

A continuación se detallan los pasos que hay que dar para seleccionar los casos que aparecen en el trabajo.

En este trabajo se ha utilizado el lenguaje de programación C++.

Para variar los parámetros que rigen el sistema hay que modificar las constantes que aparecen al principio del programa.

```
#define N 2000
#define delta_teta 0.015
#define r 1680.0 //Factor de multiplicacion
#define cant 1.000
#define alpha 1.0 //Peso de la norma
#define ciclos 25000 //Número de ciclos en actualización asíncrona
#define umbral 9.5000 //con 0.0 habrá un 100% de cooperadores iniciales y con 10.0 un 0%
#define X 3.5 //SENSIBILIDAD MAXIMA, LA SENSIBILIDAD ESTARÁ EN EL INTERVALO [0,X] CON X:(Xentera+Xdecimal)
#define Max 4 //Max=Xentera+1
```

En este caso tenemos una población de 2.000 agentes, con un factor de multiplicación $r=1680$, con el peso de la norma $\alpha = 1.0$, 5% de cooperadores iniciales y un intervalo de sensibilidad $[0, 3.5]$.

Para elegir la distribución de sensibilidades tenemos que ir a la función que inicializa la sensibilidad y seleccionar la opción que queramos, o mejor dicho anular las demás:

```
void inicializar_teta(struct individuo u[],int tam)
{
    double Num;
    double Numd;//parte decimal, nuestro numero es un decimal Num(Nume,Numd)
    double Nume;//parte entera

    //tenemos un limite X(Xe,Xd), (parte entera,parte decimal)
    for (int cont=0;cont<tam;cont++)
    {
        Nume=(rand()%Max);
        Numd=p_decimal();
        Num=Nume+Numd;
        /* if (Num>X) {Num=X*Num/Max;} //HISTOGRAMA A DERECHAS [0, 2.5](Dominio de potenciales agentes cooperadores)

        if (Num>X) //HISTOGRAMA A IZQUIERDAS [0, 2.5](Dominio de potenciales agentes defectores)
        {
            Num=X*Num/Max;
            Num=Num-2.0;
        }

        */
        // if (Num>X) {Num=Num-1;} //HISTOGRAMA CENTRADO [0, 1.5] (Dominio de agentes indecisos)
        // if (Num>X) {Num=13*Num/28;} //HISTOGRAMA CENTRADO [0, 3.5] (Dominio de agentes indecisos)
        // if (Num>X) {Num=5*Num/11;} //HISTOGRAMA CENTRADO [0, 2.5] (Dominio de agentes indecisos)
        u[cont].teta=Num;
    }
}
```

Por último, para seleccionar el modo de actualización tenemos que seleccionar la cabecera de la función que queramos y si seleccionamos actualización en paralelos (como en este caso) tenemos que anular el sorteo:

```
//FUNCION QUE CALCULA LOS CAMBIOS
//void calcular_variacion(struct individuo w[], int num_trib)////CABECERA DE ACTUALIZACION ASINCRONA
void calcular_variacion(struct individuo w[],int num_trib,int b) ////CABECERA DE ACTUALIZACION SINCRONA (PARALELOS)
{
    int nc;
    double norma;
    struct individuo aux;
    /*int b; //Hay que comentarlo para hacer la actualización en paralelos
    b=0; //Hay que comentarlo para hacer la actualización en paralelos
    b=rand()%N; //Hay que comentarlo para hacer la actualización en paralelos*/

    aux.teta=w[b].teta;
    aux.trib=w[b].trib;
    aux.u=w[b].u;

    if ((w[b].trib)==1.0)
    {
        aux.trib=0.0;
        nc=num_trib-1;
    }
    else
    {
        aux.trib=1.0;
        nc=num_trib+1;
    }
}
```

También tendremos que seleccionar la llamada a la función:

```
//AHORA CALCULAMOS LAS VARIACIONES
//CON ACTUALIZACION EN PARALELOS
for (int cont3=0;cont3<100;cont3++)
{
    num_trib=contribuyentes(v,N);
    norma=alpha*num_trib/N;
    funcion_utilidad(v,N,num_trib,norma);
    for (cont=0;cont<N;cont++)
    {
        calcular_variacion(v,num_trib,cont);
    }

    sens_media=0.0;
    for (int cont2=0;cont2<N;cont2++)
        sens_media=(v[cont2].teta)+sens_media;

    sens_media=sens_media/N;

    sout<<cont3<<" " <<num_trib<<endl;
    fout<<cont3<<" " <<sens_media<<endl;

}

/*//CON ACTUALIZACION ASINCRONA
for (cont=0;cont<ciclos;cont++)
{
    num_trib=contribuyentes(v,N);
    norma=alpha*num_trib/N;
    funcion_utilidad(v,N,num_trib,norma);
    calcular_variacion(v,num_trib);

    sens_media=0.0;
    for (int cont2=0;cont2<N;cont2++)
        sens_media=(v[cont2].teta)+sens_media;

    sens_media=sens_media/N;

    sout<<cont<<" " <<num_trib<<endl;
    fout<<cont<<" " <<sens_media<<endl;

}
*/
```