case3
- The data files in the case3/data folder are read. If the needed data files are not in that folder, they will be read form the base/data folder. This ensures that all input data are identical between base and case3, except for the files available in the case3/data folder
- Output will go to appropriate folders of case3
- However, special care is needed to ensure that the GAMS output files OptiFlow.lst and OptiFlow.log are sent to folder case3/model, see next.

Why this way
Advantages of the above handling are that there is a clear distinction between the different Cases, everything relevant and specific for a Case is in the Case folder. More specifically there are for any Case

- Documentation of the options used (found in balopt.opt and balgams.opt in the model folder of any Case)
- Documentation of the data (only files in case3/data differ from the base Case)
- Documentation of the output (OptiFlow.lst, OptiFlow.log plus any chosen output like error file, printfiles, gdx files, xls files. etc.)
- The OptiFlow code files exist only in the base Case hence exactly the same code is used for all Cases (except optiwasteopt.opt and optiwastegams.opt)
- Since the code files exist only in the base Case code maintenance is easier and safer.

3.11 Input data

Input data (sometimes referred to as being exogenous in contrast to values found through optimization which are then called endogenous) are found in the model folder. All data items are given as identifiers (in GAMS' sense) as sets, scalars or parameters. Values for each ID (identifier) is given in a separate file with the same name as the ID, and with extension .inc, e.g. file FLOW.inc contains the definitions of set FLOW while the file FLOWBOUND.inc contains the values of parameter FLOWBOUND. For units see Section 3.12 page 31.

Observe GAMS' application of default values, empty (for sets) and 0 (for parameters), hence, if no value is given, the default value is applied by GAMS.

OptiFlow specific default values are used whenever convenient, for this the GAMS special value EPS is used; see the description in the code.

3.12 Units

Any numerical value is given with respect to a certain unit, e.g. ”tonne per year”. It is therefore in general a good and useful habit to always specify the unit applied to any input data, intermediate data, variables, equations and output.

Due to the generality of OptiFlow there are certain complications. Much numerical data has a combination of units, e.g ”tonne/year” ”tonne/h”, ”kg/h”, ”Money/h”. Here, only the ”/” is common (or, more precisely, the ”/<some time unit>”). Other data has units like ”tonne”, ”MWh” or ”Money” (”Money” is used here as generic term, in the data file that the User enters it must be specified as e.g. DKK or Euro). So, in this perspective there are two types: those with, and those without the ”/”.

In intuitive terms, the units without the ”/” refer to quantities (e.g., kg, tonne, MJ, Money, MWh) while those with the ”/” refer to quantities-per-time-unit, or streams (e.g., kg/h, tonne/h, tonne/year, MJ/s or MW, Money/year). This distinction is also known in other contexts using terms as stock vs. flow, storage vs. stream flow or otherwise.

For an identifier (ID) with numerical values, i.e. a parameter, variable or equation, it will therefore in general be possible to specify the unit as being of only one of the two types. The unit will consequently be specified using either ”U” or ”U/T”; here ”U” indicates ”unit” while ”T” indicates ”time” (not ”tonne”, which will be abbreviated ”t”). The unit will be given inside a pair of parentheses, ”(U)” or ”(U/T)”. By convention, the unit is at the the very end of the description of the ID or elements in simple sets like e.g. FLOW below.

Flows will have unit type ”U/h” e.g. ”tonne/h”, ”MW”, ”Money/h”. However, the elements in this set, i.e., the individual Flows, can be given more specific information. Here is an example.

SET FLOW ”The set of all Flows”
/
HHWaste ”House hold waste (tonne/year)”
Moneystream ”Money/hour”
Elec ”Electricity (MW)”
Heat ”Heat (for district heating) (MW)”
/;

Examples of IDs that are in the ”U” type of unit are storage levels. Here the units will be e.g. ”tonne”, ”MWh” or ”Money”. Also Source, Sink and Buffer balancing variables may be of unit type ”U”. The objective function variables is of unit type ”U”.

There are some other specific IDs where the units will be pure numbers. For such cases the following convention for indications of units is

real for real numbers

int for integer

share where the numbers are in the closed interval [0; 1], i.e, such number x satisfies $0 \leq x \leq 1$. See FLOWSHAREBOUNDOUT for an example.

