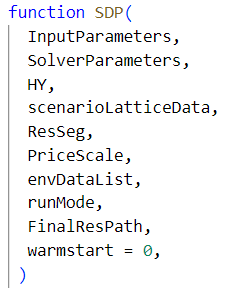
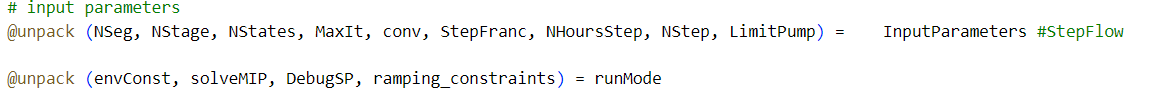
**SDP Algoritmo**

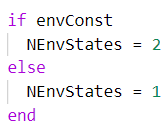
****

Creo la funzione SDP che prende in input:

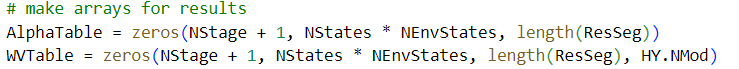
* Input Parameters: devo andare sul foglio run.jl. La funzione set\_parameters, presente nel foglio setInputParameters.jl, permette di tirare fuori i dati da configParameters.in
* Solver Parameters: CPLEX
* HY: Hydraulic data dei due power systems
* scenarioLatticeData: ?
* ResSeg: le combinazioni dei volumi (sono stati divisi i due bacini in 5 fascie. Dunque ci possono essere 25 combinazioni di volume)
* PriceScale: matrice 52(weeks)x56(3h) dei prezzi di un anno
* envDataList: lista dei vincoli ambientali
* runMode: come viene risolto il problema: lista dei true e false
* FinalResPath: ?
* Warmstart = 0: ?

Lista delle variabili che verranno estratte dalla tupla InputParameters e si trovano in configParameters.in altrimenti ogni volta bisognerebbe scrivere ad esempio InputParameters.NSeg e non semplicemente NSeg.

Lista delle variabili che verranno estratte dalla tupla runMode e si trovano in runMode.in



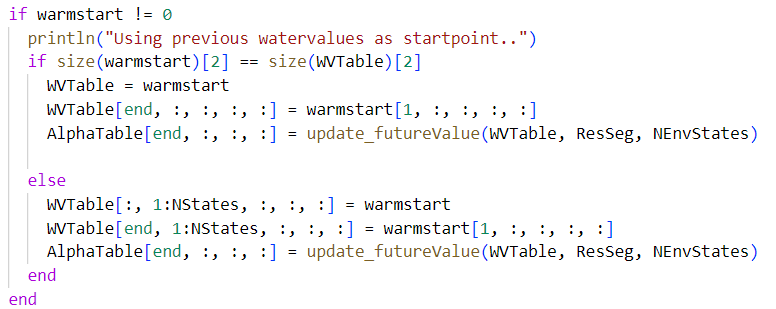
Il valore di NEnvStates verrà impostato a 2 se envConst è true e a 1 se envConst è false. Quindi 2 se ci sono e 1 se non ci sono gli environmental constraints.



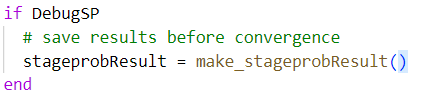
Creo due matrici e le inizializzo a zero. AlphaTable è una matrice tridimensionale che contiene il numero delle settimane +1. AlphaTable contiene 53 settimane perché alla settimana “53” ho inizialmente alpha con tutti valori nulli. Poi inizio il mio algoritmo andando a ritroso. Alla settimana 52 ho quindi dei valori calcolati. Vado avanti a ritroso fino alla settimana 1 a calcolare la matrice AlphaTable. Il problema ipotizza che la matrice debba essere uguale alla settimana 1 e “53”. Alla prima iterazione ovviamente sono diverse perché quella alla settimana “53” era nulla e dunque non vado a convergenza. Per continuare il ciclo inizializzo alla seconda iterazione la matrice AlphaTable con i valori della settimana 1 e vado così avanti fino a che non arrivo a convergenza: la differenza tra la matrice AlphaTable della settimana 53 e settimana 1 è minore di 0.001

Il numero degli stati \* 2 (oppure\*1) e il numero delle combinazioni di volume (25).

La matrice dei WT è quadrimensionale perché viene generata una per il bacino sopra e una per quello sotto.



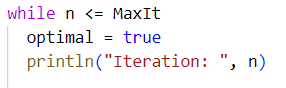
Se ho già valori di WT (warmstart! = 0) allora viene stampato che utilizzo valori precedenti come punto di partenza e non creo la matrice WT con tutti zeri.



