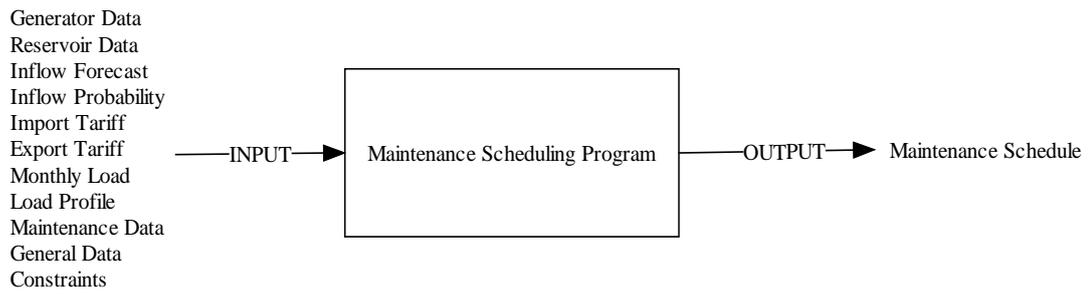
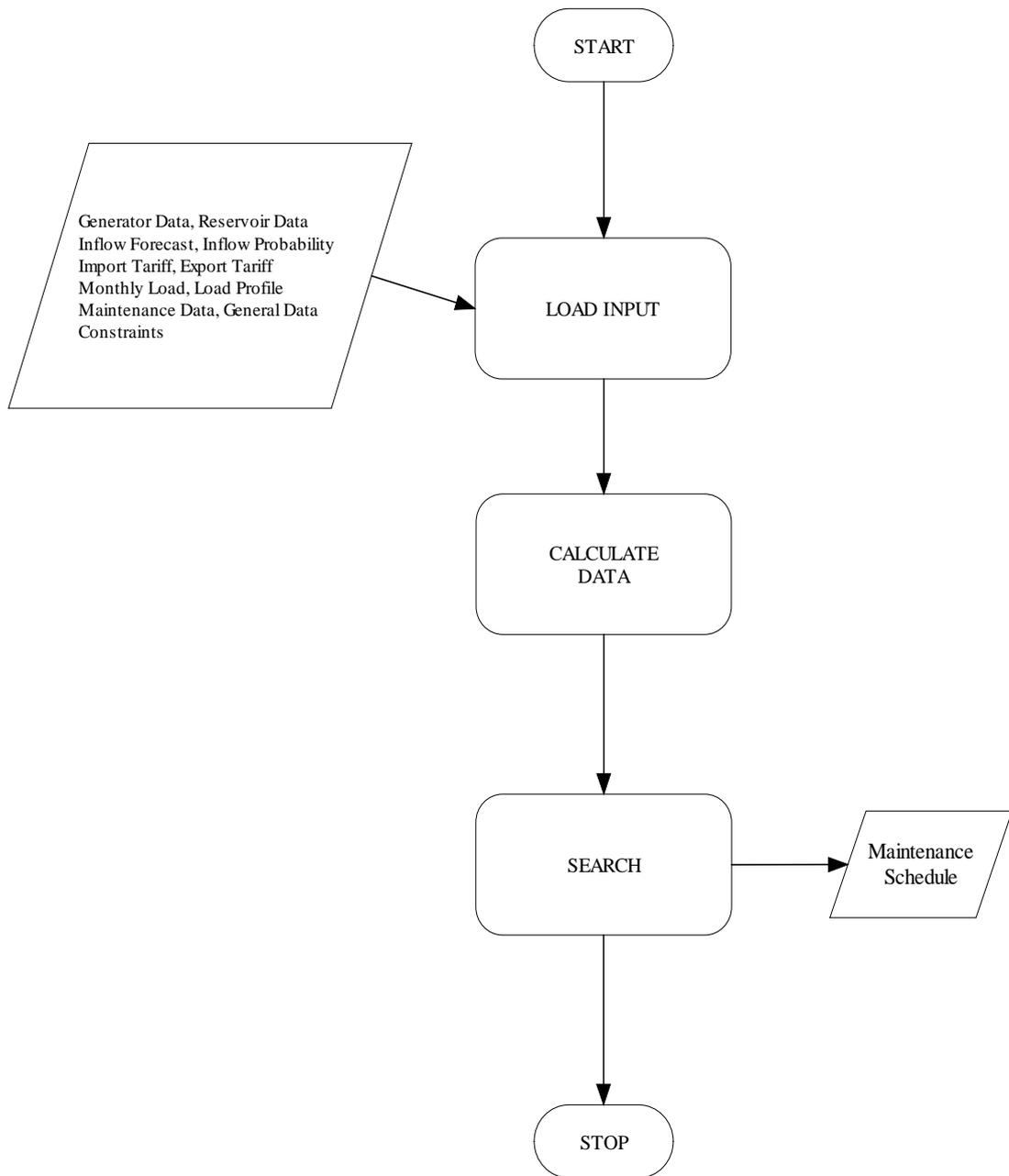


