

TIMES Objective Function Formulations

Antti Lehtilä

(Updated: Aug 31, 2008)

Contents

1	Introduction.....	2
2	Problems in the Standard Formulation.....	3
2.1	Period definitions.....	3
2.2	Problems occurring without discounting.....	3
2.3	Additional problems under discounting.....	5
3	Alternative Formulations.....	6
3.1	The MOD formulation.....	6
3.1.1	General.....	6
3.1.2	Model periods.....	6
3.1.3	Investment spreads.....	6
3.1.4	Summary of changed model generator files.....	7
3.2	The ALT formulation.....	8
3.2.1	General.....	8
3.2.2	Investment spreads.....	8
3.2.3	Capacity transfer coefficients.....	9
3.2.4	Synchronizing variable costs.....	10
3.2.5	Summary of changed model generator files.....	11
3.3	The LIN formulation.....	12
3.3.1	General.....	12
3.3.2	Variable costs.....	12
3.3.3	Other equations.....	12
3.3.4	Summary of changed model generator files.....	13
4	Benchmark Comparison of Formulations.....	14
4.1	Methodology.....	14
4.2	Description of the example model.....	14
4.3	Benchmark results.....	15
4.3.1	Example 1.....	15
4.3.2	Example 2.....	16
4.3.3	Example 3.....	17
4.3.4	Example 4.....	18
4.4	Conclusions from the benchmark exercise.....	19
5	Using the Alternative Formulations.....	20
5.1	Control switches.....	20
5.2	Impacts on results reporting.....	20
	Appendix 1: The Benchmark Example Model.....	22

1 Introduction

The purpose of the TIMES objective function is to describe in detail the criterion that is minimized by the TIMES model. It represents the total discounted cost of the entire, possibly multi-regional system over the selected planning horizon. It is also equal to the negative of the discounted total surplus (plus a constant), as discussed in the documentation, Part I, chapters 3 and 6.

The TIMES objective function includes a number of innovations compared to those of more traditional energy models such as MARKAL, EFOM, MESSAGE, etc. The objective function is quite complex and detailed, accounting all the various cost components by each individual year within the model horizon.

However, some anomalies have been identified in the standard formulation for the objective function, which is described in detail in the documentation. These anomalies are mostly related to variable period lengths and the accounting of the salvage value of investment costs in the cases with incremental investment spreads (1a and 1b).

This note describes modifications to the standard objective function formulations that can be used to eliminate the problems identified. The modifications have been implemented into the GAMS code of the model generator in the form of three distinct alternative formulations, which can be selected by the user to be applied in the model. The following acronyms are used for the alternative formulations:

- ✓ MOD – Modified objective formulation (minimal consistent modifications for achieving smoother periods and investment spread);
- ✓ ALT – Alternative objective formulation (comprehensive elimination of the problems identified, but retaining the original paradigm of constant activities within each period);
- ✓ LIN – Linearized objective formulation (comprehensive elimination of the problems identified, assuming linear evolution of flows and activities).

Both the problems identified in the standard objective formulation and the approaches used in the alternative formulations are described in this note. Finally, a benchmark exercise based on a very simple model is presented, providing an illustrative comparison of the formulations.

2 Problems in the Standard Formulation

2.1 Period definitions

The standard objective function formulation is based on certain strict assumptions concerning the model periods. These assumptions are not imposed by the model generator, but the internal consistency of the standard objective formulation depends on complying with them. According to the convention, the milestone year of each period should be at the middle of the period years. Assuming that M_t denotes the milestone year of period t , B_t denotes the first year of the period, E_t denotes the last year of the period, and $\text{FLOOR}(x)$ denotes the largest integer less than or equal to the operand x , the rule can be expressed as follows:

$$M_t = \text{FLOOR}((B_t + E_t)/2), \text{ for all } t = 1, \dots, T \quad (1)$$

Although as such this assumption does not lead to any problems in the cost accounting of the objective function, it can lead to large variations in the lengths of successive periods, which in turn can cause problems as described below. In addition, this assumption prohibits the use of some sequences of milestone years, which means that the milestone years cannot be freely defined by the user. A simple example of an impossible sequence of milestone years is (2000, 2003, 2010, 2013).

2.2 Problems occurring without discounting

The standard objective formulation has been quite thoroughly designed to be economically consistent *with and without discounting*. However, there are some cases where the standard formulation leads to distortions in the cost accounting between capacity-related costs and the corresponding activity-related costs even *without discounting*. These distortions occur only at the end of the model horizon, either due to excessive or deficient salvage value.

Examples of the different types of distortions are illustrated in Figures 1 and 2. Figure 1 illustrates a case where the duration of the investment period i is two years, and the investment spread extends to the end of horizon so that only the second investment step exceeds the horizon. In this case it is clear that the second investment step starts at B_i (see documentation for details) and therefore the salvage value should correspond to one year's time at the end of the lifetime. However, the salvage value is calculated on the basis of the investment payments occurring beyond EOH only, and therefore the salvage value is only half of the correct value (without discounting).

An example of overestimating the salvage value is illustrated in Figure 2. In this case the investment spread consists of five steps, of which two extend beyond the model horizon. The resulting salvage value corresponds to $3/5 =$

TIMES Version 2.8 User Note

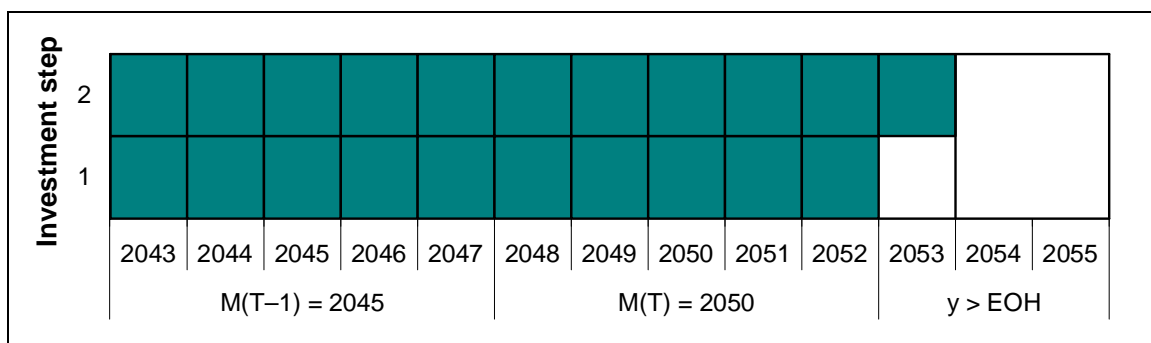


Figure 1. Example of over-estimation of investment costs in the standard formulation: Salvage value should correspond to one year of the lifetime, but only half of that is credited in the standard formulation (one year from the second investment step).

0.6 years of lifetime. However, from Figure it is clear that the salvage value should actually be zero, because the capacity is assumed to be fully available still in the last period. The resulting deficit in the investment (and fixed) costs, as accounted by the standard formulation, is shown in yellow.

In addition to these clear cost accounting problems at the end of horizon, the investment spreads used in the standard formulation can also lead to other cost distortions, regardless of discounting. In very long periods, the investment spreads are divided into D_t successive steps, each amounting to $1/D_t$ of the total capacity to be invested in the period. For example, if the period length D_t is 20 years, the investments start already 19 years before the milestone year, and can thus start even well before the previous milestone year. If the investment costs are changing over time, it is clear that in such cases the costs will not be accounted in a realistic way, because the investment cost data is taken from the start year of each investment step.

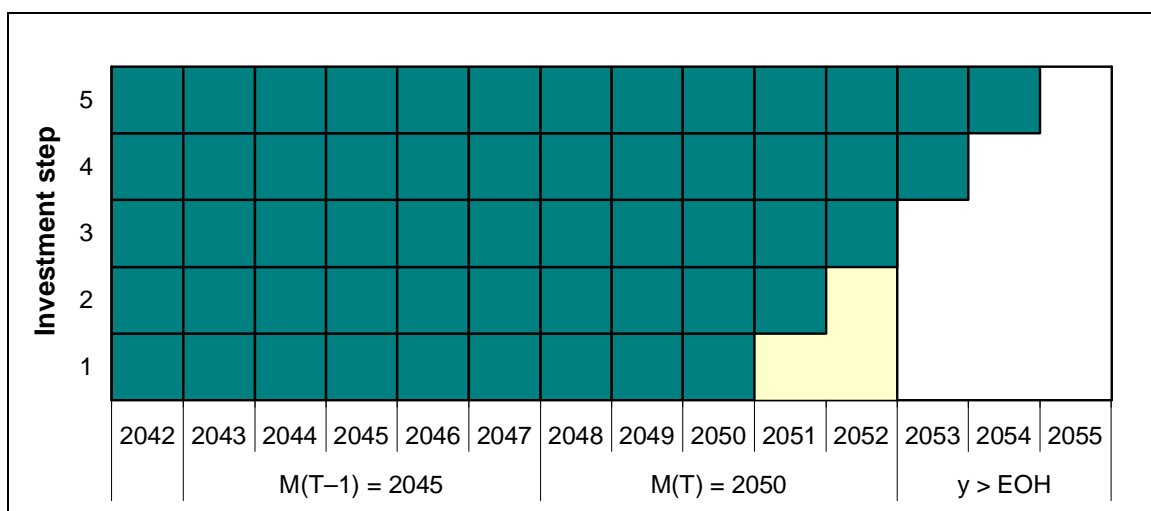


Figure 2. Example of under-estimation of investment costs in the standard formulation: No salvage value should be credited, but the standard formulation assumes the annual payments occurring at $y > EOH$ to be salvaged. The resulting cost deficit is shown in yellow.