∼ See WEIGHT_S, IWEIGHTSUMS, YVALUE for examples.

***** where the units for an ID are not identical with respect to ”U” and ”U/T” for all elements in the ID, see PROCDATA, SOSIBU2INDIC for examples.

3.13 Naming conventions

We have tried to select identifier names for the various sets, parameters etc. to facilitate the recognition of the meaning from the name.

Prefixes on identifiers:

I: internal (set, scalar or parameter). Exceptions: INDIC, INDICLIMGOAL, etc.

V: variable (see also VQ)

Q: equation

VQ: slack variable related to equation Q

Suffixes: most often indicates level of detail with respect to indexes; some examples:

_T: the finest division of time is the subdivision of the season

_S or S: the finest division of time is the subdivision of the year into seasons

_Y or Y: the year, annual

_A, _R: geography detail Area and Region, respectively

_AS, _AST: combination of geography and time detail

File extensions:

gms: gams file, the main file

inc: include file

out: output file

gdx: GAMS data exchange file

opt: option file

mss: model and solver status print file

Upper case, lower case:

GAMS is not case sensitive, thus e.g. identifiers optiwaste, OptiWaste and OPTIWASTE are interpreted to be identical. But observe that the editor that the User applies may very well be case sensitive, and so may the operating system (e.g., UNIX, Linux).

The following conventions are used

folders/files : folder and file names are lower in case except OptiFlow.gms; and all files in folder data are in upper case before the dot and followed by extension 'inc'

identifiers : FLOW elements in UPPERCASE and PROC elements in lowercase

Some attentions should be paid to the sequence of indexes in IDs. This is motivated by recognizability and mnemotechnic. It is also motivated by code execution efficiency in GAMS; in GAMS later indexes run faster, this should be taken into account in the coding.

The index sequence mostly applied is (year) $<$ (geography) $<$ (proc and flow) $<$ (season) $<$ (time). Here, the '$<$' indicates "comes before".

3.14 Energy

The OptiFlow model may be ran as a stand-alone model contered mainly on waste. However, there is a strong coupling between waste and energy - presently already in some existing countries through e.g. incineration for generation of heat and electricity, and for the future the linkage based on waste potentially being a source for recycled materials as well as an energy source. Therefore there is a need for a suitably strong representation of the waste and energy sectors and their interplay.

Energy technologies may to some extent be modelled within the above Proc/Flow network framework as any other Proc. Here, some types of energy technologies will be explained this way, similar to the modelling applied in the Balmorel model.

Incineration plants may be modelled as Backpressure and Extraction type plants; heat may be supplied from such plant or from a Heat-only boiler, and electricity may be supplied from an electricity-only plant. In the sequel basic technology types for this will be modelled.

Figure 11 illustrates the feasible regions of combinations of F_e and F_h for some of the plant types used in Balmorel.

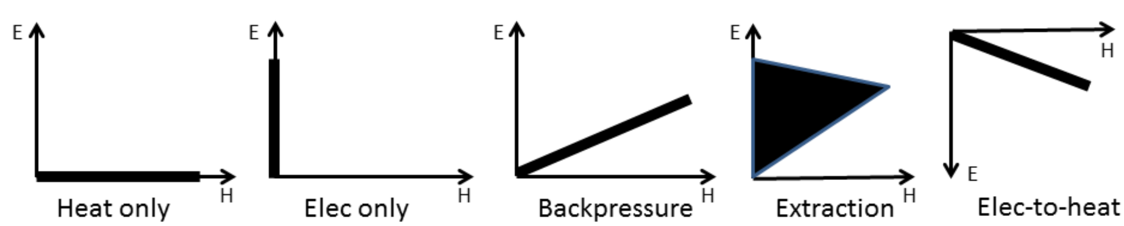

Figure 11: Operating regions for some of the energy plant types in Balmorel indicated by thick lines and solid triangle. 'H' is heat, 'E' is electricity.

Heat-only boiler type plant

Figure 11, Heat only, illustrates the heat-only plant type. In OptiFlow it is modelled as a Transform Process with heat as outFlow and some fuel as inFlow.

Electricity-only boiler type plant

Figure 11, Elec. only, illustrates the electricity-only plant type. In OptiFlow it is modelled as a Transform Process with electricity as outFlow and some fuel as inFlow.