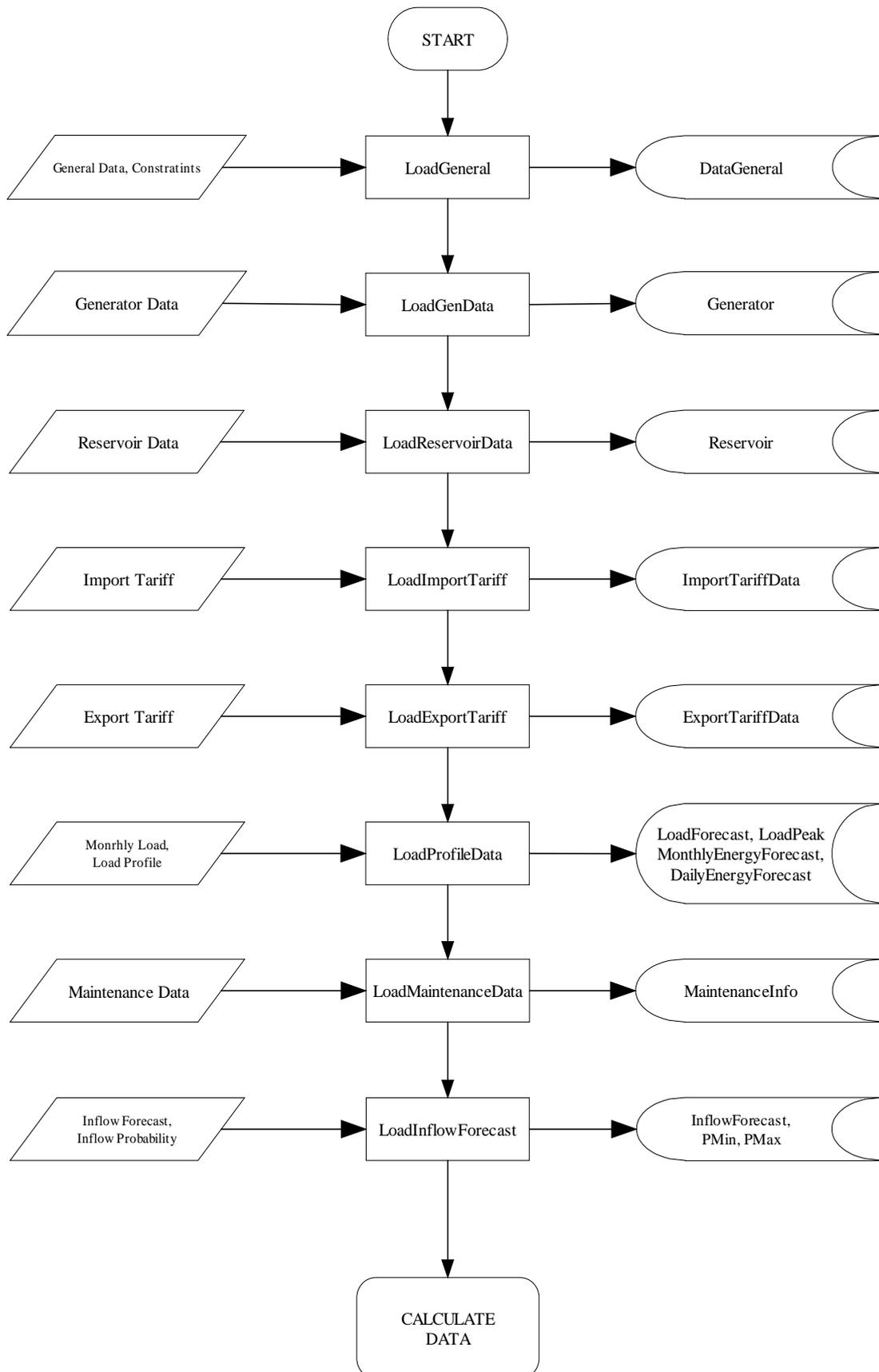
# Appendix A

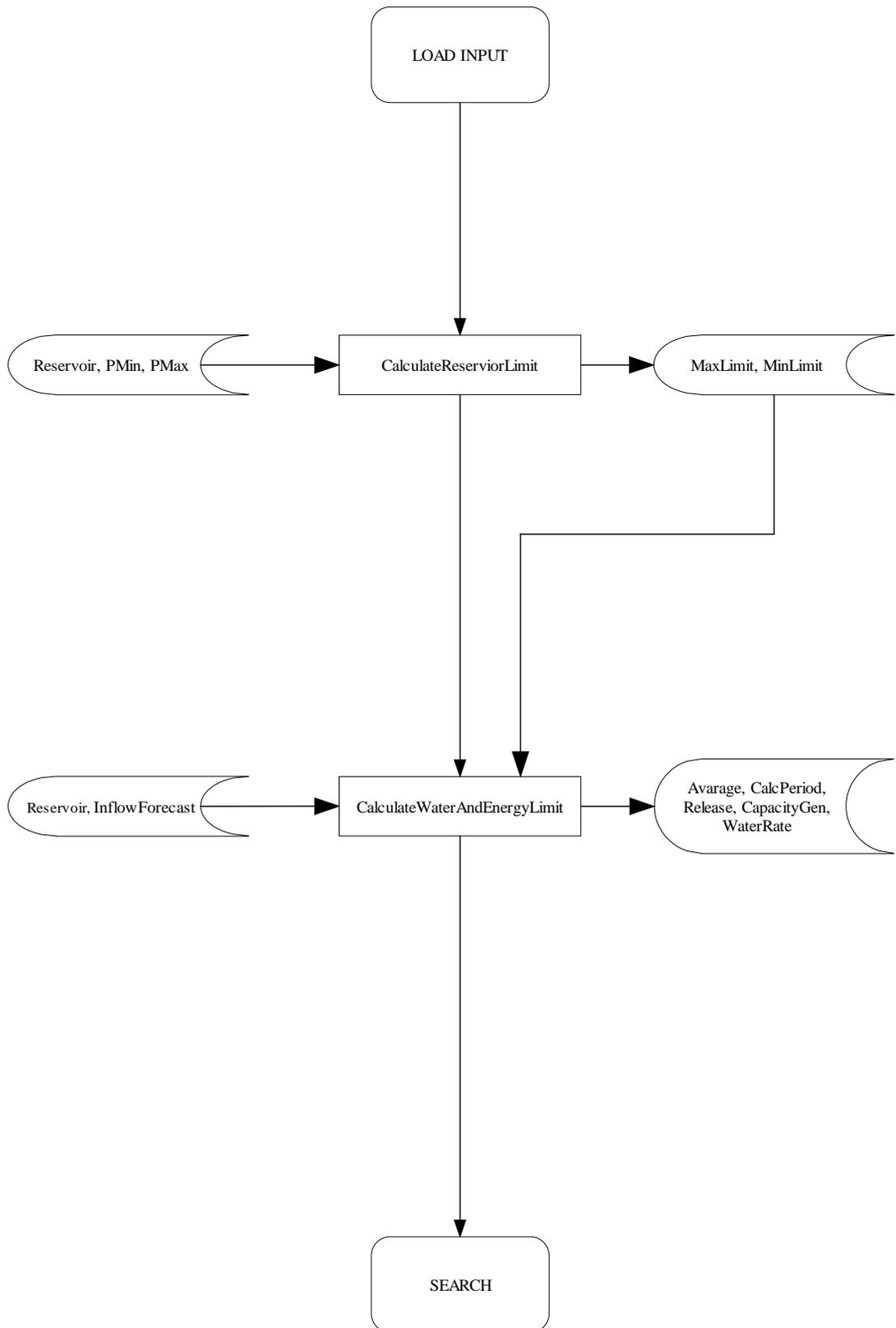
## Software Algorithm

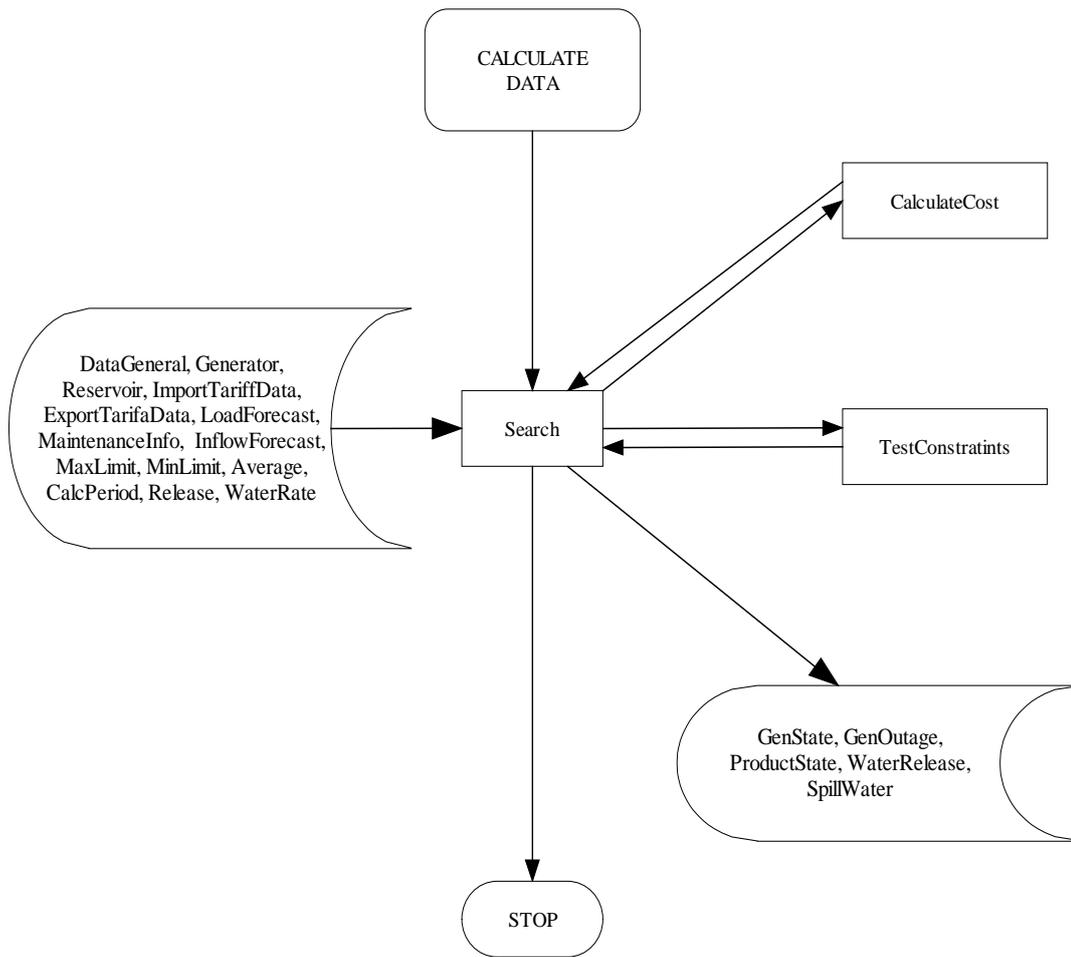
### A.1 Program Flowchart











## DataGeneral

.StartDate  
 .EndDate  
 .MaximumManPower  
 .MaximumImportMW  
 .MaximumExportMW  
Use: DataGeneral

## Generator

.UnitName  
 .Capacity  
 .New\_Status  
 .Retire\_Status  
 .New\_Retire\_Date  
 .k1  
 .q  
Use: Generator(generator#)

## Reservoir

.ReservoirName  
 .InitialVolume  
 .EndedVolume  
 .MaxTurbineFlow  
 .PeakOperation  
 .MaxCapacity  
 .InflowSeason  
 .TotalEvap  
 .NumberOfGenerator  
 .Va  
 .Vb  
 .Vc  
 .Vd  
 .Ve  
 .Vf  
Use: Reservoir(reservoir#)

## MaintenanceInfo

.MaintenanceCount  
 .MinimumAvail  
 .Maintenance(maintenance#)  
 .Reservoir  
 .UnitName  
 .PossibleStart  
 .PossibleFinish  
 .DaysNeeded  
 .ManpowerNeeded  
Use: MaintenanceInfo(generator#)

## InflowForecast

.ReservoirName  
 .Month(month#)  
Use: InflowForecast(reservoir#)

## ImportTariffData

.WeekDay(month#, hour#)  
 .Hol\_Sat\_Sun(month#, hour#)  
Use: ImportTariffData

## ExportTariffData

.WeekDay(month#, hour#)  
 .Hol\_Sat\_Sun(month#, hour#)  
Use: ExportTariffData

## CalcPeriod

.StartMonth  
 .StopMonth  
 .StartHour  
 .StopHour  
Use: CalcPeriod(reservoir#, month#)

## GenOutage

.StartDate  
 .StopDate  
USE: GenOutage(generator#, maintenance#, day#)

## LoadForecast(day#, hour#)

LoadPeak(day#)

MonthlyEnergyForecast(month#)

DailyEnergyForecast(day#)

PMin(reservoir#, month#)

PMax(reservoir#, month#)

MaxLimit(reservoir#, month#)

MinLimit(reservoir#, month#)

TotalCalcPeriod(reservoir#)

Average(reservoir#, month#)

Release(reservoir#, month#)

CapacityGen(generator#, month#)

WaterRate(generator#, month#)

GenState(generator#, day#)

ProductState(generator#, hour#)

WaterRelease(reservoir#, month#)

SpillWater(month#)

## A.2 CalculateWaterAndEnergyLimit

Input:

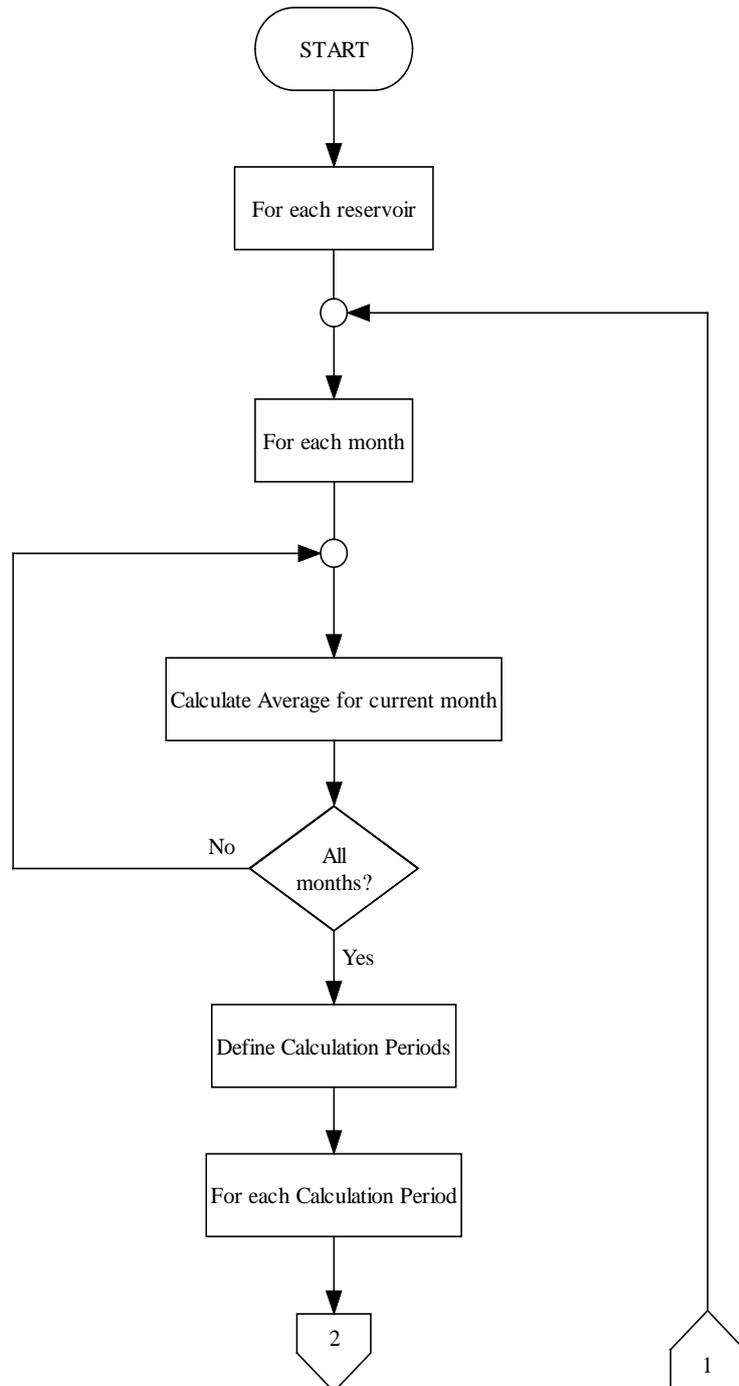
- Reservoir(reservoir#)
- InflowForecast(reservoir#)