TIMES Version 2.8 User Note

Similarly, in short periods the investment costs are spread over only a few years, and if the previous period is much longer, this can leave a considerable gap in the investment years between successive periods. Here again, if the investment costs are changing over time, this would lead to a distortion in the cost accounting.

2.3 Additional problems under discounting

When using equal period lengths, the investment spreads in Case 1a represent linearly evolving capacities between milestone years. Such linear evolution might thus also be considered as the most natural assumption for the flow levels of processes. However, in the standard formulation the investments in new capacity are not in sync with the development of activities and flows. Even though the amount of new capacity is assumed to increase gradually along with the investment spreads, the production levels are always assumed to be constant within each period. In addition, the proportion of installed capacity available in the last period of its lifetime is calculated as an undiscounted proportion of time, which causes an additional distortion under discounting, because the value of the capacity then decreases over time.

Under discounting the differences between the evolution of capacities and activities cause small distortions in the cost accounting, as the investment costs become more front-heavy than the activity costs due to the discounting. The differences become more pronounced when some or all of the periods include an even number of years, because in such periods the middle point of the incremental investment spreads does not coincide with the beginning of the period, but occurs half a year earlier in time.

As already mentioned, the basic assumptions of the standard formulation (Eq. (1)) can also easily lead to a large variation in the period lengths. Such a variation can further increase the cost distortions under discounting due to the larger differences in the timing of the available capacity (as defined by the investments) and assumed constant activity levels in each period.

3 Alternative Formulations

3.1 The MOD formulation

3.1.1 General

The MOD formulation should be considered only as a first attempt to eliminate the various small problems related to the standard formulation. Consequently, the MOD formulation helps to avoid only some of the problems. More complete elimination of the problems can be achieved by the ALT and LIN formulations, which include all the modifications already made in the MOD formulation.

The MOD formulation adds only a few modifications to the standard formulation:

- The model periods are defined in a different way; and
- The investment spreads in the investment Cases 1a and 1b are defined slightly differently.

3.1.2 Model periods

In the MOD formulation, milestone years no longer need to be in the middle of their periods. Instead, B_t and E_t are internally defined, so that the periods turn halfway between the milestone years. Therefore, only the first B_t and the last E_t need to be specified by user. B_t and E_t are defined in the following way:

$$\begin{aligned} E_t &= \text{FLOOR}((M_t + M_{t+1})/2), \text{ for all } i = 1, \dots, T-1 \\ B_t &= E_{t-1} + 1, \text{ for all } i = 2, \dots, T \end{aligned} \quad (2)$$

where M_t is the milestone year of period t .

3.1.3 Investment spreads

In the MOD formulation, investment spreads for Case 1.a and 1.b are defined slightly differently from the standard assumptions. In Case 1.a the investment steps are spread over the years $M_{t-1} + 1, \dots, M_t$ for each milestone t . Each of these steps is then additionally spread into ELIFE annual payments, just like in the standard formulation.

In Case 1.b, denoting by IPD_t the number of years between the milestone years ($IPD_t = M_t - M_{t-1}$), the modification applies only to the case where TLIFE is greater than IPD_t . In this case the spread for the first repeated cycle is

TIMES Version 2.8 User Note

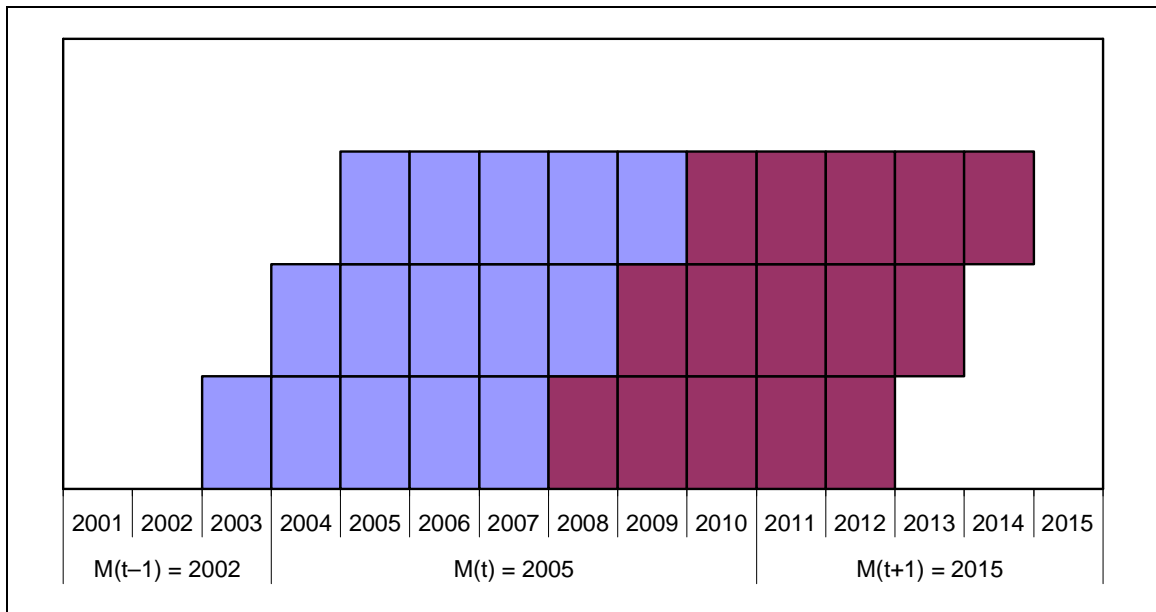


Figure 3. Example of modified investment spreads when $IPD_t=3$ and $TLIFE > IPD_t$ ($ELIFE=TLIFE$ assumed here, but $ELIFE$ can of course be different from $TLIFE$).

made similar to that in Case 1.a, and subsequent cycles appended also have the same spread form. An example of the resulting overall spread of the investment payments is shown in Figure 3, assuming in the example for simplicity that $ELIFE=TLIFE$.

The adjustments in the investment cycles are motivated by the resulting smoother development of the investment costs. With the modifications, the investment costs related to period t will in almost all cases start right after the previous milestone year, and will never start before that.

3.1.4 Summary of changed model generator files

The implementation of the MOD formulation includes changes in the following GAMS code files of the TIMES model generator:

- INITMTY.MOD (control switches)
- PPMAIN.MOD (period definition, COEF_RPTI definition)
- EQOBJINV.MOD (investment spreads)
- EQOBJFIX.MOD (investment spreads)

3.2 The ALT formulation

3.2.1 General

The ALT formulation attempts to correct all of the problems identified in the standard formulation, while maintaining the original paradigm, according to which the activities and flows are constant within each period.

The ALT formulation includes all the modifications made in the MOD formulation. In addition, it includes the following further modifications that eliminate basically all of the remaining problems in the standard formulation:

- The investment spreads in the investment Case 1b are defined slightly differently;
- The capacity transfer coefficients for newly installed capacities are defined slightly differently, so that the effective lifetime of technologies is calculated taking into account discounting;
- Variable costs are adjusted to be in sync with the available capacity.

3.2.2 Investment spreads

In the ALT formulation, the investment spreads for Case 1.a are identical to those in the MOD formulation.

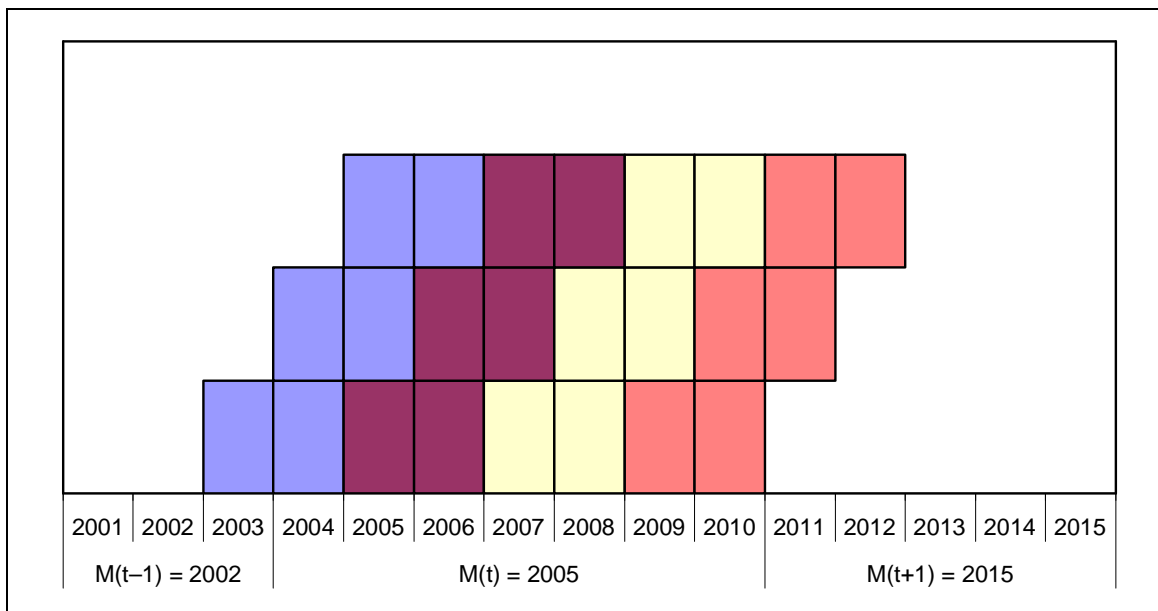


Figure 4. Example of modified investment spreads when $IPD=3$ and $TLIFE < IPD$ ($ELIFE=TLIFE$ assumed here, but $ELIFE$ can of course be different from $TLIFE$).