Back pressure type plant

Figure 11, Backpressure, illustrates the feasible region of combinations of non-negative variables F_e and F_h for a Back pressure type plant.

The following relations uses positive coefficients c_b, η and C_e, where the first one is the slope of the non-vertical part of the feasible region, the second one is the energy efficiency, and C_e is the capacity (specified as maximum electricity output), to specify the relations between F_e, F_h and F_f (fuel input):

$$F_e = F_h c_b \tag{33}$$

$$F_f = \frac{F_e + F_h}{\eta} \tag{34}$$

$$F_e \leq C_e \tag{35}$$

Hence the Back pressure plant may be modelled in OptiFlow as a Transform Proc with heat electricity as outFlow and some fuel as inFlow.

Extraction type plant

Figure 11, Extraction, illustrates the feasible region of combinations of the non-negative variables F_e and F_h for Extraction type plant.

The following uses positive coefficients c_b, c_v, η and C_e, where the former two are the slopes of the non-vertical part of the feasible region, the third one is the energy efficiency, and C_e is the capacity (specified as maximum electricity output), to specify the following relations between the non-negative variables F_e (electricity generation), F_h (heat generation) and F_f (fuel input):

$$F_e \geq F_h c_b \tag{36}$$

$$F_e \leq C_e - F_h c_v \tag{37}$$

$$F_f = \frac{F_e + F_h c_v}{\eta} \tag{38}$$

$$F_e \leq C_e \tag{39}$$

Figure 11 illustrates the feasible region of combinations of F_e and F_h based on (36) - (37), combined with the assumption that F_e, F_h and F_f are all non-negative.

To model this Balmorel system in OptiFlow some manipulation is needed. Combining (37) and (38) yields

$$F_f \leq \frac{C_e}{\eta} \tag{40}$$

This provides the OptiFlow capacity bound that substitutes the Balmorel capacity bound (39). Note that while the capacity size in Balmorel is relative to an outFlow it is in OptiFlow relative to the inFlow.

Now the extraction technology may be modelled in OptiFlow as a VSplit Proc (cf. Eqn. (7) and Figure 4) with one inFlow F_f of fuel and two outFlow F_e (electricity) and F_h (heat) as in Eqn. (41)

$$F_f = \frac{1}{\eta} F_e + \frac{c_v}{\eta} F_h \tag{41}$$

with capacity constraint (40), with relation (36), and with non-negativity assumptions on the Flow.

Note that data for (36) is to be entered in QFLOWSHAREOUT2OUTLO_T or QFLOWSHAREOUT2OUTUP_T, (40) is to be entered in PROCKAPFX with specification (here and in PROCKAPDATA) that the capacity is on the inFlow.

Alternatively, the following relations may be derived,

$$F_e \geq \frac{\eta c_b}{c_v + c_b} F_f \tag{42}$$

and

$$F_h \leq \frac{\eta}{c_v + c_b} F_f \tag{43}$$

and any one of them may be used in either QFLOWSHAREOUT2INLO_T or QFLOWSHAREOUT2INUP_T to represent (36).

Elec-to-heat type plant

Figure 11, Elec-to-heat, illustrates the Elec-to-heat type plant that takes electricity as input and gives heat as output. It may for instance represent direct electric heating or a heat pump.

Fluctuating renewable energy

Fluctuating renewable energy input - from e.g. wind, run-of-river hydro, photovoltaic, wave, solar heat - may be readily implemented in OptiFlow using data structures indicated in previous parts of this document, and the same goes for hydro with storage.

Storage

Implementation in OptiFlow of Energy storage for heat, electricity or gas has already been described, cf. Section 2.6 page 16 .

3.15 Addons

Addons the OptiFlow model are pieces of code that provide extensions to the functionality for the core model version. They are by default excluded from being applied. Hence the code is in the model, but is active only when decided to be so.

Code for the addons are found under folder addons in the base folder (Section 3.10 page 29). The code is typically held in a number of files, and consists of declarations, definitions and code snippets. Each file is to be included at specific places in the main OptiFlow code using the $include statement. The files are included according to the value of an associated option given in file optiwasteopt.opt.