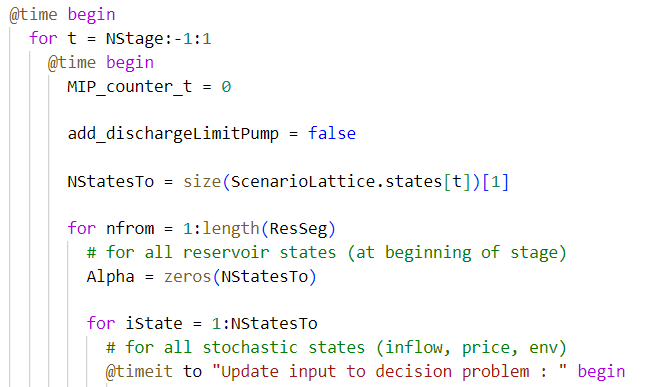
Se il codice va in crash, DebugSP = true allora il codice richiama la funzione make\_stageprobResult()che serve per archiviare i risultati del problema di ottimizzazione.



Serve per calcolare il tempo di esecuzione della funzione BuildProblem

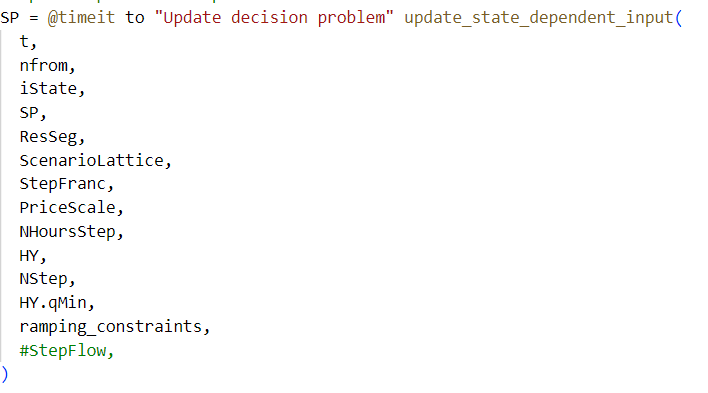


Comincio ad iterare fino a che non arrivo a MaxIt = 100. Il codice stampa l’iterazione a cui sono arrivato.

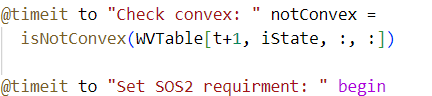


Itero dalla settimana 52 e vado indietro fino alla prima settimana. Itero per tutte le combinazioni di volume e creo la matrice alfa. Itero per tutti gli stati del sistema (5). Per tutti i cicli misuro il tempo impiegato per l’esecuzione.

Ogni settimana guardo se la funzione alpha è convessa o meno. Per essere convessa devo avere tutte le variabili di tipo intero. Nel caso in cui vengano introdotte anche variabili di tipo booleano allora devo risolvere un problema di tipo MIP (Mixed Integer Programming). MIP\_counter\_t serve per contare quante volte il problema non è più lineare ma MIX (ci sono sia variabili intere che booleane).



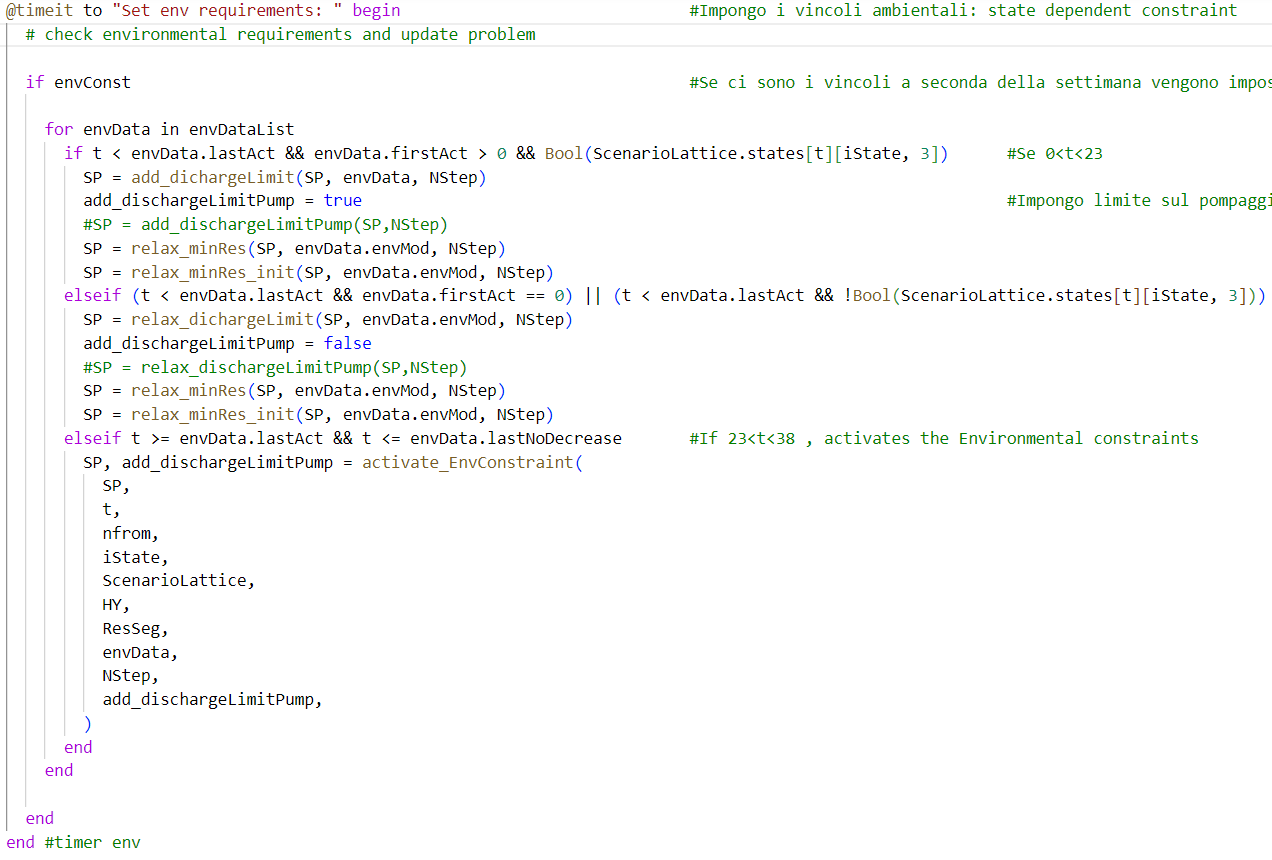
Per tutte le settimane, combinazioni di volume e stati del sistema vado ad aggiornare i prezzi e gli inflow del problema di ottimizzazione.