TIMES Version 2.8 User Note

In the investment Case 1.b, the first repeated cycle of the investment spreads is always made to be similar to that in Case 1.a, and the subsequent cycles to be appended also have the same spread form. Examples of the resulting overall spreads of the investment payments are shown in Figure 3 (for $TLIFE > IPD$) and Figure 4 (for $TLIFE < IPD$), assuming for simplicity that $EFILE = TLIFE$ in the examples.

3.2.3 Capacity transfer coefficients

The adjustments of the capacity transfer coefficients have two purposes:

- They take into account the impact of discounting in the value of the remaining capacity in each period;
- They eliminate the over- and underestimation of investment costs caused by the crediting of the salvage value at the end of the model horizon.

In simplified terms, the adjustment of the capacity transfer coefficients is done by comparing the remaining discounted technical lifetime that falls within the model horizon to the discounted time in each of the remaining model periods, one by one. If the remaining discounted lifetime is greater than the discounted time in the current period, the capacity is fully available in that period and the capacity transfer coefficient is set to 1. The discounted value of the current period is then subtracted from the remaining discounted lifetime. If at some step the remaining lifetime becomes smaller than the discounted time in the current period, that will be the last period where the technology is available, and the capacity transfer coefficient is set to the fraction of the discounted remaining lifetime to the discounted time in the period.

As described above, the adjustments will usually only affect the coefficient at the end of the technical lifetime of each technology, but also at the beginning of the lifetime, if *NCAP_ILED* is used (Cases 2a and 2b). An illustration of the contribution of discounting in the adjustments is shown in Figure 5, assuming that the discount rate is 7%, and the end of the technical lifetime falls within a 10-year period. The Figure shows the comparison between the coefficients according to the standard formulation and the adjusted coefficients.

On a more detailed level the algorithm also takes into account the detailed investment spreads. Therefore, the initial calculation of the discounted lifetime is somewhat more complex. The harmonization of the investment spreads between the Cases 1a and 1b described above was partly done in order to be able to adjust all the capacity transfer coefficients in a fully consistent way.

The adjustments in the capacity transfer coefficients eliminate the problems that occur in the standard formulation already without discounting (see Section 2.2). In addition, they also eliminate part of the distortions between capacity-related costs and activity-related costs under discounting.

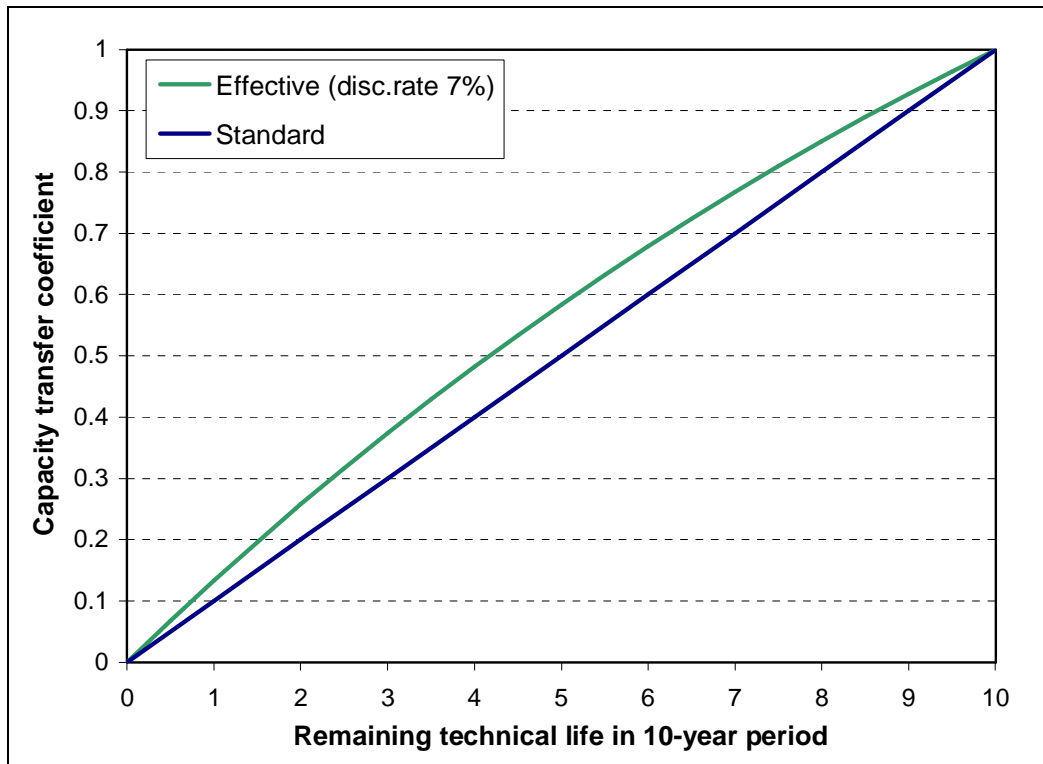


Figure 5. Simplified illustration of the impacts of discounting on the capacity transfer coefficients, assuming that the technical lifetime ends within a 10-year period. The adjusted effective coefficients are somewhat larger than the standard ones.

3.2.4 Synchronizing variable costs

With the adjustments of the capacity transfer coefficients, all the investment and fixed O&M costs are accounted in a *uniform* manner, so that in all investment cases the capacity value is consistent with the basic investment Case 1a. Thus, a *linear build-up* of available capacity between milestone years is used as the *reference*. In order to ensure consistent accounting of the variable costs as well, the variable costs should also be fully in sync with the available capacities and thereby the corresponding activities in each period.

In the ALT formulation, the final synchronization of variable costs with the available capacities is accomplished in the objective function in the following simple method: The discounted variable costs in each period are scaled by the ratio of the discounted area of the period *trapezoid* to the discounted *rectangular* period pulse function. This adjustment effectively calculates the *levelized variable costs* within each period, and then multiplies the levelized costs by the discounted sum of the activity within the period (as defined by the capacity availability), which is represented by the period trapezoid. The resulting adjusted variable costs are equivalent to the costs in a slightly adjusted rectangular pulse of period activity levels, as illustrated in Figure 6.

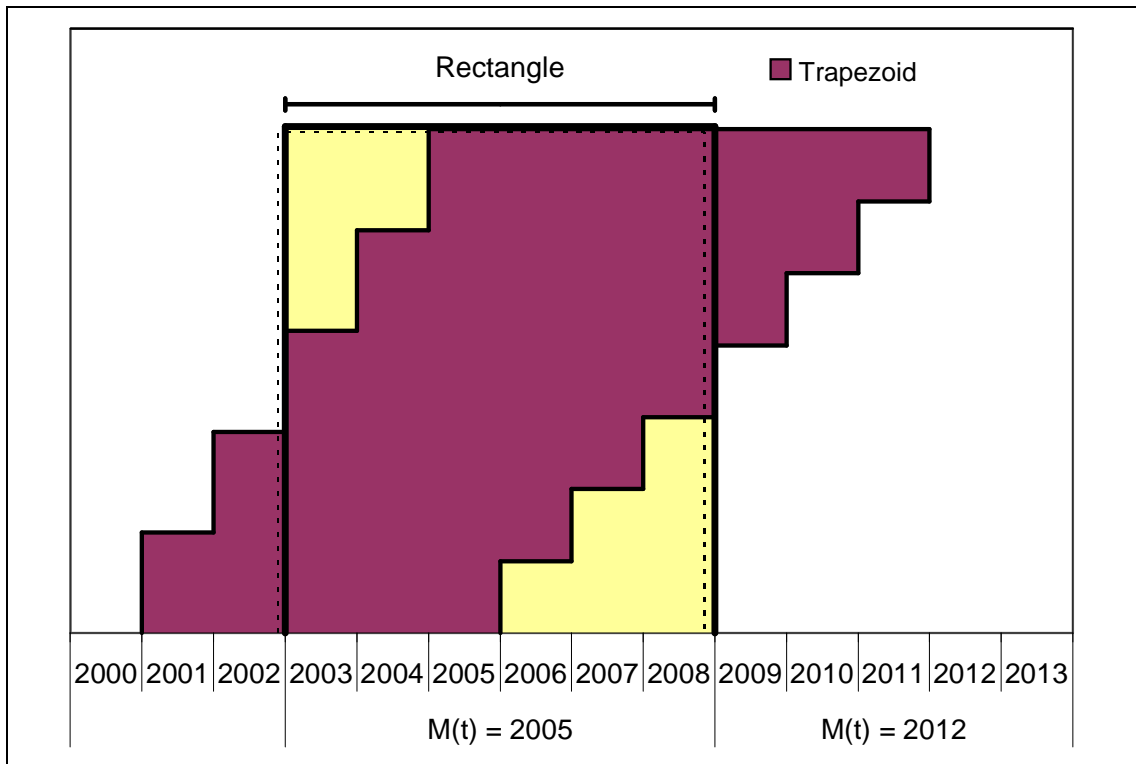


Figure 6. Example of a period rectangle (solid line) and the corresponding trapezoid, which is used for the synchronizing of variable cost with the available capacities. The resulting adjusted period rectangle is shown by the dashed line. Note that the undiscounted area of the original period rectangle is equal to the area of the trapezoid.

