ProductionEnvironmentGeneration.R

kaigm

2019-08-30

#############################################################  
# PRODUCTION ENVIRONMENT GENERATION V 0.01  
#   
#############################################################  
  
gen\_ProductionEnvironment <- function(PRODUCTION\_ENVIRONMENT) {  
  
 # if vary\_demand==0   
 # DEMAND\_BASE = lognrnd(1,VOL\_VAR,NUMB\_PRO,1);  
 # else  
 # rng('shuffle')   
  
  
## ====================== STEP 1 REALIZED DEMAND GENERATION =========================   
  
units = 10^3  
preDemand = rlnorm(NUMB\_PRO, meanlog = 0, sdlog = 0.1)  
DEMAND = ceiling(preDemand/sum(preDemand)\*units)  
  
barplot(sort(DEMAND))  
  
  
 # ## ====================== STEP 1 Determining the ACT\_CONS\_PA =========================  
 #   
 # ProductionEnvironment.NUMB\_RES = NUMB\_RES; #Amount of processes  
 # ProductionEnvironment.NUMB\_PRO = NUMB\_PRO; #Amount of products  
 #   
 #   
 # ## ====================== STEP 2 Determining the amount of cost categories (fix vs. variable costs) =========================  
 #   
 # if VOL\_SHARE\_RES == -1  
 # VOL\_SHARE\_RES\_MIN = 0.3;   
 # VOL\_SHARE\_RES\_MAX = 0.7;  
 # VOL\_SHARE\_RES = VOL\_SHARE\_RES\_MIN + (VOL\_SHARE\_RES\_MAX-VOL\_SHARE\_RES\_MIN).\*rand(1,1);  
 # else  
 # end   
 #   
 # ProductionEnvironment.UnitSize=floor(VOL\_SHARE\_RES\*ProductionEnvironment.NUMB\_RES);  
 # ProductionEnvironment.BatchSize=ProductionEnvironment.NUMB\_RES-ProductionEnvironment.UnitSize;  
 #   
 # ## ====================== STEP 2.b Determining a DMM (RES\_CONS\_PAT); =========================  
 #   
 # %% Randomization and setting clear design points.   
 # if DENS == -1  
 # DENS\_MIN = 0.4;   
 # DENS\_MAX = 0.7;  
 # DENS\_RUN = DENS\_MIN + (DENS\_MAX-DENS\_MIN).\*rand(1,1);  
 # else  
 # DENS\_MIN = DENS-0.1;  
 # DENS\_MAX = DENS;  
 # DENS\_RUN = DENS\_MIN + (DENS\_MAX-DENS\_MIN).\*rand(1,1);  
 # end   
 #   
 #   
 #   
 # %DENS\_RUN = 1 ;   
 #   
 #   
 # [RES\_CONS\_PAT,CHECK] = genRES\_CONS\_PAT(ProductionEnvironment,DENS\_RUN,COR); % generate res\_cons\_pat  
 # %[RES\_CONS\_PAT,CHECK] = genRES\_CONS\_PAT2(ProductionEnvironment,DENS\_RUN,COR); % generate res\_cons\_pat  
 #   
 # ## ====================== STEP 2.b Determining a DMM (RES\_CONS\_PAT) ===========================  
 #   
 # RC = genRC(ProductionEnvironment,VOL\_SHARE\_RES,RC\_VAR,TC);  
 # [RES\_CONS\_PATp,CostSystem,CHECK] = genCOST\_CONS\_PAT(ProductionEnvironment,CHECK,RC,RES\_CONS\_PAT,DENS\_RUN,COR);  
 #   
 #   
 #   
 # %% COMPUTING DESCRIPTIVE VALUES   
 # % Computing Resource cost percentage: How many percentage are in the  
 # RC\_sort = sort(RC,'descend');  
 # RC\_20p = sum(RC\_sort(1:(NUMB\_RES\*0.2)));  
 # CHECK.RC\_20p = RC\_20p./TC\*100;  
 #   
 # % Computing Output distribution percentage; How many percentage are in the % 20% values?  
 # DEMAND\_sort = sort(TQ,'descend');  
 # DEMAND\_20p = sum(DEMAND\_sort(1:(NUMB\_PRO\*0.2)));   
 # CHECK.DEMAND\_20p = DEMAND\_20p./sum(TQ)\*100;  
 #   
 # % Computing heterogeneity   
 # [CHECK] = measure\_heterogeneity(RES\_CONS\_PATp,ProductionEnvironment,CHECK);  
 #   
 # % Averange range between lowest and highest consumption.  
 # CORAP\_pre=max(RES\_CONS\_PATp)-min(RES\_CONS\_PATp);  
 # CHECK.CORAP=mean(CORAP\_pre)\*100;   
 #   
return(PRODUCTION\_ENVIRONMENT)  
 } # Function end