Here is a preliminary list

- Economy of scale (implying non-linearity)
- Unit commitment (implying non-linearity)

- Integration with Balmorel, cf. Section 2.9 page 16. Detailed documentation of the Balmorel model may be found at www.balmorel.com
- MCDA - deriving the Pareto optimal frontier i multi criteria analyses

4 List of IDs

The following is a list of most identifiers in OptiFlow. The most important IDs from a User's perspective have been dealt with in Section 3 starting page 18. Those not described there are mostly dealing with the internal working mechanism of OptiFlow, they are therefore more relevant for a Developer, and better studied along with the OptiFlow code. Identifiers very close to implementation and less general, e.g. aliases, acronyms and simple counters, are omitted here.

Name	Domain	Type	Unit	Page
Geography:				
CCCRRRAAA		Set		All geographical entities (CCC + RRR + AAA)
CCC	(CCCRRRAAA)	Set		All Countries
RRR	(CCCRRRAAA)	Set		All regions
AAA	(CCCRRRAAA)	Set		All areas
CCCRRR	(CCC,RRR)	Set		Regions in countries
RRRAAA	(RRR,AAA)	Set		Areas in regions
C	(CCC)	Set		Countries in the simulation
AAATRANSIT	(AAA)	Set		Areas that are for waste transit only
AAAIMPORT	(AAA)	Set		Areas that are for waste import only
IR	(RRR)	Set		Regions in the simulation
IA	(AAA)	Set		Areas in the simulation
ICA	(CCC,AAA)	Set		Relation of areas to countries in the simulation
IATRANSIT	(AAA)	Set		Areas in the simulation that are for waste transit only
IAIMPORT	(AAA)	Set		Areas in the simulation that are for waste import only
Time:				
YYY		Set		Years with data
Y	(YYY)	Set		Years in simulation
SSS		Set		All seasons
TTT		Set		All terms with Seasons
S	(SSS)	Set		Seasons in the simulation
T	(TTT)	Set		Terms within the Seasons in the simulation
WEIGHT_S	(SSS)	Par	length	Weight (relative length) of each season
WEIGHT_T	(TTT)	Par	length	Weight (relative length) of each time period
IHOURSINST	(SSS,T)	Par	hours	Length of time ST-segments (hours)
IWEIGHSUMS		Sca	hours	Sum of weight relations of the time of each S-segment in WEIGHT_S (hours)
IWEIGHSUMT		Sca	hours	Sum of weight relations of the time of each T-segment in WEIGHT_T (hours)
IWEIGHSUMTWT		Sca	hours	Length of time T-segment in single-T time scale (hours)
IHOURSINSTWT	(SSS,T)	Par	hours	Length of time S-segments in single-T time scale (hours)
IY4REMAINY	(Y)	Par	int	The number of remaining years in Y in the current model
IY4XREMAINY	(Y)	Par	int	The number of remaining years in Y for new transmission line in the current model
IY411NEXTY	(Y,Y)	Set		For any given element in IY411 (index 1): the next element in Y (index 2)
IY411PREVY	(Y,Y)	Set		For any given element in IY411 (index 1): the previous element in Y (index 2)
IORD_IY	(Y)	Par	int	Helper for kind of ord() on assigned sets
Processes and Flows:				
PROC		Set		All processes
PROCSOURCE	(PROC)	Set		Source PROC (no flow entering it)
PROCSINK	(PROC)	Set		Sink PROC (no flow leaving it)
PROCBUFFER	(PROC)	Set		Buffer PROC (flow entering and/or leaving)
PROCEXIM	(PROC)	Set		PROC for exchange possibility between areas
PROCSTORAGE	(PROC)	Set		Storage PROC with Seasonal balance, presently no up/lo
PROCSTORAGE_Y	(PROC)	Set		Storage PROC with Annual balance
PROC_T	(PROC)	Set		PROC that operate on high time resolution T
PROCSOLAR	(PROC)	Set		PROC that are solar DH production
AAAPROCKAPNEW	(AAA,PROC)	Set		Areas for possible location of new Proc capacity
FLOW		Set		All flows
TRANSDIST	(PROC,IAAAE,IAAAI)	Par	km	Distance between two areas (not necessarily symmetric) (km)
TRANSFLOWMAX	(YYY,IAAAE,IAAAI,PROC,FLOW)	Par	ton/h	Transport quantity maximum value (ton/h)
FLOWFROMTOPROC	(AAA,IPROCFROM,IPROCTO,FLOW)	Set		FLOW (index 4) from PROC (index 2) to PROC (index 3)
PROCINOUTFLOW	(AAA,PROC,IFLOWIN,IFLOWOUT,IPROCINOUTRELATION)	Par		Relationship at PROC (index 2) between FLOWIN (index 3) and FLOWOUT (index 4); interior PROC only (
PROCDATASET		Set		Process data types
PROCDATA	(PROC,PROCDATASET)	Par	*	Process data (*)
PROCKAPDATA	(PROC,FLOW,IPROCKAPFLOWINOUT)	Set		Process data: PROC capacity relative to FLOW is wrt. in or out direction (*)
PROCKAPFX	(YYY,AAA,PROC,FLOW,IFLOWINOUT)	Par	*	NO
FLOWINDIC	(FLOW)	Set		The indicator flows that we are interested in evaluating
PROCINDIC	(PROC)	Set		The indicator Proc (source/sink/buffer(/exim)) with net balances that we are interested in evaluating
SOSIBU2INDIC	(YYY,PROC,FLOW,FLOWINDIC)	Par	*	Coefficients for transformation of VSOURCE, VSINK and VBUFFER flows to FLOWINDIC values (*)
SOSIBU2INDIC_AST	(AAA,PROC,FLOW,FLOWINDIC,SSS,TTT)	Par	*	Coefficients for transformation of VSOURCE, VSINK and VBUFFER flows to FLOWINDIC values (*)
SOSIBU2INDIC_AS	(AAA,PROC,FLOW,FLOWINDIC)	Par	*	Coefficients for transformation of VSOURCE, VSINK and VBUFFER flows to FLOWINDIC values (*)
SOSIBU2INDIC_RST	(RRR,PROC,FLOW,FLOWINDIC,SSS,TTT)	Par	*	Coefficients for transformation of VSOURCE, VSINK and VBUFFER flows to FLOWINDIC values (*)