The synchronized periods adjusted according to this method still cover exactly all the original years within the model horizon, and therefore the adjustment is fully consistent with the original model specification.

3.2.5 Summary of changed model generator files

The implementation of the ALT formulation includes changes in the following GAMS code files of the TIMES model generator:

- INITMTY.MOD (control switches)
- PPMAIN.MOD (period definition, COEF_RPTI + RTP_CPTYR def.)
- COEFMAIN.MOD (call for COEF_ALT.LIN added)
- EQOBJINV.MOD (investment spreads)
- EQOBJFIX.MOD (investment spreads)
- EQOBJVAR.MOD (variable cost synchronization)

In addition, the implementation of the ALT formulation is also dependent on the following new GAMS code file (calculation of adjusted coefficients):

- COEF_ALT.LIN

3.3 The LIN formulation

3.3.1 General

Like the ALT formulation, also the LIN formulation attempts to correct all of the problems identified in the standard formulation, but by using a different approach for the final synchronization of capacities and activities. The LIN formulation assumes that activities and flows evolve linearly between milestone years. The formulation thus no longer maintains the original paradigm of the standard formulation, according to which the activities and flows should be constant within each period.

Apart from the variable cost adjustment, the LIN formulation includes all the other modifications of the ALT formulation. But instead of using period-wise adjustment coefficients for the variable costs, the final synchronization is accomplished by assuming linearly evolving activities and flows. Therefore, additional modifications are made to the calculation of all variable cost components in the objective function, as well as to all other dynamic equations involving activities or flows.

Both the ALT and LIN formulations eliminate basically all the problems identified in the standard formulation. However, as the approaches are different with respect to the evolution of activity and flow levels, the results from any models will in almost all cases be somewhat different under the ALT and LIN formulations. But the differences should be quite small, and solely related to the differing assumptions on the activity and flow levels.

3.3.2 Variable costs

The most significant additional modifications in the LIN formulation are related to the variable cost accounting. The objective components for all variable costs have been modified assuming linear evolution of activities and flows between milestone years.

The linearization of the flows and activities is done by a straightforward *linear interpolation* of the activity and flow variables between the milestone years. The interpolation is discretized over individual years.

3.3.3 Other equations

In addition to the variable cost components in the objective function, the following further modifications are done in the LIN formulation:

TIMES Version 2.8 User Note

- The cumulative constraints on commodity production (EQ(I)_CUMNET and EQ(I)_CUMPRD) are modified to include linear interpolation of the commodity variables involved;
- The cumulative constraints on commodity and flow taxes and subsidies (EQ(I)_CUMCST) are modified to include linear interpolation of the commodity and flow variables involved;
- The dynamic equations of the Climate module are modified to include linear interpolation of the variables involved;
- The inter-period storage equations are modified to include linear interpolation of the flow variables involved;
- The cumulative user constraints for activities and flows are modified to refer to the volume of the linearized period trapezoid instead of the period rectangle (see Figure 6).

Moreover, in the LIN formulation the activity of *inter-period storage* equations is measured at the milestone year (in the standard formulation it is measured at the end of each period). In addition, new EQ_STGIPS equations are added to ensure that the storage level remains non-negative at the end of each period. Consequently, two EQ_STGIPS equations are generated in each period with a period length longer than one year.

3.3.4 Summary of changed model generator files

The implementation of the LIN formulation includes changes in the following GAMS code files of the TIMES model generator:

- INITMTY.MOD
- PPMAIN.MOD
- COEFMAIN.MOD
- EQDECLR.MOD
- EQMAIN.MOD
- EQOBJ.MOD
- EQOBJVAR.MOD
- EQOBJELS.MOD
- EQCUMCOM.MOD
- EQBNDST.MOD
- COEF_EXT.CLI
- EQU_EXT.CLI
- EQDAMAGE.MOD

In addition, the implementation of the LIN formulation is also dependent on the following new GAMS code files:

- COEF_ALT.LIN
- EQSTGIPS.LIN
- EQSTGAUX.LIN

4 Benchmark Comparison of Formulations

4.1 Methodology

The alternative formulations can be transparently compared by using a very simple example model and running it under varying assumptions concerning period definitions with all of the different objective function formulations.

The results from the example model can be well compared to those obtained when running the same model with one-year periods, because when using one-year periods none of the problems identified with the standard objective function appear, and all costs are accounted quite accurately. Therefore, ideally the results from running the model with longer periods should be as close as possible to the results from running the model with one-year periods.

In the benchmark the example model is run with 4, 5 or 6 periods, and the cost results are subsequently compared to the results from running the same model horizon but assuming one-year periods (these are referred to as *reference results*).

4.2 Description of the example model

The example model used in the benchmark is extremely simple. The model has the following characteristics:

- ✓ Only one demand, which is linearly increasing between 2000 and 2030;
- ✓ Only one new technology producing the demand commodity, with investment and fixed costs and a reasonably typical TLIFE of 20 years;
- ✓ One existing technology producing the demand; however, instead of using a residual capacity the phase-out of the existing capacity is simulated by using NCAP_BND and TLIFE=1, and annualized costs equal to that of the new technology;
- ✓ Both technologies use a single fuel with a price increasing linearly between 2000 and 2030;
- ✓ The general discount rate is 7%.

The residual capacity was chosen to be simulated by using capacity bounds, because the costs of residual capacities are not accounted accurately in the TIMES objective function (because they are constant and do not affect the solution in any way). Therefore, using PRC_RESID would have caused some artificial cost differences in the objective function under the different formulations.

TIMES Version 2.8 User Note

4.3 Benchmark results

4.3.1 Example 1

The milestone years of Example 1 and the corresponding periods according to the standard formulation are shown below. The resulting values of the main objective components are shown in Table 1, and the marginal costs of the demand are shown in Figure 7, and compared to the reference run.

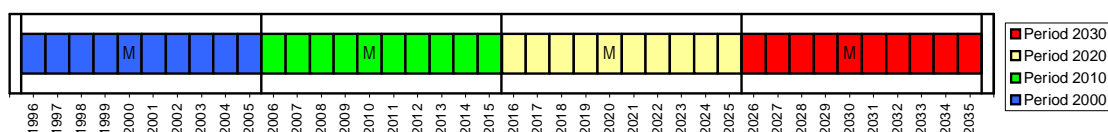


Table 1. Benchmark results for Example 1.

Benchmark Results	Reference	STD	MOD	ALT	LIN
Objective component values					
Investment costs (minus salvage)	417853	426659	426659	417853	417853
Fixed costs	378929	386915	386915	378929	378929
Variable costs	276664	273122	273122	278979	276664
Differences to reference					
Investment costs	0	2.1%	2.1%	0.0%	0.0%
Fixed costs	0	2.1%	2.1%	0.0%	0.0%
Variable costs	0	-1.3%	-1.3%	0.8%	0.0%

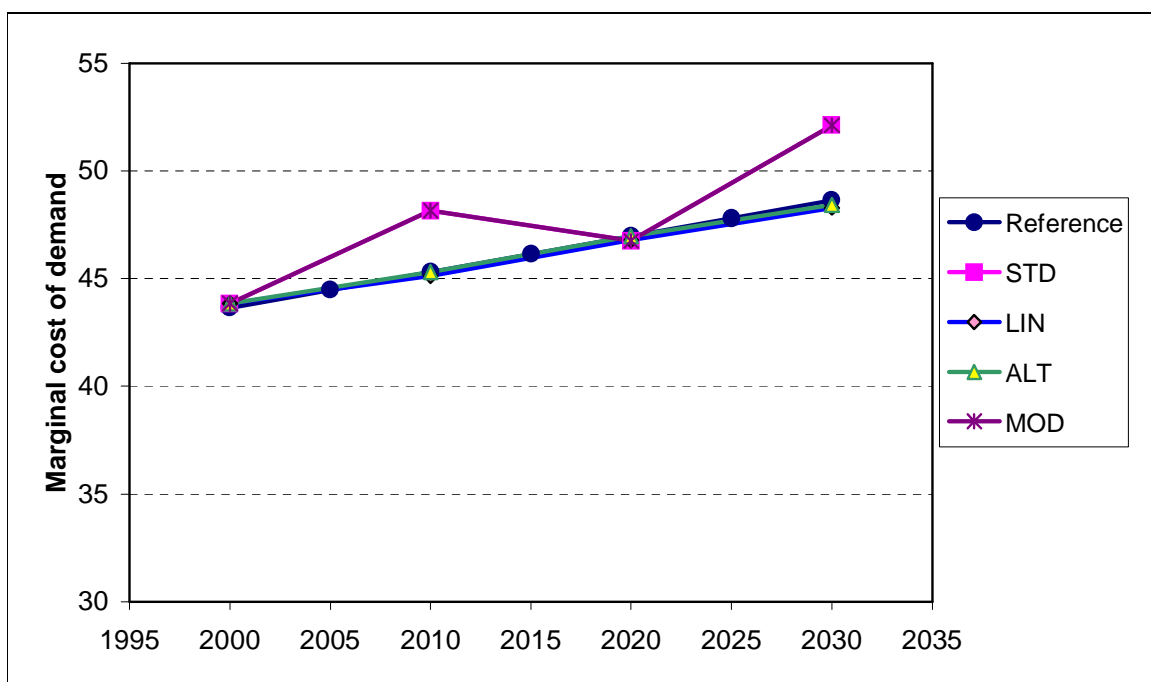


Figure 7. Comparison of the marginal cost of the demand in Example 1.