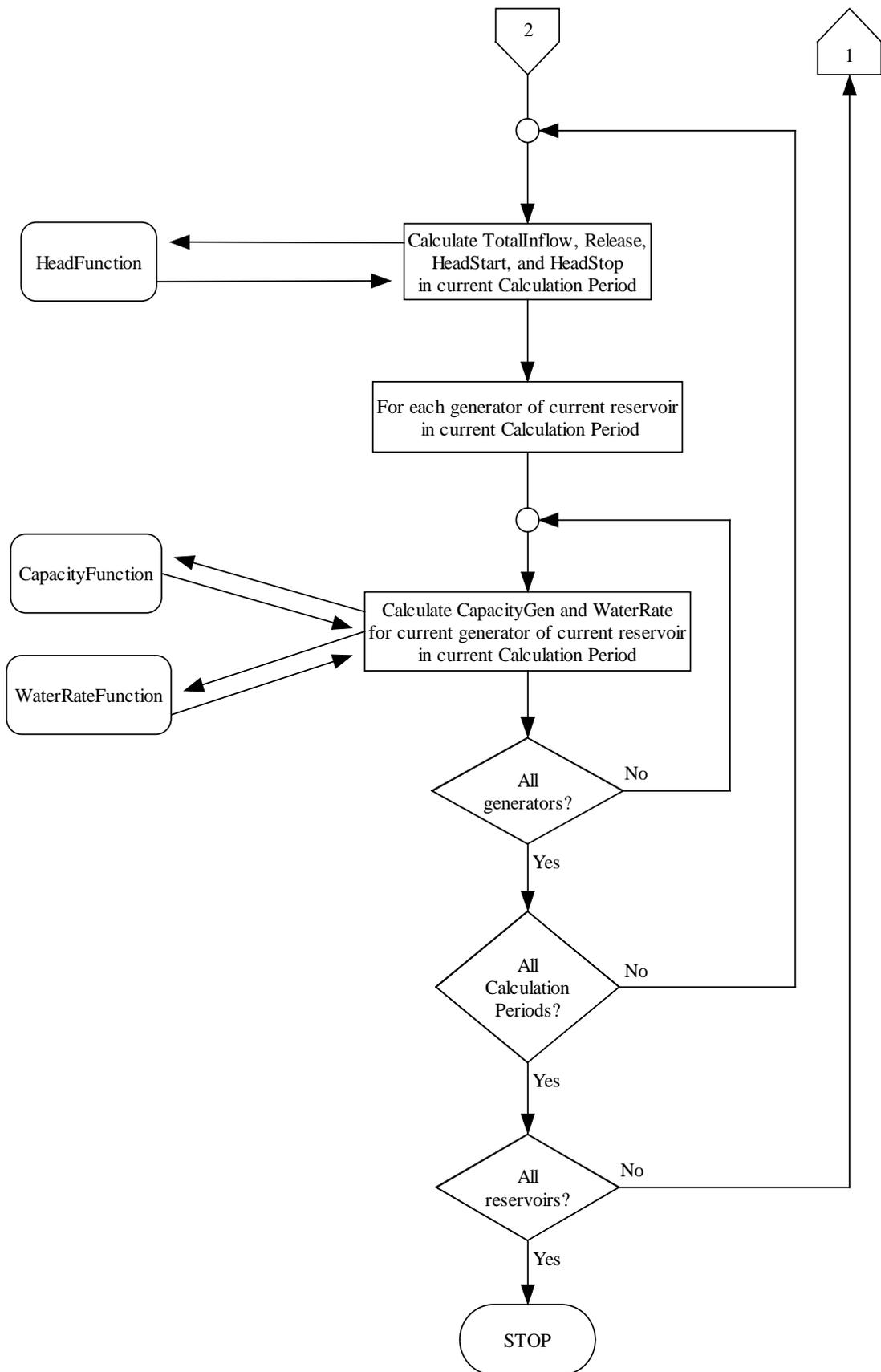
Output:

- Average(reservoir#, month#)
- CalcPeroid(reservoir#, calcP#)
- TotalCalcPeriod(reservoir#)
- Release(reservoir#, calcP#)
- CapacityGen(generator#, calcP#)
- WaterRate(generator#, calcP#)

LocalVariable:

- TotalInflow(calcP#)
- HeadStart(calcP#)
- HeadStop(calcP#)





## Calculate Average for current month

```
Average(current reservoir, current month) =
    (MaxLimit(current reservoir, current month) + MinLimit(current reservoir, current month)) / 2
```

## Define Calculation Periods

```
calcP = 1
StartMonth = 1 (or first month of maintenance schedule)
Do
  CalcPeriod(current reservoir, calcP).StartMonth = StartMonth
  StopMonth = StartMonth
  If StopMonth = total number of months Then
    CalcPeriod(current reservoir, calcP).StopMonth = StopMonth
    Exit Do
  Else
    Do until Average(current reservoir, StopMonth) <= Average(current reservoir, StopMonth + 1)
      StopMonth = StopMonth + 1
      If StopMonth >= total number of months Then
        CalcPeriod(current reservoir, calcP).StopMonth = StopMonth
        Exit Do
      End If
    Loop
  End If
  If StopMonth = StartMonth Then
    If Average(current reservoir, StopMonth) < Average(current reservoir, StopMonth + 1)
      Do
        StopMonth = StopMonth + 1
        If StopMonth >= total number of months Then
          CalcPeriod(current reservoir, calcP).StopMonth = StopMonth
          Exit Do
        End If
      Loop until Average(current reservoir, StopMonth) >= Average(current reservoir, StopMonth + 1)
    End If
  End If
  CalcPeriod(current reservoir, calcP).StopMonth = StopMonth
  If StopMonth >= total number of months Then
    Exit Do
  End If
  calcP = calcP + 1
  StartMonth = StopMonth + 1
Loop until StopMonth >= total number of months

TotalCalcPeriod(current reservoir) = calcP (save total number of calculation periods of current reservoir)
```

## Calculate Release, HeadStart, and HeadStop in current Calculation Period

TotalInflow(current calcP) = Sum of InflowForecast(current reservoir).Month(month#) from  
 month# = StartMonth of current Calculation Period to StopMonth of current Calculation Period

Select Case current calcP

Case current calcP is first period

$$\begin{aligned} \text{Release}(\text{current reservoir, current calcP}) = & \text{Reservoir}(\text{current reservoir}).\text{InitialVolume} \\ & + \text{TotalInflow}(\text{current calcP}) \\ & - \text{Average}(\text{current reservoir, StopMonth of Current calculation Period}) \\ & - (\text{Reservoir}(\text{current reservoir}).\text{TotalEvap} * \text{number of months in current Calculation Period}) \end{aligned}$$

$$\text{HeadStart}(\text{current calcP}) = \text{HeadFunction}(\text{current reservoir, Reservoir}(\text{current reservoir}).\text{InitialVolume})$$

$$\text{HeadStop}(\text{current calcP}) = \text{HeadFunction}(\text{current reservoir, Average}(\text{current reservoir, StopMonth of current Calculation Period}))$$

Case current calcP is between first period and last period

$$\begin{aligned} \text{Release}(\text{current reservoir, current calcP}) = & \text{Average}(\text{current reservoir, StopMonth of previous Calculation Period}) \\ & + \text{TotalInflow}(\text{current calcP}) \\ & - \text{Average}(\text{current reservoir, StopMonth of Current calculation Period}) \\ & - (\text{Reservoir}(\text{current reservoir}).\text{TotalEvap} * \text{number of months in current Calculation Period}) \end{aligned}$$

$$\text{HeadStart}(\text{current calcP}) = \text{HeadFunction}(\text{current reservoir, Average}(\text{current reservoir, StopMonth of previous Calculation Period}))$$

$$\text{HeadStop}(\text{current calcP}) = \text{HeadFunction}(\text{current reservoir, Average}(\text{current reservoir, StopMonth of current Calculation Period}))$$

Case current calcP is last period

$$\begin{aligned} \text{Release}(\text{current reservoir, current calcP}) = & \text{Average}(\text{current reservoir, StopMonth of previous Calculation Period}) \\ & + \text{TotalInflow}(\text{current calcP}) \\ & - \text{Reservoir}(\text{current reservoir}).\text{EndedVolume} \\ & - (\text{Reservoir}(\text{current reservoir}).\text{TotalEvap} * \text{number of months in current Calculation Period}) \end{aligned}$$

$$\text{HeadStart}(\text{current calcP}) = \text{HeadFunction}(\text{current reservoir, Average}(\text{current reservoir, StopMonth of current Calculation Period}))$$

$$\text{HeadStop}(\text{current calcP}) = \text{HeadFunction}(\text{current reservoir, Average}(\text{current reservoir, Reservoir}(\text{current reservoir}).\text{EndedVolume}))$$

End Select

## Calculate CapacityGen and WaterRate for current generator of current reservoir in current Calculation Period

$$\begin{aligned} \text{CapacityGen}(\text{current generator, current calcP}) = & \text{CapacityFunction}(\text{current generator,} \\ & (\text{HeadStart}(\text{current reservoir, current calcP}) \\ & + \text{HeadStop}(\text{current reservoir, current calcP})) / 2) \end{aligned}$$

$$\begin{aligned} \text{WaterRate}(\text{current generator, current calcP}) = & \text{WaterRateFunction}(\text{current generator,} \\ & (\text{HeadStart}(\text{current reservoir, current calcP}) \\ & + \text{HeadStop}(\text{current reservoir, current calcP})) / 2) \end{aligned}$$

### A.3 HeadFunction

Input:

- Reservoir Number  
- Volume

Output:

- Head value

Calculate Head value with formula  $y = V_a x^5 + V_b x^4 + V_c x^3 + V_d x^2 + V_e x + V_f$

```
Return Head = Reservoir(reservoir#).Va * Volume * Volume * Volume * Volume * Volume)
+ (Reservoir(reservoir#).Vb * Volume * Volume * Volume * Volume)
+ (Reservoir(reservoir#).Vc * Volume * Volume * Volume)
+ (Reservoir(reservoir#).Vd * Volume * Volume)
+ (Reservoir(reservoir#).Ve * Volume)
+ Reservoir(reservoir#).Vf
```

### CapacityFunction

Input:

- Generator Number  
- Head

Output:

- Capacity value (MW)

Calculate Head value with formula  $Capacity = \frac{k_1 \times q \times head}{1000}$

```
Capacity = 0
Capacity = (Generator(generator#).k1 * Generator(generator#).q * Head) / 1000
If Capacity > Generator(generator#).Capacity Then
  (NOTE: If the calculated value is greater than unit's capacity, use the unit's capacity instead)
  Capacity = Generator(generator#).Capacity
End If
Return Capacity
```

### WaterRateFunction

Input:

- Generator Number  
- Head

Output:

- Water Rate

Calculate Water Rate with formula  $WaterRate = \frac{q \times 3.6}{Capacity}$

```
Return WaterRate = (Generator(generator#).q * 3.6) / CapacityFunction(generator#, Head)
```