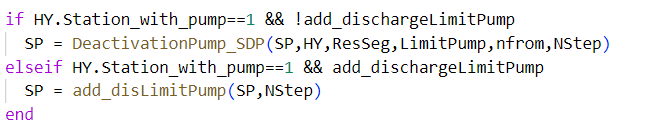
Controllo se la funzione WVTable è convessa o meno. Se non lo è allora la linearizzo usando SOS2. Una funzione è convessa se presi due punti del grafico e tracciando il segmento che li congiunge, questo giace al di sopra del grafico o coincide con una parte di esso. In questo caso riesco a valutare i massimi e i minimi della funzione.

Nel caso in cui la funzione non sia convessa, divido il grafico in n punti e traccio tra due punti consecutivi il segmento che li congiunge. Vado ad analizzare la pendenza tra segmenti successivi e dunque capirò se incontro punti di massimi o di minimo. Questo processo serve per analizzare alpha e dunque capire quali sono i punti di ottimo (massimo) che vanno ad ottimizzare il problema.

Se la funzione non è convessa e se l’iterazione è >1 oppure se warmstart è diversa da zero allora deve essere linearizzata con SOS2. La linearizzazione avviene separatamente per il bacino sopra e quello sotto (HY.NMod == 2 e HY:NMod == 1).

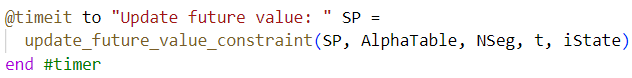
Se i vincoli ambientali sono impostati allora inizio il ciclo for. Se sono tra la settimana 0 e 23 devo imporre il limite sul pompaggio: posso turbinare e pompare ma non posso scendere sotto 52 Mm3.

Se sono tra la settimana 23 e 38 allora devo attivare l’environmental constraint: devo restare sopra una determinata soglia al termine della settimana che nel caso in esame è di 87.44 Mm3.

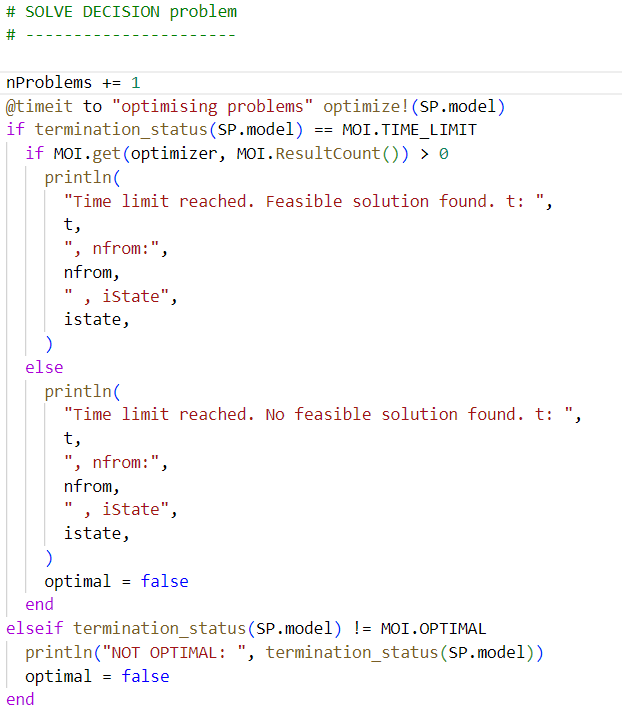


Se ho il pompaggio e se non ci sono restrizioni sul pompaggio allora i vincoli sul pompaggio vengono disattivati.

Se ho il pompaggio e se ci sono restrizioni sul pompaggio allora i vincoli vengono attivati.