TIMES Version 2.8 User Note

4.3.2 Example 2

The milestone years of Example 1 and the corresponding periods according to the standard formulation are shown below. The resulting values of the main objective components are shown in Table 2, and the marginal costs of the demand are shown in Figure 8, and compared to the reference run.

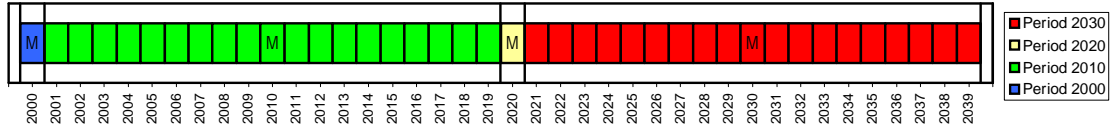


Table 2. Benchmark results for Example 2.

Benchmark Results	Reference	STD	MOD	ALT	LIN
Objective component values					
Investment costs (minus salvage)	345228	383005	355597	345228	345228
Fixed costs	313070	347327	322472	313070	313070
Variable costs	238674	249050	235133	240899	238674
Differences to reference					
Investment costs	0	10.9%	3.0%	0.0%	0.0%
Fixed costs	0	10.9%	3.0%	0.0%	0.0%
Variable costs	0	4.3%	-1.5%	0.9%	0.0%

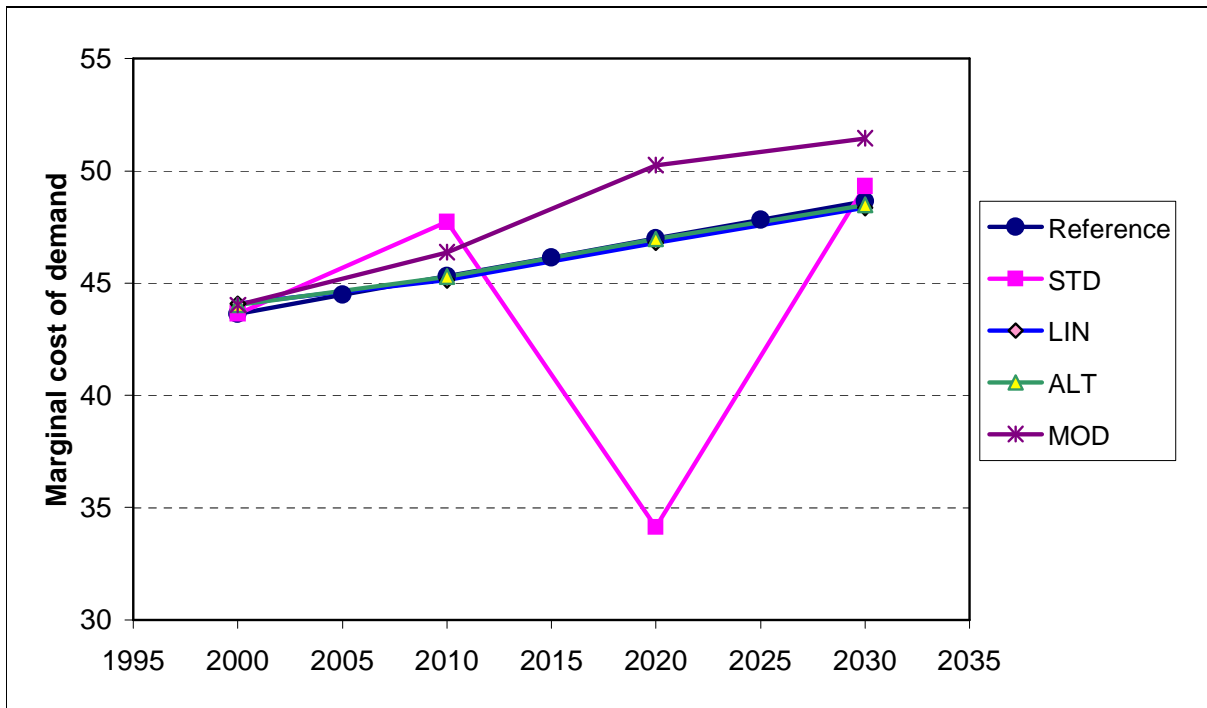


Figure 8. Comparison of the marginal cost of the demand in Example 2.

TIMES Version 2.8 User Note

4.3.3 Example 3

The milestone years of Example 1 and the corresponding periods according to the standard formulation are shown below. The resulting values of the main objective components are shown in Table 3, and the marginal costs of the demand are shown in Figure 9, and compared to the reference run.

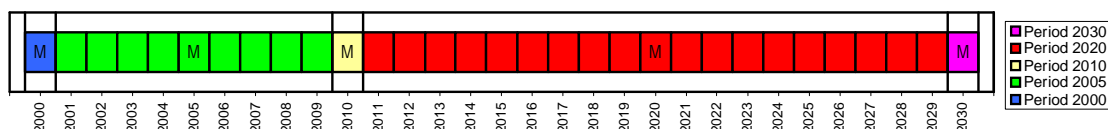


Table 3. Benchmark results for Example 3.

Benchmark Results	Reference	STD	MOD	ALT	LIN
Objective component values					
Investment costs (minus salvage)	315026	335742	321135	315026	315026
Fixed costs	285681	304467	291221	285681	285681
Variable costs	212997	217500	211147	214216	212997
Differences to reference					
Investment costs	0	6.6%	1.9%	0.0%	0.0%
Fixed costs	0	6.6%	1.9%	0.0%	0.0%
Variable costs	0	2.1%	-0.9%	0.6%	0.0%

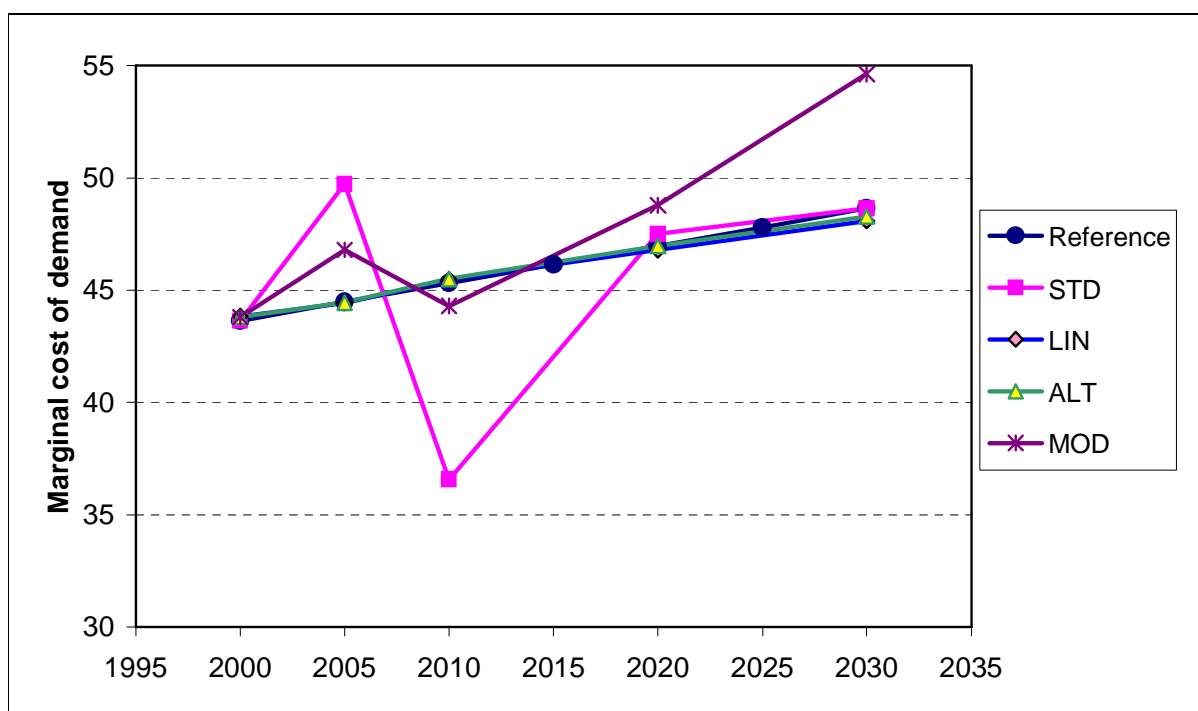


Figure 9. Comparison of the marginal cost of the demand in Example 3.