### A.4 CalculateReservoirLimit

**Input:**

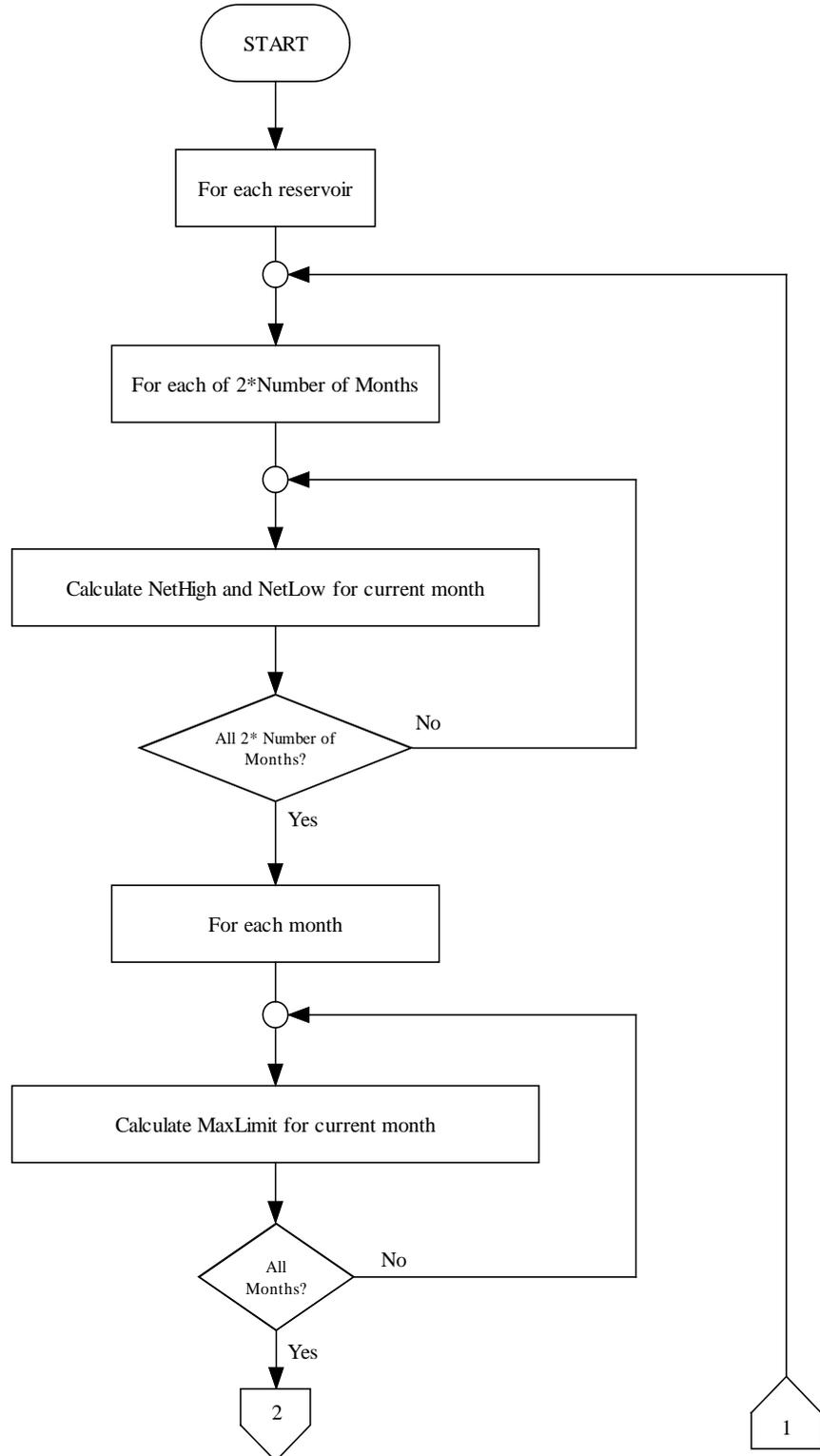
- PMin(reservoir#, month#)
- PMax(reservoir#, month#)
- Reservoir(reservoir#)

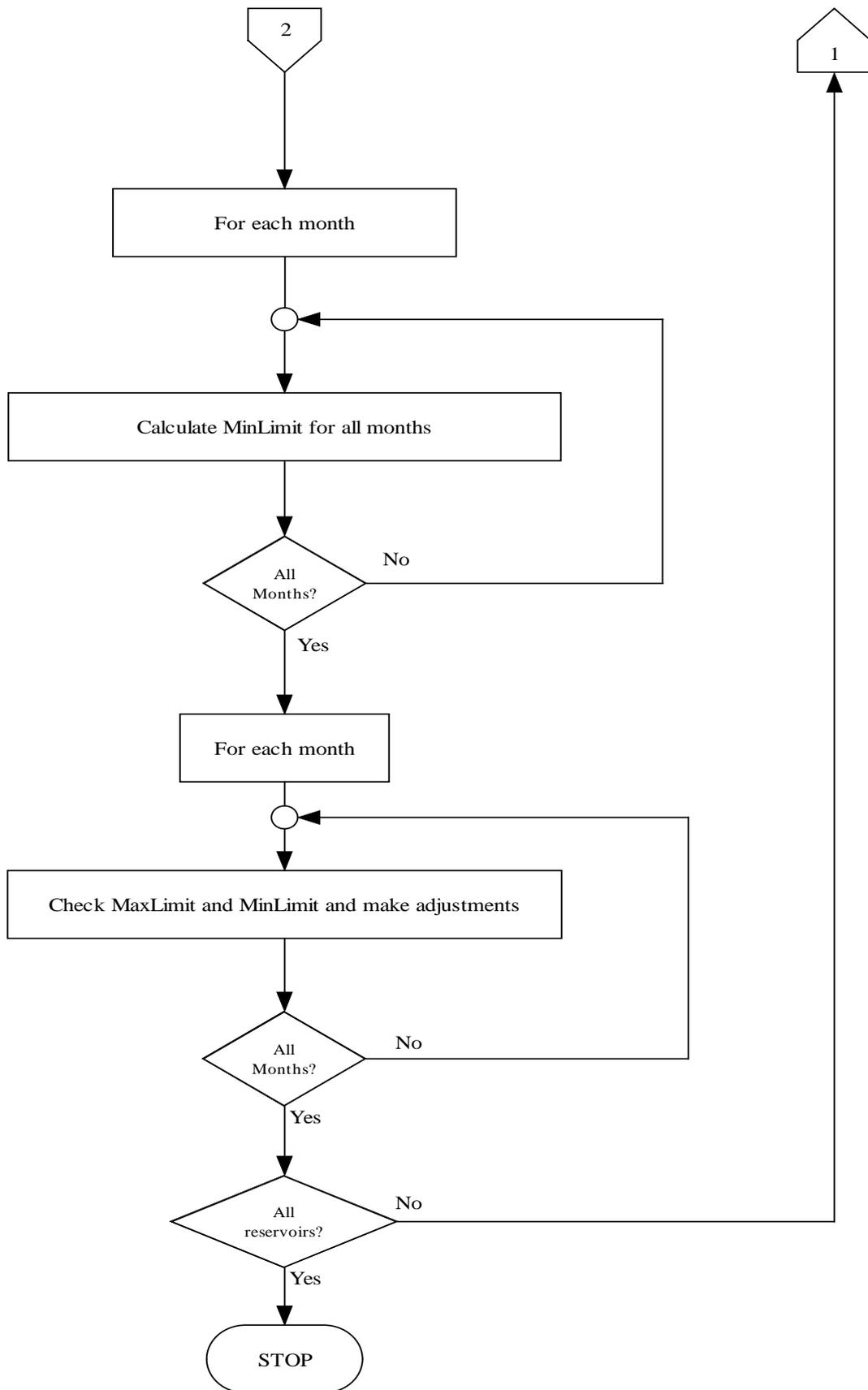
**Output:**

- MaxLimit(reservoir#, month#)
- MinLimit(reservoir#, month#)

**LocalVariable:**

- NetHigh(reservoir#, month#)
- NetLow(reservoir#, month#)





Calculate NetHigh and NetLow for each of 2\*Number of Months

```

NetHigh(current reservoir, current month) = PMax(current reservoir, current month) -
                                             Reservoir(current reservoir).MaxTurbineFlow(current month)

NetHigh(current reservoir, current month + 12) = NetHigh(current reservoir, current month)

NetLow(current reservoir, current month) = PMin(current reservoir, current month) -
                                             Reservoir(current reservoir).PeakOperation(current month)

NetLow(current reservoir, current month + 12) = NetLow(current reservoir, current month)

```

Calculate MaxLimit for each month

```

MaxLimit(current reservoir, current month) = Reservoir(current reservoir).Maxcapacity
next month = current month + 1
Do
  MaxLimit(current reservoir, current month) = MaxLimit(current reservoir, current month) -
                                                NetHigh(current reservoir, next month)
  next month = next month + 1
Loop Until NetHigh(current reservoir, next month) < 0

```

Calculate MinLimit for each month

```

MinLimit(current reservoir, current month) = 0
next month = current month + 1
Do
  MinLimit(current reservoir, current month) = MinLimit(current reservoir, current month) -
                                                NetLow(current reservoir, next month)
  next month = next month + 1
Loop Until NetLow(current reservoir, next month) < 0

```

Check MaxLimit and MinLimit and make adjustments for each month

```

If MaxLimit(current reservoir, current month) > Reservoir(current reservoir).Maxcapacity Then
  MaxLimit(current reservoir, current month) = Reservoir(current reservoir).Maxcapacity
End If

If MinLimit(current reservoir, current month) < 10% of Reservoir(current reservoir).Maxcapacity Then
  MinLimit(current reservoir, current month) = 10% of Reservoir(current reservoir).Maxcapacity
End If

If MaxLimit(current reservoir, current month) < MinLimit(current reservoir, current month) Then
  MaxLimit(current reservoir, current month) = MinLimit(current reservoir, current month)
End If

```

### A.5 Search

Input:

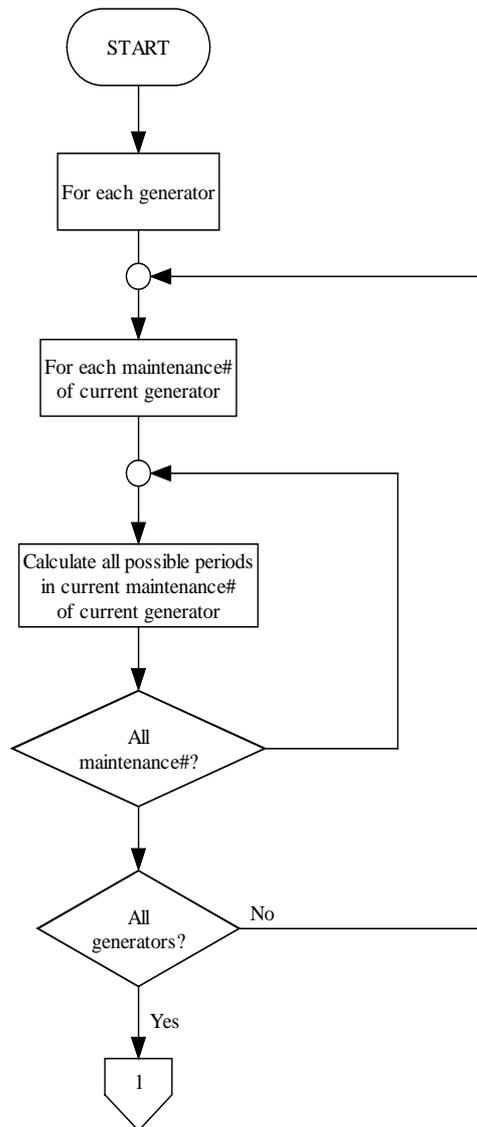
- DataGeneral
- Generator(generator#)
- Reservoir(reservoir#)
- ImportTariffData
- ExportTariffData
- LoadForecast(day#, hour#)
- LoadPeak(day#)
- MaintenanceInfo(generator#)
- InflowForecast(reservoir#)
- MaxLimit(reservoir#, month#)
- MinLimit(reservoir#, month#)
- Average(reservoir#, month#)
- CalcPeriod(reservoir#, month#)
- TotalCalcPeriod(reservoir#)
- Release(reservoir#, month#)
- CapacityGen(generator#, calcP#)
- WaterRate(generator#, month#)