TRANSCOSTDATASET		Set		Transport cost element data and mapping to FLOWINDIC
TRANSDISTWEIGHT		Set		Internal set for handling of transport related cost (*)
TRANSFUEL	(YYY,TRANSCOSTDATASET,PROC,FLOW)	Par	*	Transport fuel use
TRANSCOST	(YYY,TRANSCOSTDATASET,TRANSDISTWEIGHT,PROC,FLOWINDIC)	Par	*	Transport cost in terms of INDIC (*)
ILOUPFXSET		Set		Set representing lower, upper and fixed values
SOSIBUBOUND	(AAA,PROC,FLOW,ILOUPFXSET)	Par	*	Bounds on Source, Sink and Buffer Process Flows - Annual values (*)
SOSIBUFLOW_VAR_T	(AAA,PROC,FLOW,SSS,TTT)	Par		Variation in some Source, Sink or Buffer Flows over the year ( )
ISOSIBUFLOW_SUMST	(AAA,PROC,FLOW)	Par		Annual time weighted amount of SOSIBUFLOW_VAR_T values ( )
FLOWBOUND	(YYY,AAA,IPROCFROM,IPROCTO,FLOW,ILOUPFXSET)	Par	U/h	Bounds on individual interior FLOWs [U/h]
FLOWSHAREOUT2IN	(AAA,IPROCFROM,IPROCTO,FLOW,ILOUPFXSET)	Par	share	Bounds on shares (of inFlow) of variable splits of outflow from PROC (share)
FLOWBOUNDSHAREIN	(AAA,IPROCFROM,IPROCTO,FLOW,ILOUPFXSET)	Par	share	Bounds on shares of sum of inflow to PROC (share)
FLOWSHAREOUT2OUT	(AAA,PROC,IFLOWIN,IFLOWOUT,IFLOWOUT2,ILOUPFXSET)	Par	share	Fixed relation (index 4 divided by 5) between two split FLOWs leaving PROC (share)
SOLHFLH	(AAA)	Par	h	Full load hours of operation of solar DH plants in every AAA (h)
RATEOFRETURN		Sca		Rate of return of the investments ( )
IPROCINDIC	(PROCINDIC,FLOWINDIC)	Set		Set of relevant combinations of (PROCINDIC ,INDIC)
IINDICLIMGOALSET		Set		Limits and desirable values of INDIC
INDICLIMGOAL	(FLOWINDIC,IINDICLIMGOALSET)	Par	*	Limits, desirable quantities and weight of INDIC (*)
ITWITHPROC	(PROC,T)	Set		The relevant T for any PROC - differs between PROC in/not-in PROC_T
IPROCINTERIOR	(PROC)	Set		Set of PROC that are neither Sources nor Sinks nor Buffers nor Exims nor PROCSTORAGE nor PROCSTOR
IY4GREMAINY	(Y,PROC)	Par	int	The number of remaining years in Y for new technology in the current Balbase4 model
IAPROCKAPNEW	(Y,AAA,PROC)	Set		Area, Proc where technology may be invested based on APKN and some implicit constraints
ITRANSFROMTO	(IAAAE,IAAAI)	Set		Set of Areas that are connected by transport (not necessarily symmetric)
IPROCKAPFLOWINOUT		Set		
IPROCINOUTRELATION		Set		Possible relationships between process inflow and outflow
IPROCINOUTFLOW	(AAA,PROC,IFLOWIN,IFLOWOUT)	Set		Set of combinations of (AAA,PROC,IFLOWIN,IFLOWOUT) in PROCINOUTFLOW
IPRIOONEMANY	(AAA,PROC,IFLOWIN,IFLOWOUT)	Par		Indicator for use with onemany (VSPLIT): 0: not onemany, 1: onemany primary, 2: onemany secondary; only
IPRIOMANYONE	(AAA,PROC,IFLOWIN,IFLOWOUT)	Par		Indicator for use with manyone (VJOIN): 0: not manyone, 1: manyone primary, 2: manyone secondary; only p