TIMES Version 2.8 User Note

4.3.4 Example 4

The milestone years of Example 1 and the corresponding periods according to the standard formulation are shown below. The resulting values of the main objective components are shown in Table 4, and the marginal costs of the demand are shown in Figure 10, and compared to the reference run.

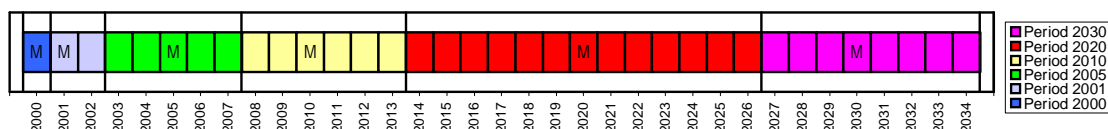


Table 4. Benchmark results for Example 4.

Benchmark Results	Reference	STD	MOD	ALT	LIN
Objective component values					
Investment costs (minus salvage)	330728	344604	335748	330728	330728
Fixed costs	299920	312504	304472	299920	299920
Variable costs	226346	227396	223971	227826	226346
Differences to reference					
Investment costs	0	4.2%	1.5%	0.0%	0.0%
Fixed costs	0	4.2%	1.5%	0.0%	0.0%
Variable costs	0	0.5%	-1.0%	0.7%	0.0%

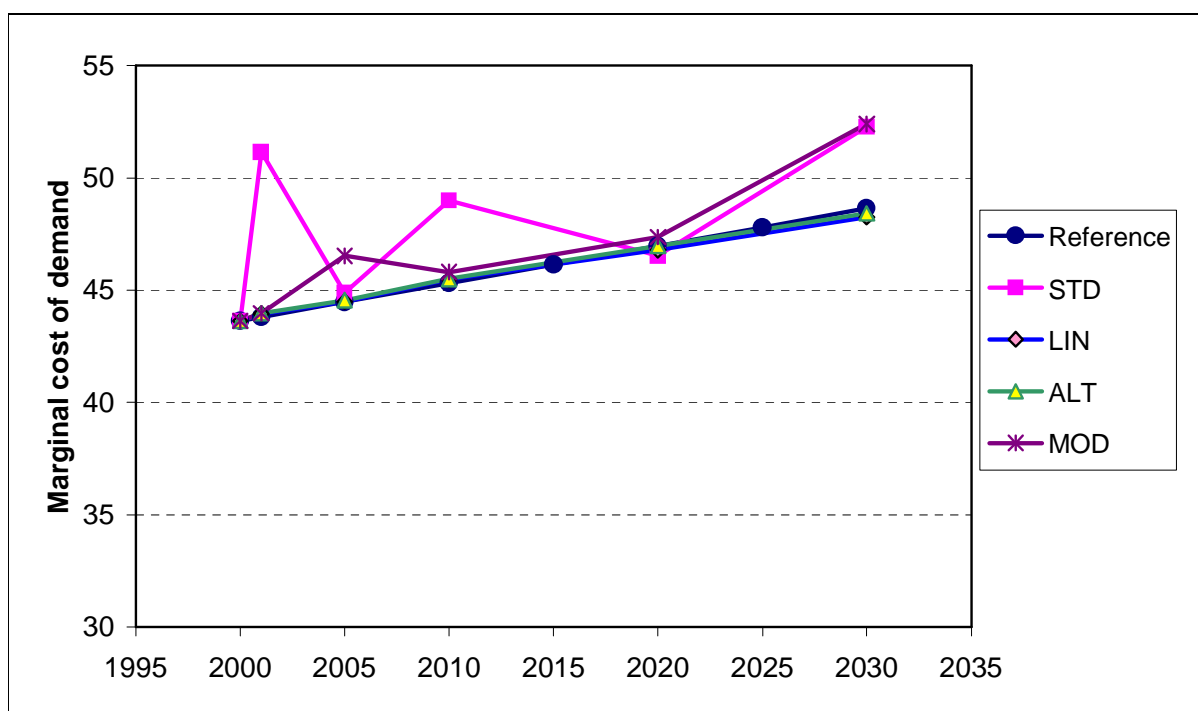


Figure 10. Comparison of the marginal cost of the demand in Example 4.

4.4 Conclusions from the benchmark exercise

According to the benchmark results, the standard formulation can lead to notable differences in the value of the objective components as compared to the reference runs (with one-year periods). The differences were largest in the four-period case with variable period lengths, where the capacity-related costs were 11% larger than in the standard formulation, and the variable costs were 4% larger. On the other hand, both the ALT and LIN formulations showed very small or zero cost differences in all cases.

However, one should note that as such the total cost differences do not necessarily mean distortions in the results. If all costs would be proportionally larger, the differences would have no impact on the 'physical' results of the model. Nevertheless, if the cost differences for the capacity-related costs and variable costs are of different magnitude, then they do indicate a clear impact on the relative competitiveness of technologies. According to the results, such a distorting impact may become notable in the standard formulation.

In both the ALT and LIN formulations, all the capacity-related costs are accounted fully accurately, so that they are equal to the costs in the reference results. In the LIN formulation also the variable costs can be very accurately accounted, even with zero differences to the reference results, provided that the changes in the cost data develop linearly between milestone years. Consequently, under the simplified assumption that all changes in the model data are linear over the model horizon, the LIN formulation one can produce identical model results regardless of the choice of milestone years. And even if the changes in the model data were not linear over time, the impact of the choice of the milestone years can be expected to be small.

The marginal cost results confirm the conclusions about the possible distortions in technology competitiveness that can arise when using the standard formulation. With variable period lengths, the marginal costs may fluctuate considerably around the correct values in the standard formulation, but behave much better in both the ALT and LIN formulations. By adding more periods, the variation in period lengths can be reduced, thereby reducing also the cost distortions. Hence, even if the problems in the standard formulation are generally quite small if reasonably smooth periods are used, the advantage of the ALT and LIN formulations is that *one can use less model periods* without too significant impacts on the model results.

Finally, with larger period lengths the differences in variable costs can also become notable between assumptions on constant flows and linearly evolving flows. In the Benchmark examples the overall differences in variable costs over the whole model horizon were at most 4.3%, which is explained by the assumed increasing unit costs, together with the standard assumptions on constant flows within periods. The LIN formulation can effectively eliminate also such distortions in variable costs.

5 Using the Alternative Formulations

5.1 Control switches

The alternative objective formulations can be activated by using any of the following control settings in the RUN file:

```
$SET OBJ AUTO (default)
$SET OBJ STD
$SET OBJ MOD
$SET OBJ ALT
$SET OBJ LIN
```

The default setting (AUTO) means that whenever the period definitions comply with the assumptions of the original standard formulation, the standard formulation will be used. However, if the period definitions do not comply with the assumptions of the standard formulation, then the MOD formulation is used. Currently the VEDA-FE user interface forces the period definitions to comply with the standard formulation, and therefore AUTO=STD when VEDA-FE is used. Due to being the default, using the AUTO option is actually never explicitly needed.

The period definitions comply with the assumptions of the standard formulation whenever the following condition holds:

$$t_i = \text{FLOOR}((B_i + E_i)/2), \text{ for all } i = 1, \dots, T \quad (1)$$

where t_i , $i = 1, \dots, T$, is the milestone year of period i , and B_i and E_i are the beginning and ending year of the period i , respectively.

Note that all the three alternative objective formulations have the advantage that the milestone years can be specified fully flexibly, without the restrictions imposed by the assumptions of the standard formulation (see section 2.1).

5.2 Impacts on results reporting

The meaning of all the levels of variables in the results is the same in the STD, MOD and ALT formulations. In these cases, the period-specific variable levels always represent the constant average level of the variable over the period in question. However, in the LIN formulation the interpretation is different. In the LIN formulation all the period-specific variable levels represent the level at the milestone year, and not a constant average value over the period.

TIMES Version 2.8 User Note

When reporting the annualized marginal cost results, slightly different undiscounting factors are needed in the new objective formulations ALT and LIN. As the undiscounting of the marginal costs is automatically done by the model generator, the user normally does not need to be concerned about this difference. However, should the user wish to undiscount some marginal cost results manually, the appropriate undiscounting factors should be used.