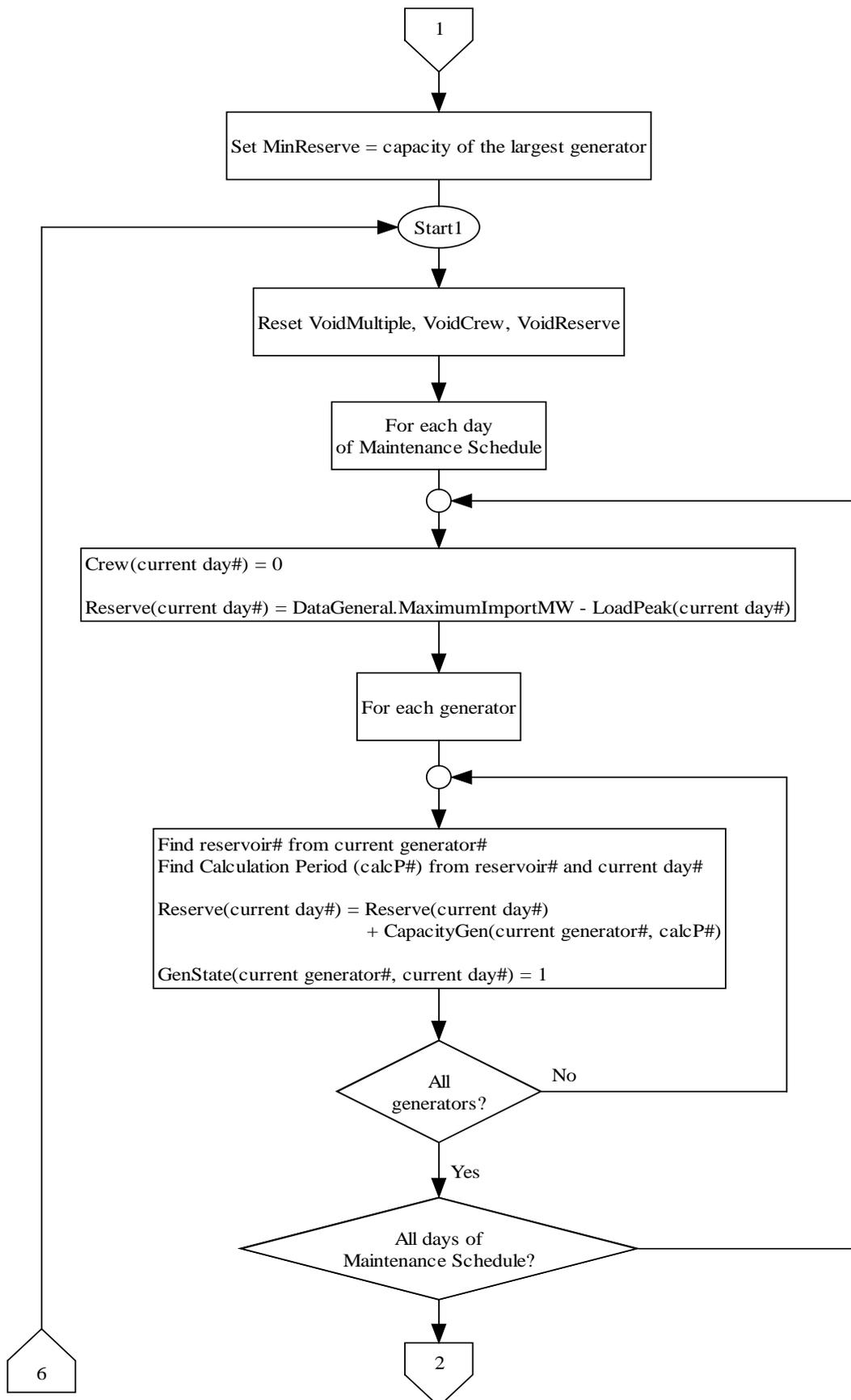
Output:

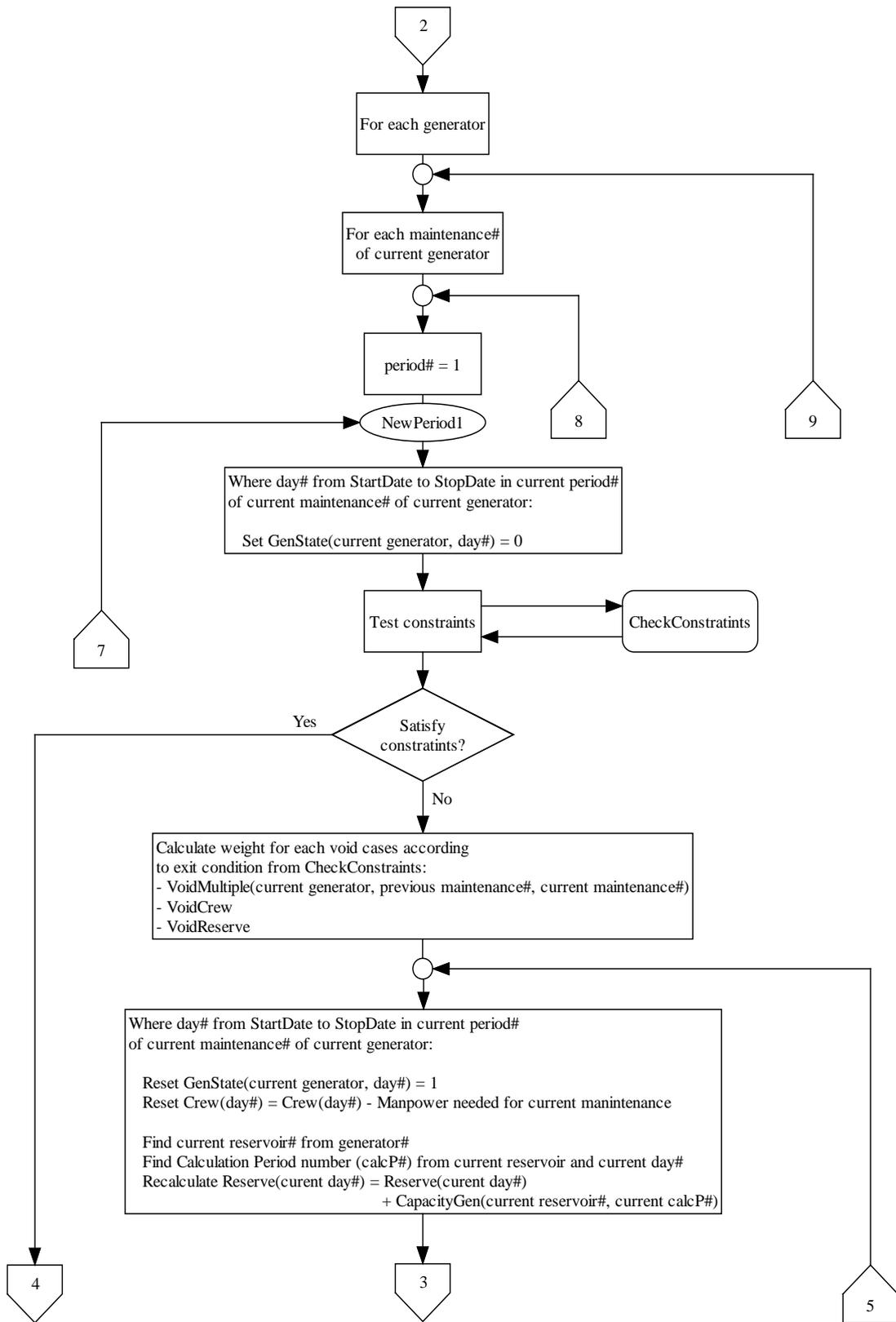
- GenState(generator#, day#)
- GenOutage(generator#, maintenance#, day#)
- ProductState(generator#, hour#)
- WaterRelease(reservoir#, month#)
- SpillWater(month#)

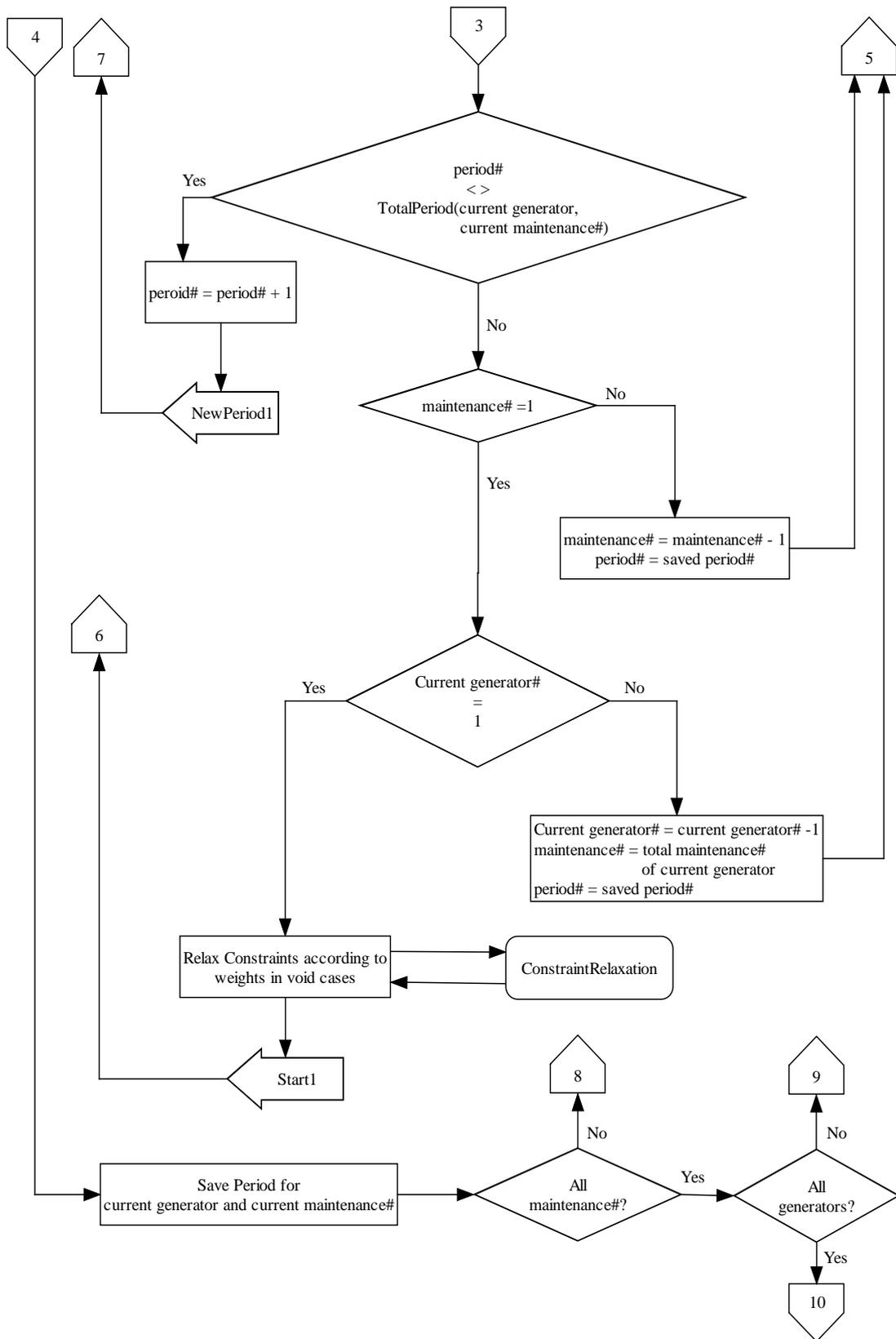
LocalVariable:

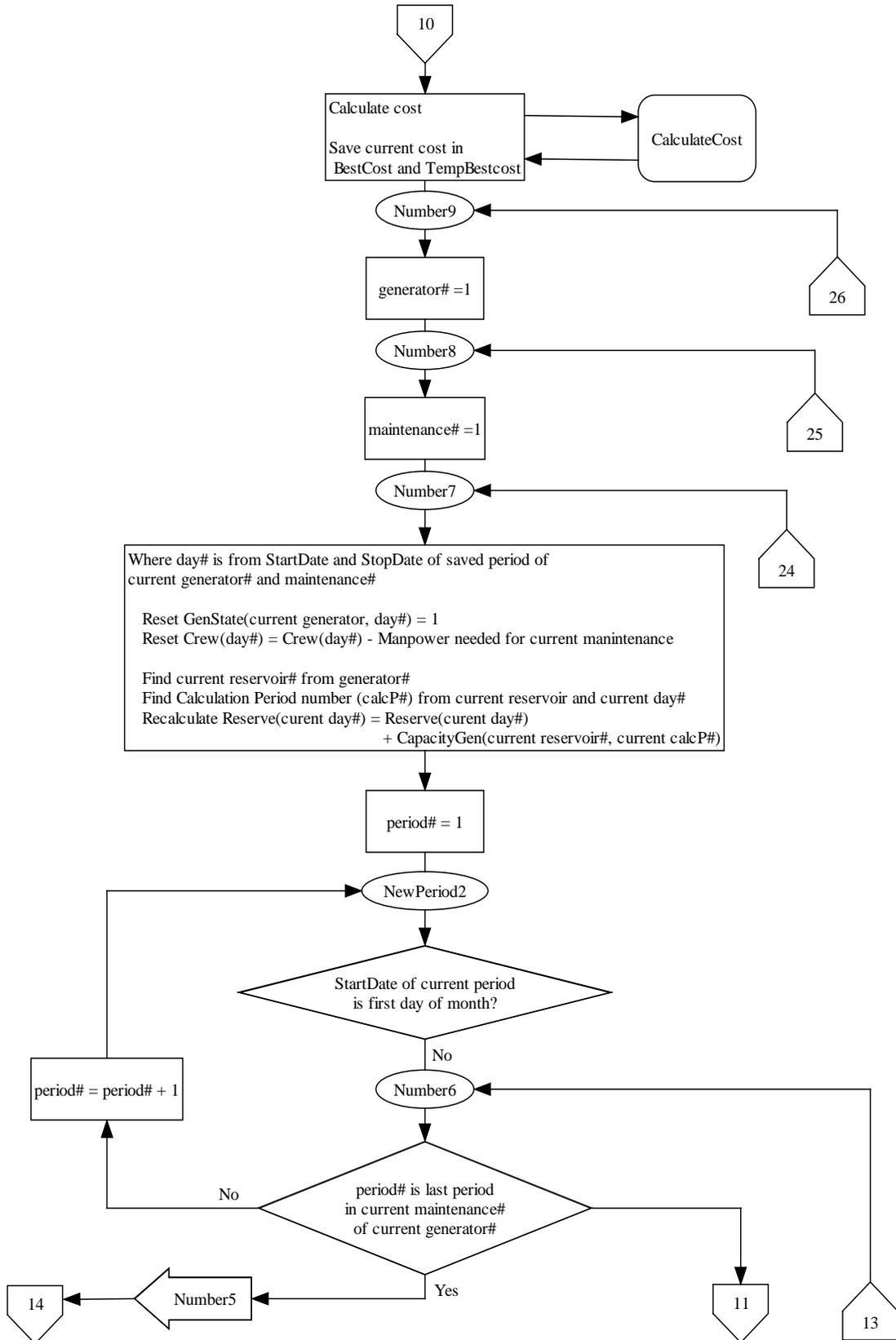
- TotalPeriod(generator#, maintenance#)
- VoidMultiple(generator#, maintenance# - 1, maintenance#)
- VoidCrew
- VoidReserve
- Crew
- Reserve
- MinReserve