ILEAVEPROC	(AAA,PROC,FLOW)	Set		For each PROC: the set of FLOW that originate from this PROC (based on FLOWFROMTOPROC)
IENTERPROC	(AAA,PROC,FLOW)	Set		For each PROC: the set of FLOW that enter this PROC (based on FLOWFROMTOPROC)
INEGPROCININFLOW	(AAA,PROC,IFLOWIN,IFLOWIN2)	Set		Set with two incoming FLOW to a PROC (negative value of PROCINOUTFLOW)
INEGPROCOUTOUTFLOW	(AAA,PROC,IFLOWOUT,IFLOWOUT2)	Set		Set with two outcoming FLOW to a PROC (negative value of PROCINOUTFLOW)
IFLOWEXIM	(FLOW)	Set		Flows that may transported between areas
ITRANSFLOWFROMTO	(IAAAE,IAAAI,FLOW)	Set		Set of Area pairs that are connected by transport of FLOW in direction from index1 to index2
Energy system:				
GDATASET		Set		Energy generation technology data
IGCND	(G)	Set		Condensing technologies
IGBPR	(G)	Set		Back pressure technologies
IGEXT	(G)	Set		Extraction technologies
IGHOB	(G)	Set		Heat-only boilers
IGETOH	(G)	Set		Electric heaters, heatpumps, electrolysis plants
IGHSTO	(G)	Set		Heat seasonal storage technologies
IGHHNOSTO	(G)	Set		Technologies generating heat-only, except storage
IGEE	(G)	Set		Technologies generating electricity only
IGHH	(G)	Set		Technologies generating heat only
IGEH	(G)	Set		Technologies generating electricity and heat
IGE	(G)	Set		Technologies generating electricity
IGH	(G)	Set		Technologies generating heat
IAGK_Y	(AAA,G)	Set		Area, technology with positive capacity current simulation year
DH	(YYY,AAA)	Par	(MWh)	Annual heat consumption
Variables:				
VOBJW		fVar		
VOBJQ		fVar		
VOBJWQ		fVar		
VFLOWINDICVALUE	(Y,FLOWINDIC)	fVar	U	Quantities of indicators (time weighted) (U)
VFLOW	(Y,AAA,IPROCFROM,IPROCTO,FLOW,S,T)	pVar	U/h	FLOW endogenous quantities between two PROC (U/h)
VFLOWSOURCE	(Y,AAA,PROCSOURCE,FLOW,S,T)	pVar	U/h	FLOW quantities from Source (U/h)
VFLOWSINK	(Y,AAA,PROCSINK,FLOW,S,T)	pVar	U/h	FLOW quantities to Sink (U/h)
VFLOWBUFFER	(Y,AAA,PROCBUFFER,FLOW,S,T)	fVar	U/h	Net FLOW quantities to Buffer (U/h)
VFLOWTRANS	(Y,IAAAE,IAAAI,PROC,FLOW,S,T)	pVar	ton/h	Transport quantities (ton/h)
VSTORAGEVOL	(Y,AAA,PROCSTORAGE,FLOW,S,T)	pVar	U	Content in PROCSTORAGE (U)
VSTORAGEVOL_Y	(Y,AAA,PROCSTORAGE_Y,FLOW,S)	pVar	U	Content in PROCSTORAGE_Y (U)
VQPROCBALANCE	(Y,AAA,PROC,IFLOWIN,IFLOWOUT,S,T,IPLUSMINUS)	pVar	U/h	Feasibility-ensuring variable with high penalty cost - values are all zero in a feasible model instance (U/h)
VQPROCSOURCEBALANCE	(Y,AAA,PROCSOURCE,FLOW,S,T,IPLUSMINUS)	pVar	U/h	Feasibility-ensuring variable with high penalty cost - values are all zero in a feasible model instance (U/h)
VQPROCSINKBALANCE	(Y,AAA,PROCSINK,FLOW,S,T,IPLUSMINUS)	pVar	U/h	Feasibility-ensuring variable with high penalty cost - values are all zero in a feasible model instance (U/h)