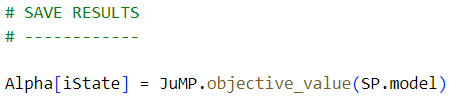


Aggiorno il problema di ottimizzazione e dunque i constraints.

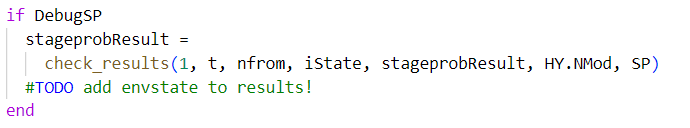


Ottimizzo il problema. Ci sono tre casi:

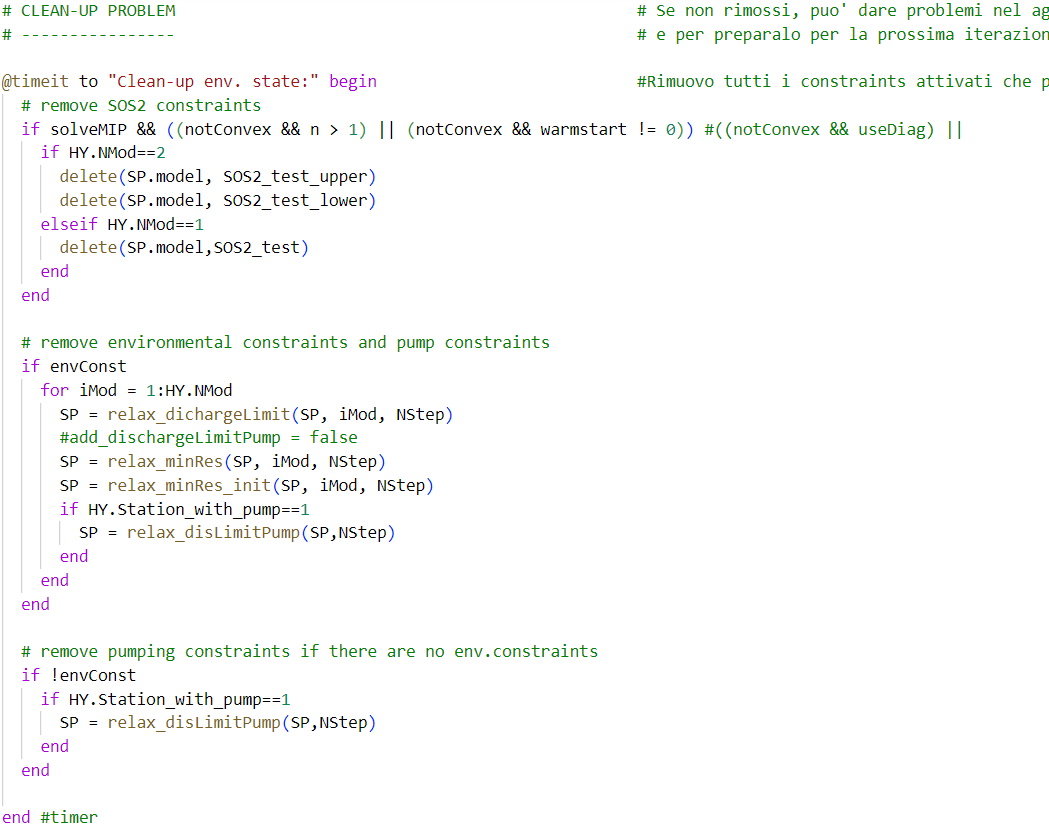
* Il modello raggiunge il time limit e trova una soluzione feasible
* Il modello raggiunge il time limit e non trova una soluzione feasible -> optimal = false
* Il modello non è ottimale -> optimal = false

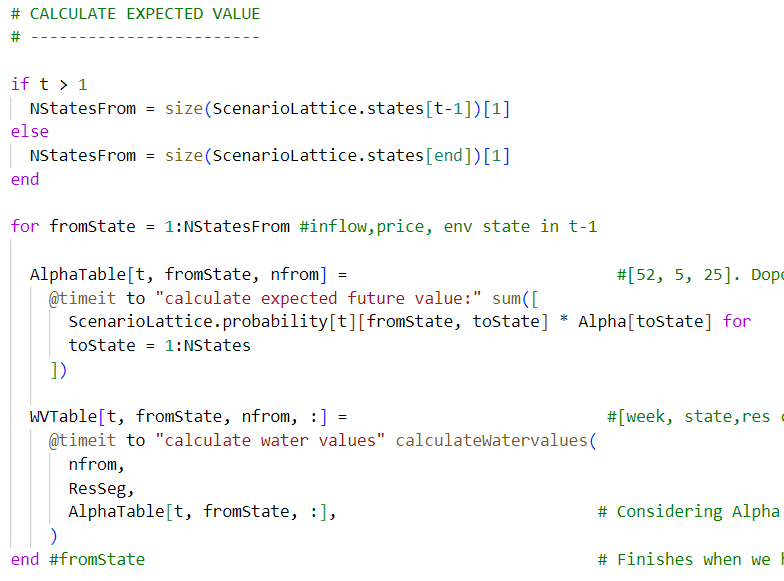


Salvo i risultati della matrice alpha per tutti e 5 gli stati. Ottengo gli expected future profits.