The correct undiscounting factors, $COEF_PVT(r,t)$, for the different objective function formulations are the following (using GAMS notation):

Variant	Correct formula for the basic undiscounting factors
STD	$COEF_PVT(r,t) = \text{SUM}(\text{PERIODYR}(t,y), \text{OBJ_DISC}(r,y))$
MOD	$COEF_PVT(r,t) = \text{SUM}(\text{PERIODYR}(t,y), \text{OBJ_DISC}(r,y))$
ALT	$COEF_PVT(r,t) = \text{SUM}(\text{TPULSEYR}(t,y), \text{TPULSE}(t,y) * \text{OBJ_DISC}(r,y))$
LIN	$COEF_PVT(r,t) = \text{SUM}(\text{TPULSEYR}(t,y), \text{TPULSE}(t,y) * \text{OBJ_DISC}(r,y))$

The components of the formulas above are the following:

- PERIODYR is a two-dimensional set containing all pairs (t_i, y) , where t_i is a milestone year ($i = 1, \dots, T$) and y is a year within the period containing the milestone year t_i , i.e. in the range $[B_i, E_i]$;
- OBJ_DISC(r,y) is the discounting factor for region r and year y ;
- TPULSEYR is a two-dimensional set containing all pairs (t_i, y) , where t_i is a milestone year ($i = 1, \dots, T$) and y is a year in the range (t_{i-1}, t_{i+1}) , or $[B_1, t_{i+1})$, if t_{i-1} does not exist, or $(t_{i-1}, E_T]$, if T_{i+1} does not exist;
- TPULSE is a two dimensional parameter defined over TPULSEYR having the following values (describing the height of the period trapezoid by year, see Figure 6):

$$\begin{aligned} TPULSE(t_1, y) &= 1 \text{ for } B_1 \leq y \leq t_1, \\ TPULSE(t_i, y) &= (y - t_{i-1}) / (t_i - t_{i-1}) \text{ for } t_{i-1} < y \leq t_i, \\ TPULSE(t_i, y) &= (t_{i+1} - y) / (t_{i+1} - t_i) \text{ for } t_i < y < t_{i+1}, \\ TPULSE(t_T, y) &= 1 \text{ for } t_T < y \leq E_T, \end{aligned}$$

As indicated above, the original undiscounting factors should only be used under the STD and ALT formulations. In both the ALT and LIN formulations the slightly more complex *trapezoid approach* has to be used for doing the undiscounting appropriately. This is so because in the ALT formulation the discounted rectangular period activities are adjusted to be consistent with the corresponding period trapezoids. In the LIN formulation this assumption is inherent (due to the linear evolution of flows).

Appendix 1: The Benchmark Example Model

```
$ONEMPTY  
$ONEPS  
$ONWARNING
```

```
SET ALL_REG  
/  
  IMPEXP  
  MNRNW  
  REG1  
/
```

```
SET REG  
/  
  REG1 'REG1'  
/
```

```
SET CUR  
/  
CUR  
/
```

```
SET DATAYEAR  
/  
  2000 '2000'  
  2001 '2001'  
  2005 '2005'  
  2010 '2010'  
  2020 '2020'  
  2030 '2030'  
/
```

```
SET TS_GROUP  
/  
REG1. ANNUAL. ANNUAL  
/
```

```
SET COM_GRP  
/  
OILCRD  
TPES  
/
```

```
SET COM  
/  
OILCRD  
TPES  
/
```

```
SET COM_TMAP  
/  
REG1. DEM. TPES  
REG1. NRG. OILCRD  
/
```

TIMES Version 2.8 User Note

```
SET PRC
/
IMPOI LCRD1
TECTPES1
TECTPES0
/

SET PRC_MAP
/
REG1. DMD. TECTPES0
REG1. DMD. TECTPES1
REG1. I RE. IMPOI LCRD1
/

SET PRC_TSL
/
REG1. IMPOI LCRD1. ANNUAL
REG1. TECTPES0. ANNUAL
REG1. TECTPES1. ANNUAL
/

SET PRC_ACTUNT
/
REG1. IMPOI LCRD1. OI LCRD. PJ
REG1. TECTPES0. TPES. PJ
REG1. TECTPES1. TPES. PJ
/

SET TOP
/
REG1. TECTPES0. OI LCRD. I N
REG1. TECTPES0. TPES. OUT
REG1. TECTPES1. OI LCRD. I N
REG1. TECTPES1. TPES. OUT
/

SET TOP_I RE
/
IMPEXP. OI LCRD. REG1. OI LCRD. IMPOI LCRD1
/

SET PRC_DESC
/
REG1. IMPOI LCRD1 'Import of Crude Oil'
REG1. TECTPES0 'Base year Demand technology'
REG1. TECTPES1 'New Demand technology'
/

SET COM_DESC
/
REG1. OI LCRD 'Crude Oil'
REG1. TPES 'Total primary energy supply'
/

PARAMETER G_DRATE /
REG1. 2000. CUR 0. 07
/;
```

TIMES Version 2.8 User Note

PARAMETER NCAP_BND

/

REG1. 0. TECTPES0. UP 2
REG1. 1996. TECTPES0. UP 1000
REG1. 2000. TECTPES0. UP 1000
REG1. 2010. TECTPES0. UP 0
REG1. 2000. TECTPES1. UP 0

/

PARAMETER

ACT_EFF ' ' /

REG1. 2000. TECTPES0. ACT. ANNUAL 1
REG1. 2000. TECTPES1. ACT. ANNUAL 1

/

PARAMETER

COM_PROJ ' ' /

REG1. 2000. TPES 1000
REG1. 2030. TPES 2000

/

PARAMETER

IRE_PRICE ' ' /

REG1. 2000. IMPOI LCRD1. 0I LCRD. ANNUAL. IMPEXP. IMP. CUR 10
REG1. 2030. IMPOI LCRD1. 0I LCRD. ANNUAL. IMPEXP. IMP. CUR 15

/

PARAMETER

NCAP_COST ' ' /

* For TECTPES0, cost is equal to annual payment of TECTPES1

REG1. 2000. TECTPES0. CUR 17.6435
REG1. 2000. TECTPES1. CUR 200

/

PARAMETER

NCAP_FOM ' ' /

REG1. 2000. TECTPES0. CUR 16
REG1. 2000. TECTPES1. CUR 16

/

PARAMETER

NCAP_TLIFE ' ' /

REG1. 2000. TECTPES0 1
REG1. 2000. TECTPES1 20

/

TIMES Version 2.8 User Note

Appendix 2a: The RUN files for Example 1

```
$TITLE TIMES -- VERSION 2.8.0
OPTION LIMROW=1000, LIMCOL=1000, SOLPRINT=ON, ITERLIM=999999;
OPTION RESLIM=50000, PROFILE=1, SOLVEOPT=REPLACE;

$OFFLISTING

*-----*
* BATINCLUDE calls should all be with lower case file names!!! *
*-----*

* initialize the environment variables
$ SET REDUCE 'YES'
$ SET VDA YES
$ SET CLI YES
$ SET DEBUG 'NO'
$ SET DUMPSOL 'NO'
$ SET SOLVE_NOW 'YES'
$ SET MODEL_NAME 'TIMES'
$ SET SOLVEDA 'YES'
$ SET OBJ STD

* merge declarations & data
$ ONMULTI

* the times-slices MUST come 1st to ensure ordering OK
SET ALL_TS
/
ANNUAL
/

* perform fixed declarations
$BATINCLUDE initsys.mod

* declare the (system/user) empties
$ BATINCLUDE initsys.mod DBG

$BATINCLUDE benchmark.dd

* One-year periods!
SET MILESTONYR / 1996*2035 /;

$SET RUN_NAME %SYSTEM.FN%

B(T) = YEARVAL(T);
E(T) = YEARVAL(T);

g_dyear = 2000;

$ SET VEDA VDD 'YES'

* do the rest
$ BATINCLUDE maindrv.mod mod
```

TIMES Version 2.8 User Note

```
$TITLE TIMES -- VERSION 2.8.0
OPTION LIMROW=1000, LIMCOL=1000, SOLPRINT=ON, ITERLIM=999999;
OPTION RESLIM=50000, PROFILE=1, SOLVEOPT=REPLACE;
```

```
$OFFLISTING
```

```
*-----*
* BATINCLUDE calls should all be with lower case file names!!! *
*-----*
```

```
* initialize the environment variables
$ SET REDUCE 'YES'
$ SET VDA YES
$ SET CLI YES
$ SET DEBUG 'NO'
$ SET DUMPSOL 'NO'
$ SET SOLVE_NOW 'YES'
$ SET MODEL_NAME 'TIMES'
$ SET SOLVEDA 'YES'
$ SET OBJ STD
```

```
* merge declarations & data
$ ONMULTI
```

```
* the times-slices MUST come 1st to ensure ordering OK
```

```
SET ALL_TS
```

```
/
```

```
ANNUAL
```

```
/
```

```
* perform fixed declarations
$BATINCLUDE initsys.mod
```

```
* declare the (system/user) empties
$ BATINCLUDE initsys.mod DBG
```

```
$BATINCLUDE benchmark.dd
```

```
* Four periods: 1996-2035
SET MILESTONYR /2000, 2010, 2020, 2030/;
```

```
PARAMETER B /
2000 1996, 2010 2006, 2020 2016, 2030 2026
```

```
/;
```

```
PARAMETER E /
2000 2005, 2010 2015, 2020 2025, 2030 2035
```

```
/;
```

```
$SET RUN_NAME %SYSTEM.FN%
```

```
g_dyear = 2000;
```

```
$ SET VEDA VDD 'YES'
```

```
* do the rest
$ BATINCLUDE maindrv.mod mod
```