VQPROCBUFFERBALANCE	(Y,AAA,PROCBUFFER,FLOW,S,T,IPLUSMINUS)	pVar	U/h	Feasibility-ensuring variable with high penalty cost - values are all zero in a feasible model instance (U/h)
VQFLOWSHAREOUT2INUP	(Y,AAA,PROC,IFLOWIN,IFLOWOUT,S,T,ILOUPFXSET,IPLUSMINUS)	pVar	U/h	Feasibility-ensuring variable with high penalty cost - values are all zero in a feasible model instance (U/h)
VQFLOWSHAREOUT2INLO	(Y,AAA,PROC,IFLOWIN,IFLOWOUT,S,T,ILOUPFXSET,IPLUSMINUS)	pVar	U/h	Feasibility-ensuring variable with high penalty cost - values are all zero in a feasible model instance (U/h)
VQFLOWSHAREOUT2INFX	(Y,AAA,PROC,IFLOWIN,IFLOWOUT,S,T,ILOUPFXSET,IPLUSMINUS)	pVar	U/h	Feasibility-ensuring variable with high penalty cost - values are all zero in a feasible model instance (U/h)
VQHEQ	(Y,AAA,S,T,IPLUSMINUS)	pVar	U/h	Feasibility-ensuring variable with high penalty cost - values are all zero in a feasible model instance (U/h)
VPROCKAPACCUMNET	(Y,AAA,PROC)	pVar	MW	Accumulated new investments (Presently not: minus any decommissioning of them due to lifetime expiration)
VPROCKAPNEW	(Y,AAA,PROC)	pVar	/h	New capacity (/h)
VGE_T	(Y,AAA,G,S,T)	pVar	MW	Electricity generation (MW), existing units
VGH_T	(Y,AAA,G,S,T)	pVar	MW	Heat generation (MW), existing units
Equations:				
QOBJW		Equ	U	Objective function value assuming weighted objectives values technique (U)
QOBJQ		Equ	U	Objective function value assuming parameterised quantities technique (U)
QOBJWQ		Equ	U	Objective function value assuming weighted objectives values and parameterised quantities technique (U)
QPROCSOURCEBALANCE	(Y,AAA,PROCSOURCE,FLOW,S,T)	Equ	U/h	Balance at Source node (U/h)
QPROCSINKBALANCE	(Y,AAA,PROCSINK,FLOW,S,T)	Equ	U/h	Balance at Sink node (U/h)
QTEST_ITWITHPROC	(Y,AAA,PROCBUFFER,FLOW,S,T)	Equ		
QPROCBUFFERBALANCE_S	(Y,AAA,PROCBUFFER,FLOW,S,T)	Equ	U/h	Heat balance in district heating networks - NEJ generel, but includes Heat balance.. (U/h)
QPROCBUFFERBALANCE_T	(Y,AAA,PROCBUFFER,FLOW,S,T)	Equ	U/h	Heat balance in district heating networks - NEJ generel, but includes Heat balance.. (U/h)
QSTORAGEBALANCE	(Y,AAA,PROCSTORAGE,FLOW,S,T)	Equ	U	Balance at TimeLinkstorage node (U)
QSTORAGEBALANCE_Y	(Y,AAA,PROCSTORAGE_Y,FLOW,S)	Equ	U	Balance of Annual Storage (U)
QSTORAGEVOLLIM	(Y,AAA,PROC,FLOW,S,T)	Equ	U	Weekly storage capacity limit (U)
QSTORAGEVOLLIM_Y	(Y,AAA,PROC,FLOW,S)	Equ	U	Seasonal storage capacity limit (U)