Se il sistema va in crash, va a cercare i risultati nella funzione check\_results che è responsabile di aggiornare stageprobResult con i dati ottenuti dalla soluzione del problema di ottimizzazione.

Vado a rimuovere tutti i vincoli presenti (li pulisco) e riparto dallo stato successivo del problema dove saranno riaggiornati.

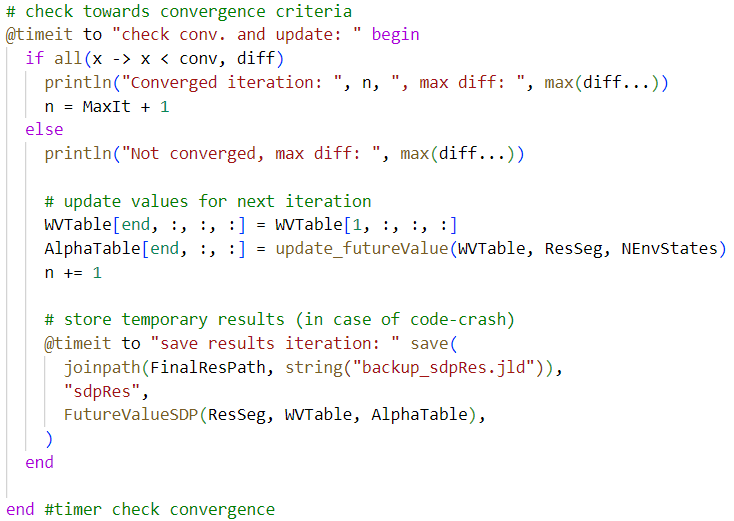
Non ho capito il primo if

Calcolo la matrice AlphaTable = è calcolata come la sommatoria del prodotto fra il profitto alpha e la probabilità di passare da alpha alla settimana t a quella dopo

Calcolo la matrice di WTable. Qual è la correlazione con la matrice alpha? Alpha è in euro. I WV sono in €/Mm3. Dunque la matrice WTable è calcolata dividendo il profitto per la differenza di volume tra la settimana t e la t-1. Il check sulla convergenza è sui WV dunque è necessario il calcolo dei WV dalla matrice alpha.



Tramite la funzione check\_diff vado a determinare la differenza tra la prima “53” e ultima settimana “1” che devono avere valori uguali. Sono fuori dal ciclo for per le settimane: ho già iterato per tutto l’anno.



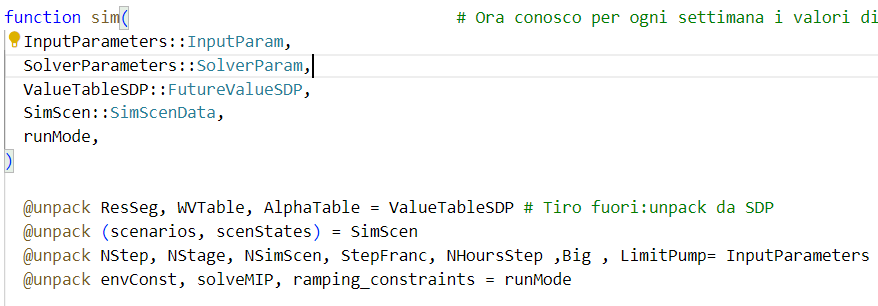
Dopo aver calcolato la differenza con la funzione precedente, se ho raggiunto la convergenza: x<conv allora il problema è finito, altrimenti se non è raggiunta la convergenza devo aggiornare le matrici WTable e AlphaTable che verranno utilizzate come punto di partenza per la prossima iterazione.

I risultati vengono temporaneamente salvati nel file “backup\_sdpRes.jld” nel caso in cui il codice va in crash.

COSA NON HO CAPITO:

* Non ho capito la questione dei MIP: cosa sono e a cosa servono
* Come vengono richiamate le variabili tra i vari fogli. Ad esempio InputParameters. Ho cercato di creare un programma per stampare graficamente i risultati seguendo quello che hai fatto in Plots.jl ma dice che quella variabile non esiste.
* Una spiegazione sulla convessità delle funzioni e perché devono essere linearizzate con SOS2
* Cosa contiene warmstart
* Cosa significa Bool(ScenarioLattice.states[t][iState,3])
* Se c’è un modo per vedere cosa contengono i DataFrame (tipo in RStudio). Inoltre per le funzioni riesco a vedere il riferimento (in quale foglio sono state scritte). Per le variabili ci può fare un’operazione simile a control+tasto sinistro del mouse per vederne il valore?

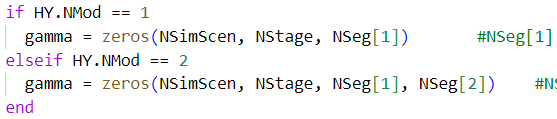
**SIM algoritmo**

Qui non capisco cosa faccia questa parte di codice in cui considera prima se ho due bacini e poi se ho un bacino soltanto.