TIMES Version 2.8 User Note

Appendix 2b: The RUN files for Example 2

```
$TITLE TIMES -- VERSION 2.8.0
OPTION LIMROW=1000, LIMCOL=1000, SOLPRINT=ON, ITERLIM=999999;
OPTION RESLIM=50000, PROFILE=1, SOLVEOPT=REPLACE;

$OFFLISTING

*-----*
* BATINCLUDE calls should all be with lower case file names!!! *
*-----*

* initialize the environment variables
$ SET REDUCE 'YES'
$ SET VDA YES
$ SET CLI YES
$ SET DEBUG 'NO'
$ SET DUMPSOL 'NO'
$ SET SOLVE_NOW 'YES'
$ SET MODEL_NAME 'TIMES'
$ SET SOLVEDA 'YES'
$ SET OBJ STD

* merge declarations & data
$ ONMULTI

* the times-slices MUST come 1st to ensure ordering OK
SET ALL_TS
/
ANNUAL
/

* perform fixed declarations
$BATINCLUDE initsys.mod

* declare the (system/user) empties
$ BATINCLUDE initsys.mod DBG

$BATINCLUDE benchmark.dd

* One-year periods!
SET MILESTONYR / 2000*2039 /;

$SET RUN_NAME %SYSTEM.FN%

B(T) = YEARVAL(T);
E(T) = YEARVAL(T);

g_dyear = 2000;

$ SET VEDA VDD 'YES'

* do the rest
$ BATINCLUDE maindrv.mod mod
```

TIMES Version 2.8 User Note

```
$TITLE TIMES -- VERSION 2.8.0
OPTION LIMROW=1000, LIMCOL=1000, SOLPRINT=ON, ITERLIM=999999;
OPTION RESLIM=50000, PROFILE=1, SOLVEOPT=REPLACE;
```

```
$OFFLISTING
```

```
*-----*
* BATINCLUDE calls should all be with lower case file names!!! *
*-----*
```

```
* initialize the environment variables
```

```
$ SET REDUCE 'YES'
$ SET VDA YES
$ SET CLI YES
$ SET DEBUG 'NO'
$ SET DUMPSOL 'NO'
$ SET SOLVE_NOW 'YES'
$ SET MODEL_NAME 'TIMES'
$ SET SOLVEDA 'YES'
$ SET OBJ STD
```

```
* merge declarations & data
```

```
$ ONMULTI
```

```
* the times-slices MUST come 1st to ensure ordering OK
```

```
SET ALL_TS
```

```
/
```

```
 ANNUAL
```

```
/
```

```
* perform fixed declarations
```

```
$BATINCLUDE initsys.mod
```

```
* declare the (system/user) empties
```

```
$ BATINCLUDE initsmt.mod DBG
```

```
$BATINCLUDE benchmark.dd
```

```
* Four periods: 2000-2039
```

```
SET MILESTONYR /2000, 2010, 2020, 2030/;
```

```
PARAMETER B /
```

```
2000 2000, 2010 2001, 2020 2020, 2030 2021
```

```
/;
```

```
PARAMETER E /
```

```
2000 2000, 2010 2019, 2020 2020, 2030 2039
```

```
/;
```

```
$SET RUN_NAME %SYSTEM.FN%
```

```
g_dyear = 2000;
```

```
$ SET VEDAVIDD 'YES'
```

```
* do the rest
```

```
$ BATINCLUDE maindrv.mod mod
```

TIMES Version 2.8 User Note

Appendix 2c: The RUN files for Example 3

```
$TITLE TIMES -- VERSION 2.8.0
OPTION LIMROW=1000, LIMCOL=1000, SOLPRINT=ON, ITERLIM=999999;
OPTION RESLIM=50000, PROFILE=1, SOLVEOPT=REPLACE;

$OFFLISTING

*-----*
* BATINCLUDE calls should all be with lower case file names!!! *
*-----*

* initialize the environment variables
$ SET REDUCE 'YES'
$ SET VDA YES
$ SET CLI YES
$ SET DEBUG 'NO'
$ SET DUMPSOL 'NO'
$ SET SOLVE_NOW 'YES'
$ SET MODEL_NAME 'TIMES'
$ SET SOLVEDA 'YES'
$ SET OBJ STD

* merge declarations & data
$ ONMULTI

* the times-slices MUST come 1st to ensure ordering OK
SET ALL_TS
/
ANNUAL
/

* perform fixed declarations
$BATINCLUDE initsys.mod

* declare the (system/user) empties
$ BATINCLUDE initsys.mod DBG

$BATINCLUDE benchmark.dd

* One-year periods!
SET MILESTONYR / 2000*2030 /;

$SET RUN_NAME %SYSTEM.FN%

B(T) = YEARVAL(T);
E(T) = YEARVAL(T);

g_dyear = 2000;

$ SET VEDA VDD 'YES'

* do the rest
$ BATINCLUDE maindrv.mod mod
```

TIMES Version 2.8 User Note

```
$TITLE TIMES -- VERSION 2.8.0
OPTION LIMROW=1000, LIMCOL=1000, SOLPRINT=ON, ITERLIM=999999;
OPTION RESLIM=50000, PROFILE=1, SOLVEOPT=REPLACE;
```

```
$OFFLISTING
```

```
*-----*
* BATINCLUDE calls should all be with lower case file names!!! *
*-----*
```

```
* initialize the environment variables
```

```
$ SET REDUCE 'YES'
$ SET VDA YES
$ SET CLI YES
$ SET DEBUG 'NO'
$ SET DUMPSOL 'NO'
$ SET SOLVE_NOW 'YES'
$ SET MODEL_NAME 'TIMES'
$ SET SOLVEDA 'YES'
$ SET OBJ STD
```

```
* merge declarations & data
```

```
$ ONMULTI
```

```
* the times-slices MUST come 1st to ensure ordering OK
```

```
SET ALL_TS
```

```
/
```

```
 ANNUAL
```

```
/
```

```
* perform fixed declarations
```

```
$BATINCLUDE initsys.mod
```

```
* declare the (system/user) empties
```

```
$ BATINCLUDE initsmt.mod DBG
```

```
$BATINCLUDE benchmark.dd
```

```
* Five periods: 2000-2030
```

```
SET MILESTONYR /2000, 2005, 2010, 2020, 2030/;
```

```
PARAMETER B /
```

```
2000 2000, 2005 2001, 2010 2010, 2020 2011, 2030 2030
```

```
/;
```

```
PARAMETER E /
```

```
2000 2000, 2005 2009, 2010 2010, 2020 2029, 2030 2030
```

```
/;
```

```
$SET RUN_NAME %SYSTEM.FN%
```

```
g_dyear = 2000;
```

```
$ SET VEDA VDD 'YES'
```

```
* do the rest
```

```
$ BATINCLUDE maindrv.mod mod
```

TIMES Version 2.8 User Note

Appendix 2d: The RUN files for Example 4

```
$TITLE TIMES -- VERSION 2.8.0
OPTION LIMROW=1000, LIMCOL=1000, SOLPRINT=ON, ITERLIM=999999;
OPTION RESLIM=50000, PROFILE=1, SOLVEOPT=REPLACE;

$OFFLISTING

*-----*
* BATINCLUDE calls should all be with lower case file names!!! *
*-----*

* initialize the environment variables
$ SET REDUCE 'YES'
$ SET VDA YES
$ SET CLI YES
$ SET DEBUG 'NO'
$ SET DUMPSOL 'NO'
$ SET SOLVE_NOW 'YES'
$ SET MODEL_NAME 'TIMES'
$ SET SOLVEDA 'YES'
$ SET OBJ STD

* merge declarations & data
$ ONMULTI

* the times-slices MUST come 1st to ensure ordering OK
SET ALL_TS
/
ANNUAL
/

* perform fixed declarations
$BATINCLUDE initsys.mod

* declare the (system/user) empties
$ BATINCLUDE initsmt.mod DBG

$BATINCLUDE benchmark.dd

* One-year periods!
SET MILESTONYR / 2000*2034 /;

$SET RUN_NAME %SYSTEM.FN%

B(T) = YEARVAL(T);
E(T) = YEARVAL(T);

g_dyear = 2000;

$ SET VEDAVIDD 'YES'

* do the rest
$ BATINCLUDE maindrv.mod mod
```

TIMES Version 2.8 User Note

```
$TITLE TIMES -- VERSION 2.8.0
OPTION LIMROW=1000, LIMCOL=1000, SOLPRINT=ON, ITERLIM=999999;
OPTION RESLIM=50000, PROFILE=1, SOLVEOPT=REPLACE;

$OFFLISTING

* -----*
* BATINCLUDE calls should all be with lower case file names!!! *
* -----*

* initialize the environment variables
$ SET REDUCE 'YES'
$ SET VDA YES
$ SET CLI YES
$ SET DEBUG 'NO'
$ SET DUMPSOL 'NO'
$ SET SOLVE_NOW 'YES'
$ SET MODEL_NAME 'TIMES'
$ SET SOLVEDA 'YES'
$ SET OBJ STD

* merge declarations & data
$ ONMULTI

* the times-slices MUST come 1st to ensure ordering OK
SET ALL_TS
/
ANNUAL
/

* perform fixed declarations
$BATINCLUDE initsys.mod

* declare the (system/user) empties
$ BATINCLUDE initsys.mod DBG

$BATINCLUDE benchmark.dd

* Six periods: 2000-2034
SET MILESTONYR /2000, 2001, 2005, 2010, 2020, 2030/;

PARAMETER B /
2000 2000, 2001 2001, 2005 2003, 2010 2008, 2020 2014, 2030 2027
/;
PARAMETER E /
2000 2000, 2001 2002, 2005 2007, 2010 2013, 2020 2026, 2030 2034
/;

$SET RUN_NAME %SYSTEM.FN%

g_dyear = 2000;

$ SET VEDA VDD 'YES'

* do the rest
$ BATINCLUDE maindrv.mod mod
```