QGSOLAR	(Y,AAA,PROCSOLAR,PROC,FLOW,S,T)	Equ	MWh/h	Undispatchable production from solar DH plants (MWh/h)
QFLOWSHAREOUT2INUP	(Y,AAA,PROC,IFLOWIN,IFLOWOUT,S,T)	Equ	share	Maximum shares (of inFlow) of split FLOWs leaving PROC as outFlow (share)
QFLOWSHAREOUT2INLO	(Y,AAA,PROC,IFLOWIN,IFLOWOUT,S,T)	Equ	share	Minimum shares (of inFlow) of split FLOWs leaving PROC as outFlow (share)
QFLOWSHAREOUT2INFX	(Y,AAA,PROC,IFLOWIN,IFLOWOUT,S,T)	Equ	share	Fixed shares (of inFlow) of split FLOWs leaving PROC as outFlow (share)
QFLOWSHAREOUT2OUTLO_T	(Y,AAA,PROC,IFLOWIN,IFLOWOUT,IFLOWOUT2,S,T)	Equ	share	Lower value of relation (index 4 divided by 5) between two split FLOWs leaving PROC (share)
QFLOWSHAREOUT2OUTUP_T	(Y,AAA,PROC,IFLOWIN,IFLOWOUT,IFLOWOUT2,S,T)	Equ	share	Upper value of relation (index 4 divided by 5) between two split FLOWs leaving PROC (share)
QFLOWSHAREOUT2OUTFX_T	(Y,AAA,PROC,IFLOWIN,IFLOWOUT,IFLOWOUT2,S,T)	Equ	share	Fixed relation (index 4 divided by 5) between two split FLOWs leaving PROC (share)
QINDICINDICMAX	(Y,AAATRANSIT,PROC,FLOW,S,T,Y,FLOWINDIC,Y,FLOWINDIC)	Equ	U	Maximal annual quantity (sum-of-years) of INDIC (U)
QPROCKAPACCUMNET	(Y,AAA,PROC)	Equ	U/h	NOT FINISHED Accumulated new investments minus decommissioning of previous investments due to lifetim
QPROCKAP_UP	(Y,AAA,IPROCFROM,IPROCTO,FLOW,S,T)	Equ	U/h	Capacity constraint on non-storage Proc (U/h)
Models:				
OPTIWASTEW		Mod		
Various:				
IOF1000		Sca	real	Multiplier 1000
IOF1000000		Sca	real	Multiplier 1000000
IOF0001		Sca	real	Multiplier 0.001
IOF0000001		Sca	real	Multiplier 0.000001
IOF3P6		Sca	real	Multiplier 3.6
IOF24		Sca	real	Multiplier 24
IOF8760		Sca	real	Multiplier 8760
IOF8784		Sca	real	Multiplier 8784
IOF365		Sca	real	Multiplier 365
IOF1_		Sca	real	Multiplier 1 (special, posssibly to disappear in future versions
ISCALAR1		Sca	*	(Context dependent)
ISCALAR2		Sca	*	(Context dependent)
ISCALAR3		Sca	*	(Context dependent)
IGDXDIFFSET		Set		For use with the GDXDIFF utility
IPLUSMINUS		Set		Violation of equation - positive or negative direction, respectively
IPENALTYQ		Sca	real	Penalty on violation of equation
ILOUPBOUNDS		Sca	real	Default lower and upper bounds to guard against unbounded solution

Table 6: Identifiers in OptiWaste. Column Types refers to GAMS types: Set is set, Par is parameter, Sca is scalar, fVar, pVar are free and positive variables, Equ is equation, Mod is model.

Index