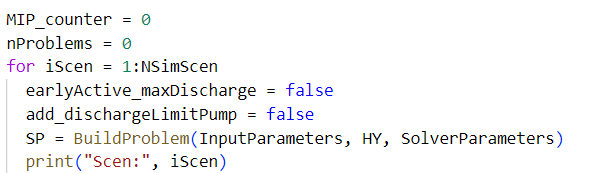
La parte di simulazione viene fatta successivamente a SDP. Quindi prendo in input quelle variabili al cui interno ci sono gli elementi in unpack.



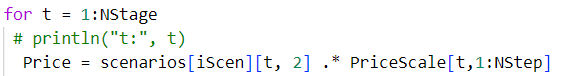
Creo tutte le matrici e le inizializzo a zero. Poi iterando vanno inseriti i valori. Sono poi quelle che verranno inserite nei fogli Excel sulla cartella risultati.



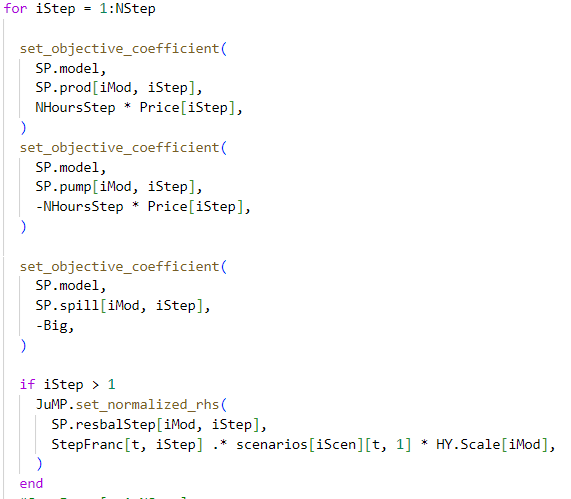
Cosa rappresenta gamma? Gamma viene usato per linearizzare la funzione (“sono le pendenze”)



Inizio a calcolarmi i valori delle matrici nulle create sopra per i 100 scenari.

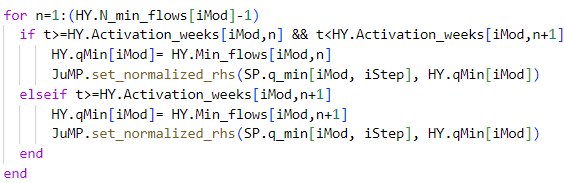


Per ogni settimana (da 1 a 52) vado a calcolarmi il prezzo. Viene fatta una moltiplicazione con i fattori scala dei prezzi.

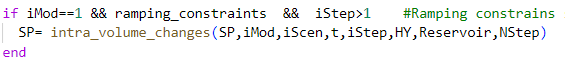


Per ogni Step (56) cambiano i prezzi, le quantità d’acqua. Setta il coefficiente della funzione obbiettivo. Nella funzione obbiettivo , davanti a prod nella funzione object in stageprob metto il coefficinete di set\_objective\_coefficinet.

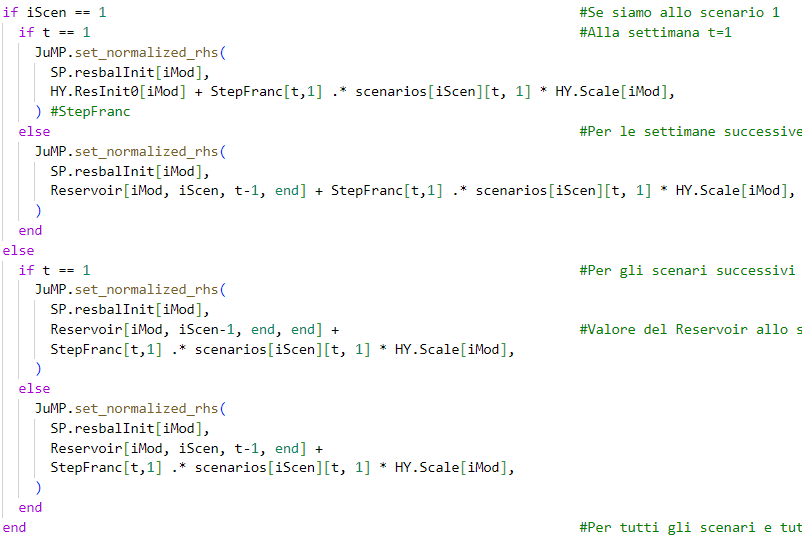
If iStep: StepFranc è la matrice di variazione 3h. Scenario è l’inflow totale della settimana t all’anno simulato iScen e HY.scale è il fattore scala.