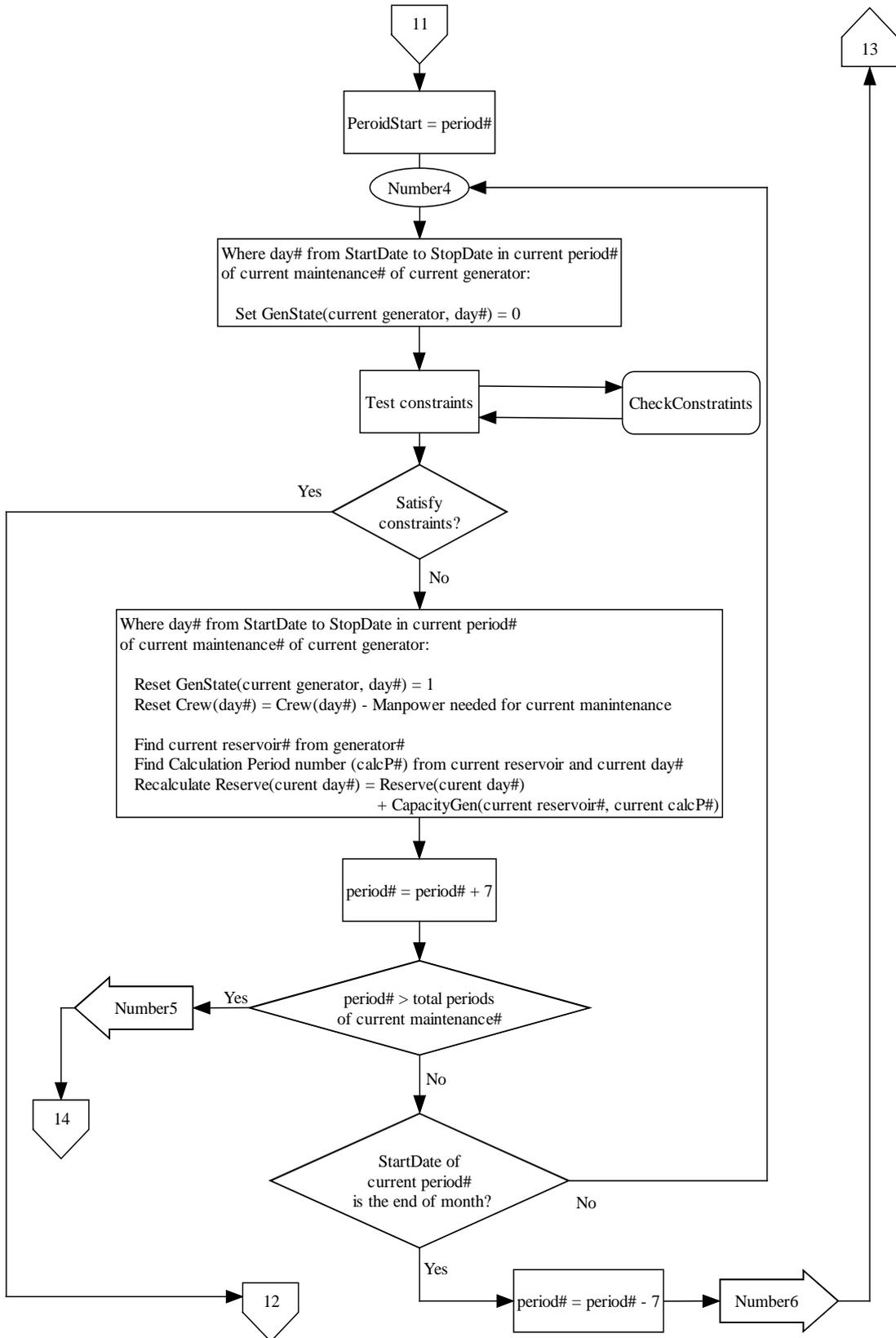


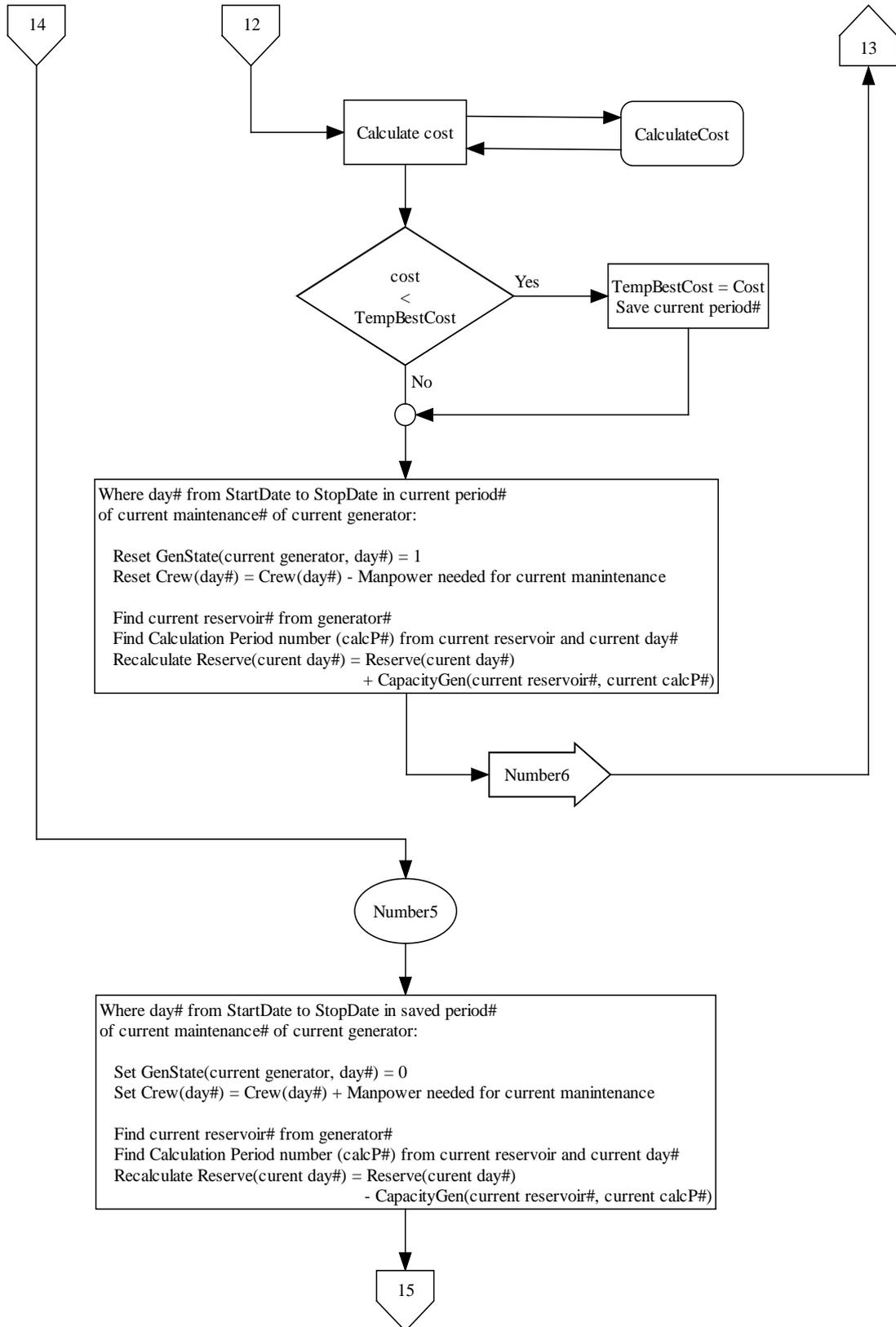


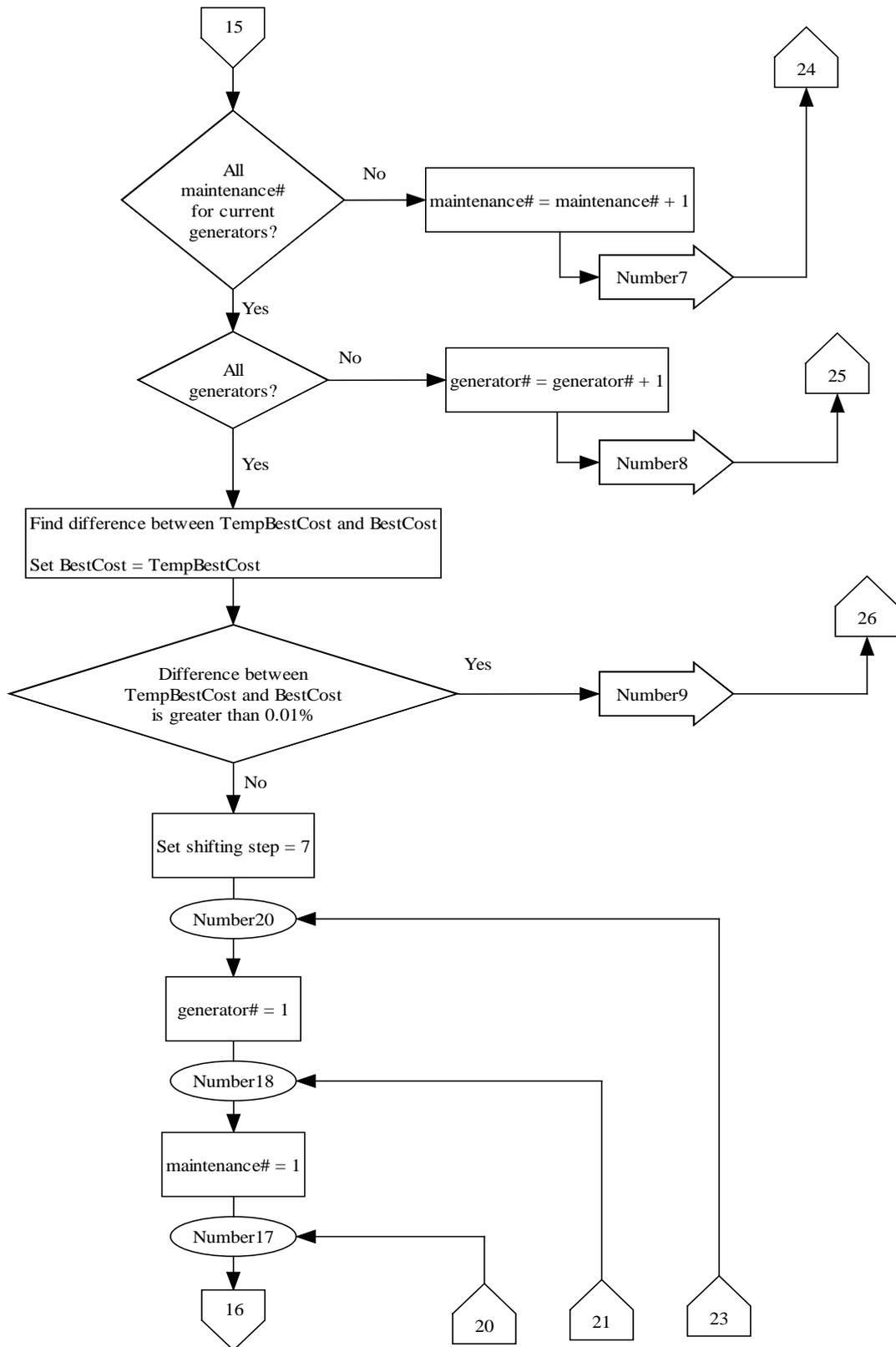


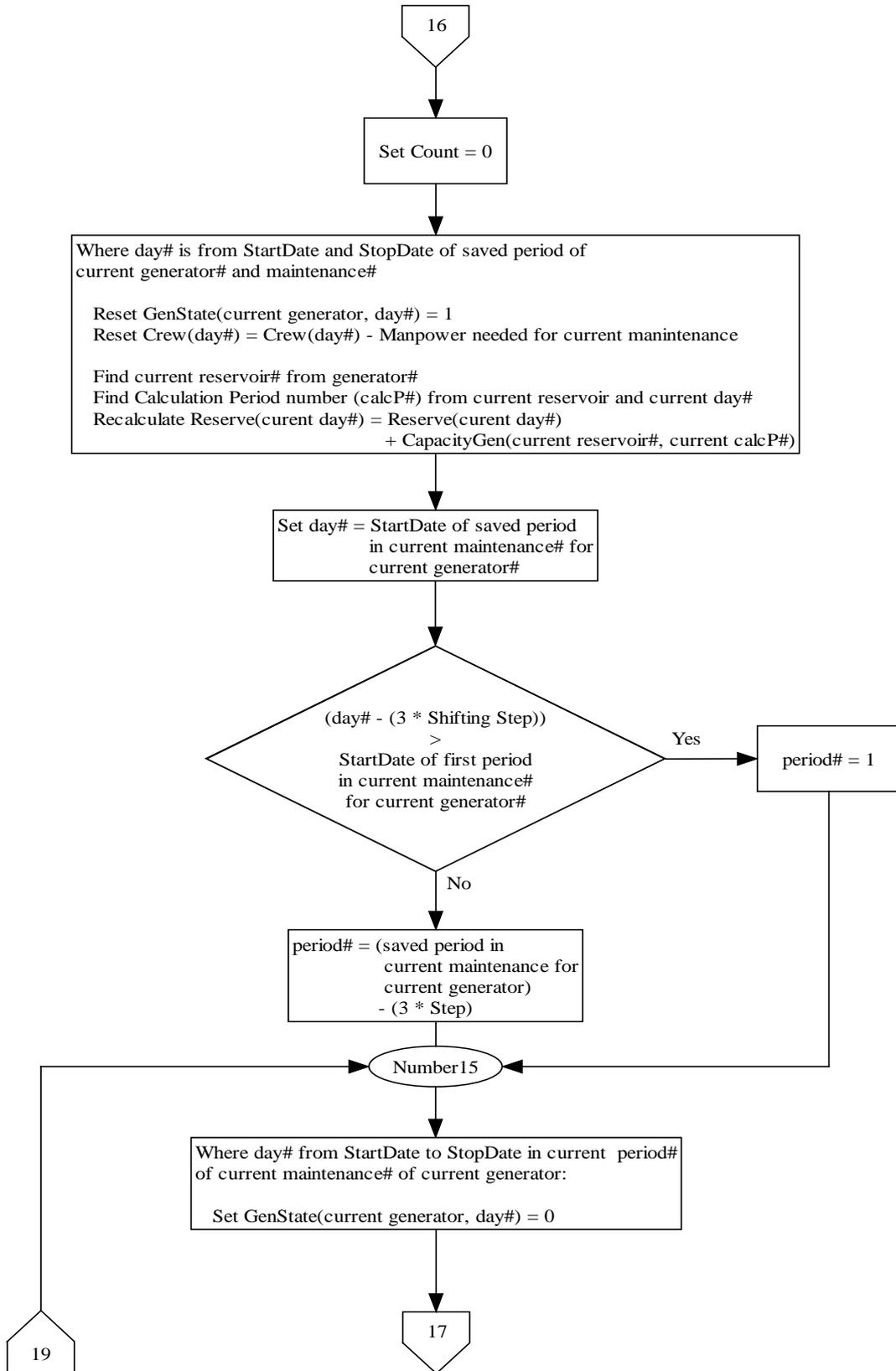


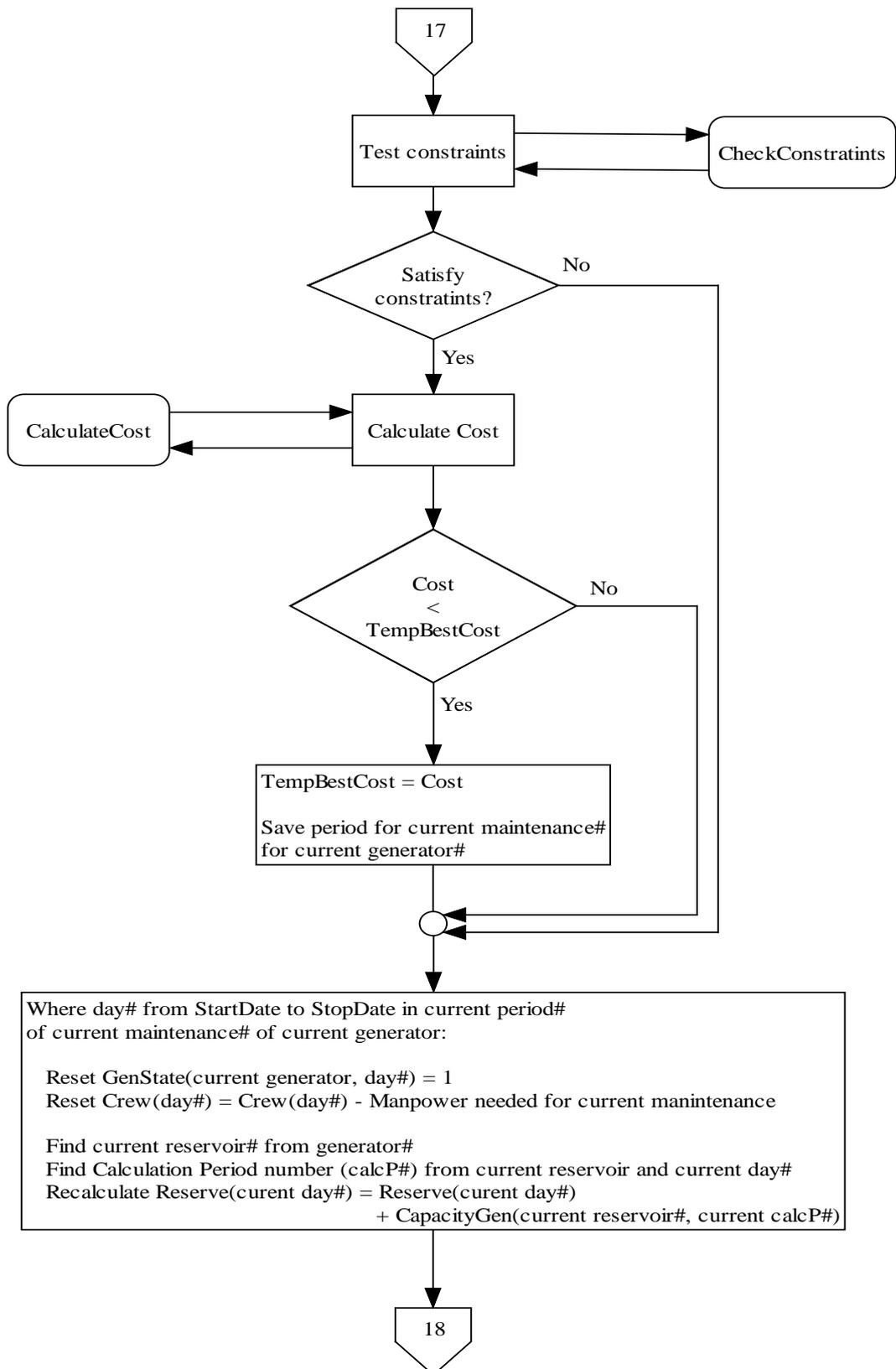


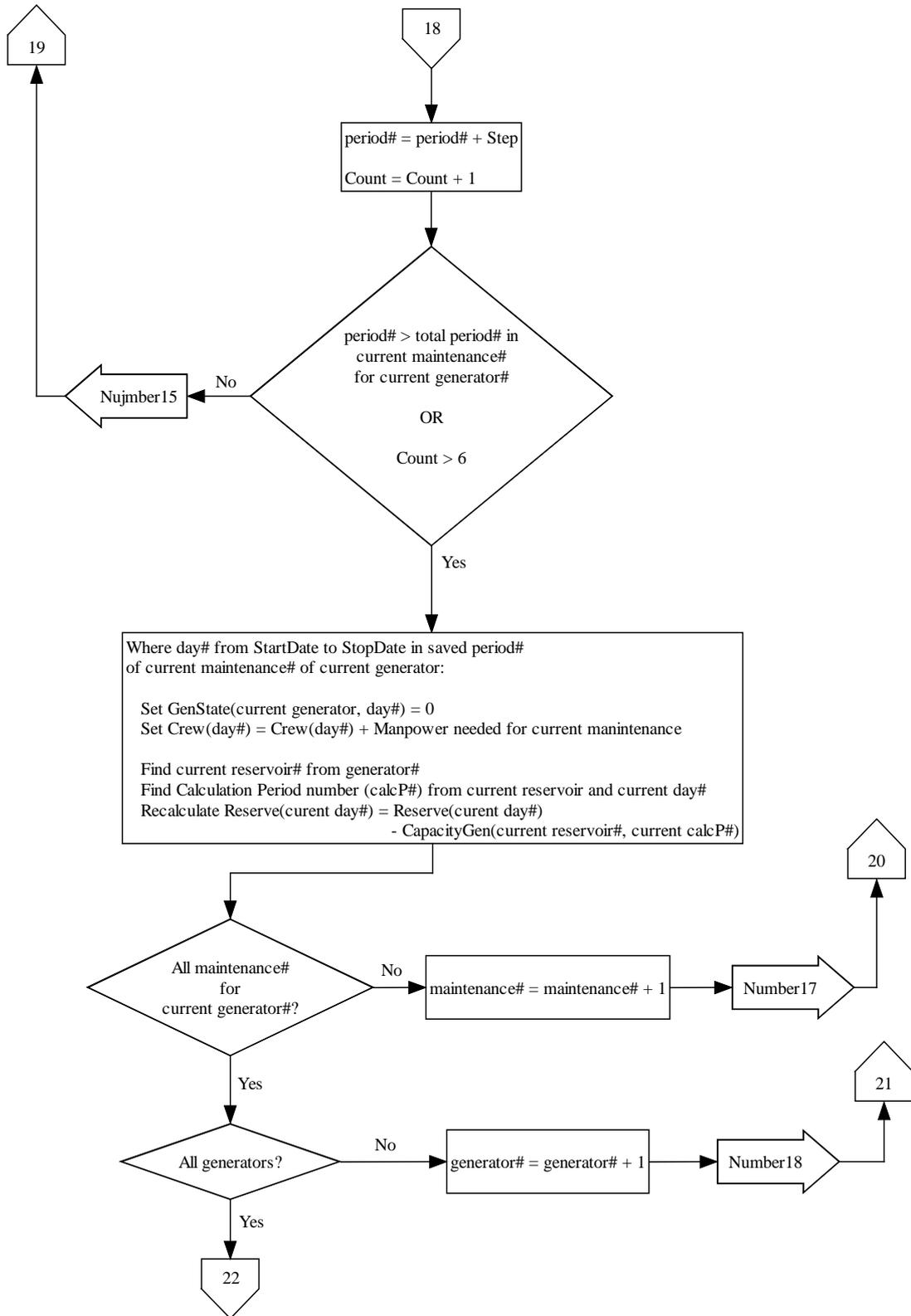


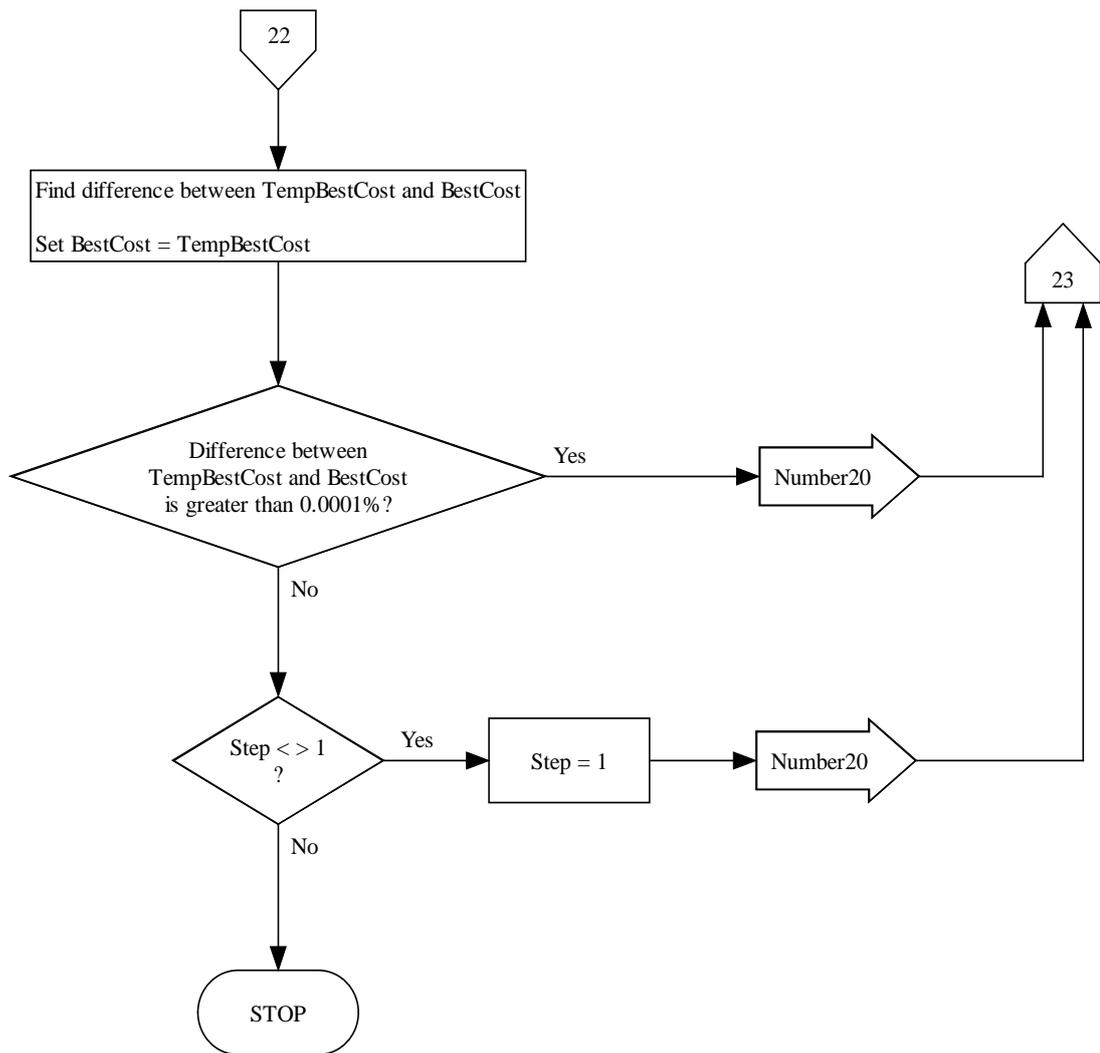












### A.6 Calculate Cost

Input:

- DataGeneral
- Generator(generator#)
- Reservoir(reservoir#)
- ImportTariffData
- ExportTariffData
- LoadForecast(day#, hour#)
- LoadPeak(day#)
- MaintenanceInfo(generator#)
- InflowForecast(reservoir#)
- MaxLimit(reservoir#, month#)
- MinLimit(reservoir#, month#)
- Average(reservoir#, month#)
- CalcPeriod(reservoir#, month#)
- TotalCalcPeriod(reservoir#)
- Release(reservoir#, month#)
- CapacityGen(generator#, calcP#)
- WaterRate(generator#, month#)
- Flow(hour#)
- LoadTemp(hour#)

Output:

- Revenue

LocalVariable:

- SellPrice(hour#)
- BuyPrice(hour#)
- XLoadTemp(hour#)
- XLoadTemp1(hour#)
- ReleaseAcc(1 To 3, 1 To 12)
- ReleaseAccTemp
- ReleaseTemp
- MaxRelease(month#)
- MinRelease(month#)
- WaterEnd(month#)
- OrderHour(hour#)
- OrderHour1(hour#)
- OrderGen(1 To 10)
- PowerGen()
- WaterRateGen()
- Revenue
- Spacing
- SpillTemp As Double