Serve per cambiare i valori di MEF. Aggiorno i valori di qmin in stageprob su Minimun environmental flow nella funzione constraint

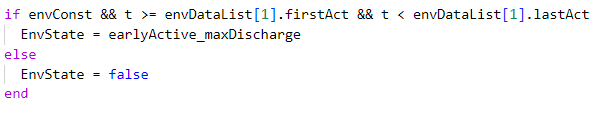


Vado a considerare se sono sul bacino sopra, se ho i ramping constraint (sono solo sul bacino superiore) e per tutti e 56 gli step escluso il primo.

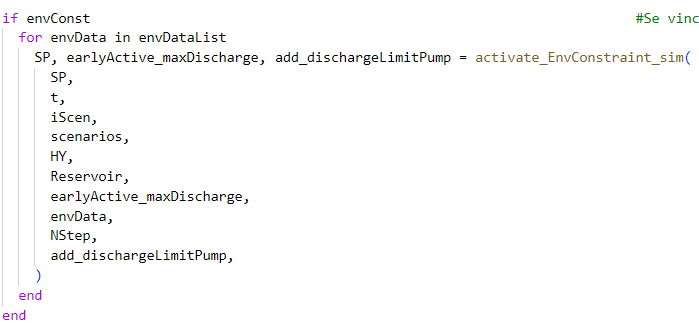


Se siamo allo scenario 1 e alla prima settimana allora ??? per tutte le altre settimane ma sempre allo scenario 1 va a ???. Per tutti gli altri scenari e alla prima settimana…., per tutti gli altri scenari e le altre settimane.

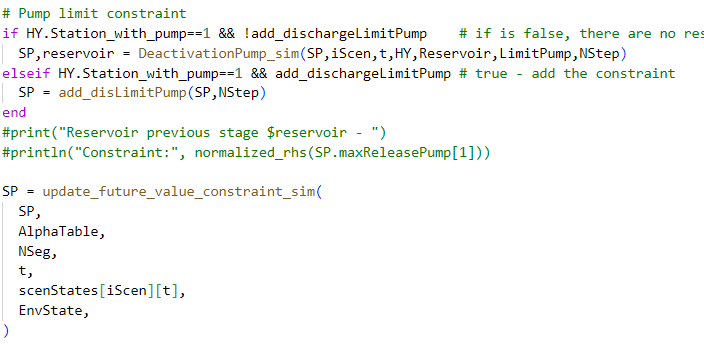
Aggiorno i volumi. Il volume dello scenario 80 alle ultime 3h deve essere …. Dello scenario 81 alle prime 3h



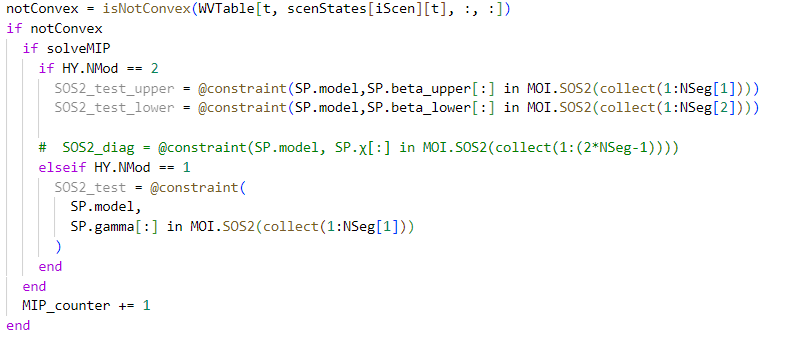
Se ci sono gli environmental contraints e sono in determinate settimane dell’anno allora vado ad attivare l’earlyActive\_maxDischarge.



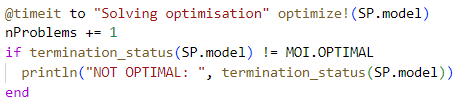
Vado ad attivare se sono attivi gli environmental contraint la funzione activate\_EnvConstraint\_sim.



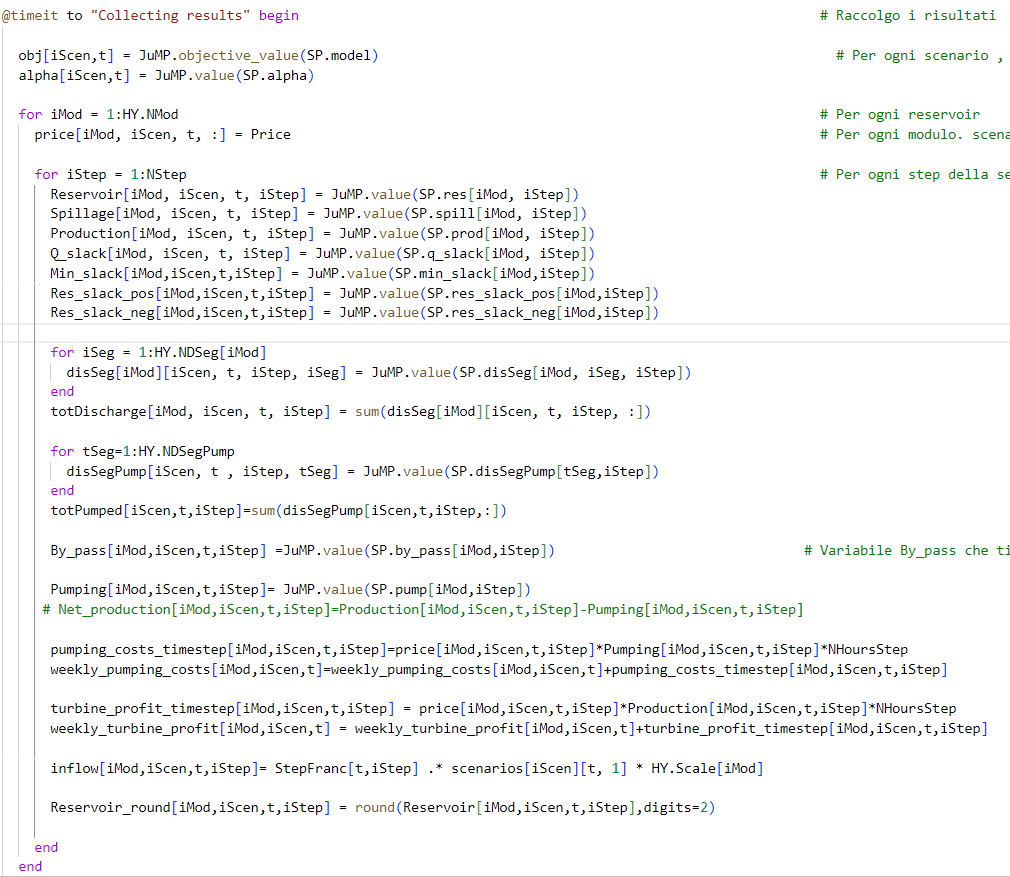
La prima parte serve per attivare o disattivare la pompa considerando il volume limite entro il quale si può fare. Quindi se sono attivati i limiti o meno vado ad aggiornare il modello SP.



Vado a vedere la convessità della matrice WT. Se non è convessa ho i due casi (se ci sono due bacini o un bacino solo). Va a linearizzare?



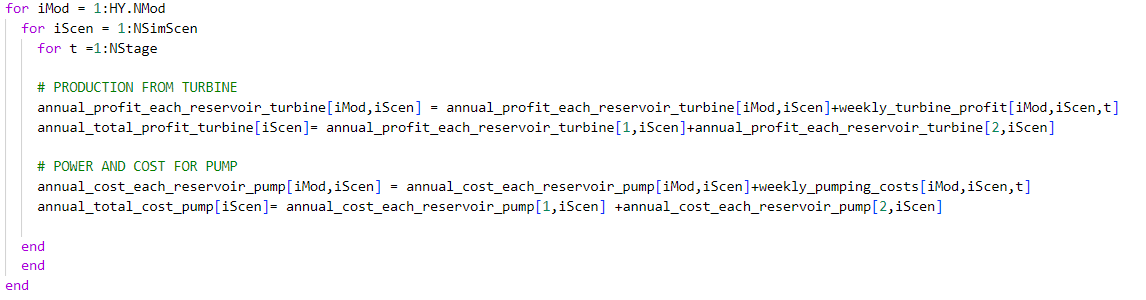
Faccio girare il modello per ottimizzarlo.

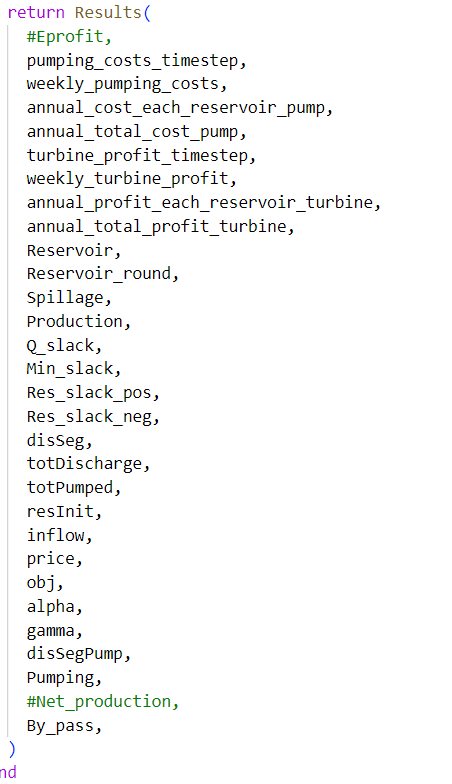


Inizio a collezionare tutti i risultati. Per ogni scenario e settimana aggiorno obj? e la matrice alpha.

Per ogni bacino, scenario (1:100), settimana (52), step (56) vado ad aggiornare il vettore dei prezzi

Per ogni step vado ad aggiorare tutte le matrici.

Per ogni bacino, per ogni scenario e per ogni settimana va a calcolare il profitto per ogni bacino e il profitto totale che poi viene stampato nel return. Viene aggiornato ogni settimana andando ad aggiungere alla somma il profitto della settimana appena passata.



Return di tutte le matrici che inizialmente avevo inizializzato a zero contenenti valori.