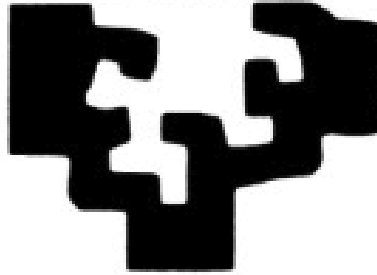


eman ta zabal zazu



Universidad
del País Vasco

Euskal Herriko
Unibertsitatea

Facultad de Informática

Informatika Fakultatea

TITULACIÓN: Ingeniería en Informática

PSOmodels: User Manual

Alumno/a: Maider Gaztelu Aza

Profesores: Blanca Rosa Cases Gutierrez y Abdelmalik Moujahid

Proyecto Fin de Carrera, julio de 2013

Índice general

1. Introducción	9
2. Aplicaciones del programa	11
3. Proceso de instalación	13
4. Elementos de la interfaz gráfica	15
5. ¿Cómo usar PSOmodels?	23

Índice de figuras

4.1. Pestañas de la Interfaz Netlogo	15
4.2. Pestaña de ejecución de Netlogo	16
4.3. Zona 1 en la pestaña de ejecución	16
4.4. Zona 3 en la pestaña de ejecución	17
4.5. Zona 2 en la pestaña de ejecución	17
4.6. Sección 1 de la zona 2: choose model	18
4.7. Sección 2 de la zona 2: run experiment	18
4.8. Sección 3 de la zona 2: run experiment	19
4.9. Sección 4 de la zona 2: parameters	20
4.10. Sección 5 de la zona 2: parameters	20
4.11. Sección 6 de la zona 2: change view	20
4.12. Sección 7 de la zona 2: user guide	21
4.13. Sección 8 de la zona 2: model information	21
4.14. Sección 9(i) y 10 (d) de la zona 2	21

Índice de cuadros

5.1. valores por defecto de la aplicación	24
---	----

Capítulo 1

Introducción

Este manual tiene como objetivo facilitar la experiencia del usuario en el trabajo con el modelo PSOmodels. Está enfocado a utilizar aplicaciones específicas de este modelo a nivel usuario. Por tanto, si desea ampliar este modelo o desarrollar nuevos modelos consulte el manual propio de Netlogo.

Capítulo 2

Aplicaciones del programa

Este modelo de Netlogo permite experimentar con el algoritmo conocido como Particle Swarm Optimization(PSO). A diferencia de otros programas escritos en otros lenguajes, este permite simular y ver en tiempo real el vuelo de las partículas de PSO. Además, permite experimentar con algunas variantes y simplificaciones que diversos científicos realizaron partiendo del algoritmo original. Posee una gran cantidad de parámetros que incluyen valores por defecto recomendados por algunos científicos en sus artículos, pero también permite que el usuario modifique estos parámetros para poder experimentar y comprender mejor el funcionamiento del algoritmo.

Capítulo 3

Proceso de instalación

Disponemos de 2 versiones del programa: (a) una que se ejecuta desde la interfaz de NetLogo que requiere tener en nuestro equipo la carpeta de trabajo de Netlogo 5.0 o superior y (b) otra como Applet que se visualizará en nuestro navegador y requiere al menos Java 5.0.

a)

Necesario:

- NetLogo 5.0.4 que se puede descargar desde:
<http://ccl.northwestern.edu/netlogo/download.shtml>
- Extensión 'goo'¹ que se puede descargar desde:
<https://github.com/downloads/NetLogo/Goo-Extension/goo-20120119.zip>
- Archivo PSOModels.nlogo en cualquier directorio de nuestro ordenador

Ejecutar aplicación desde: PSOModels.nlogo

b)

Necesario: Dentro de una misma carpeta deberemos tener lo siguiente:

- carpeta que contiene la extension 'array'
- carpeta que contiene la extension 'table'
- carpeta que contiene la extension 'goo'
- el archivo 'NetLogoLite.jar' (se puede copiar del directorio donde está instalado NetLogo 5.0)
- el archivo PSOModels.nlogo
- el archivo PSOModels.html

Ejecutar aplicación desde: PSOModels.html

¹La carpeta que contiene la extensión goo deberá ser añadida a la subcarpeta 'extensions' de la carpeta NetLogo 5.0.

Capítulo 4

Elementos de la interfaz gráfica

Cuando arrancamos Netlogo, en la parte superior de la pantalla observamos tres pestañas¹ (ejecutar, información y código) tal como se muestra en la Figura 4.1. En la primera de las pestañas (ejecutar) se encuentra este modelo de PSO y será por tanto la pestaña en la que nos centraremos en este manual. En la segunda pestaña (información) podremos ver información relativa a nuestro modelo y en la última pestaña (código) encontraremos los procedimientos que se encargan de llevar a cabo la ejecución de nuestro modelo, es decir, el código de programación.

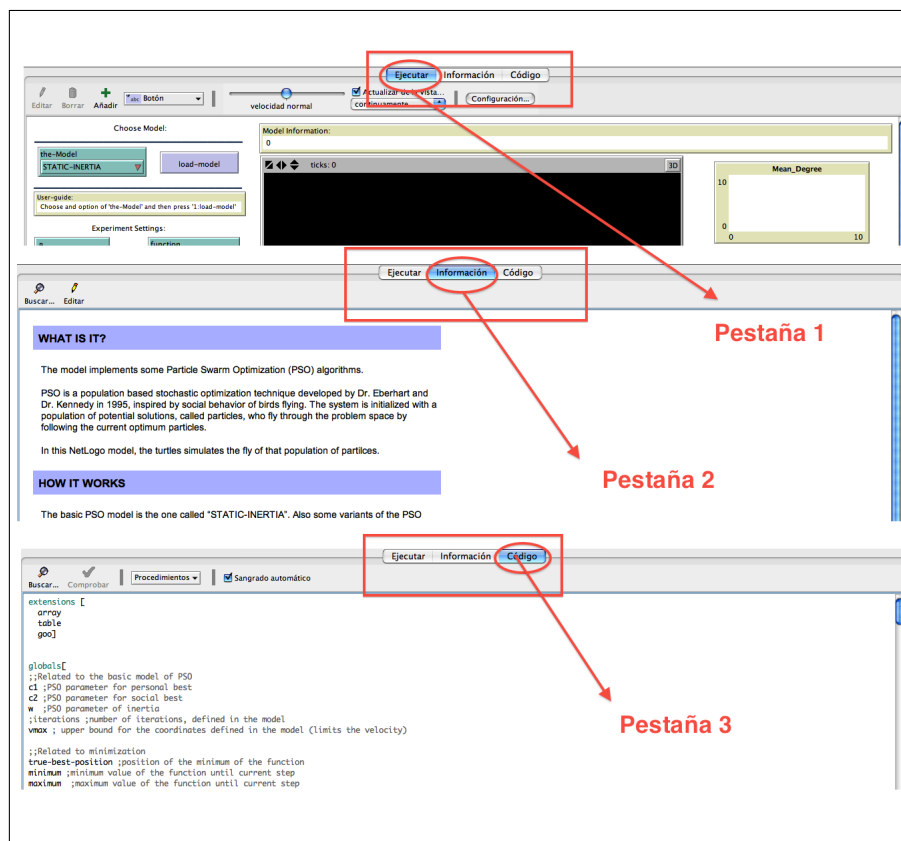


Figura 4.1: Pestañas de la Interfaz Netlogo

¹En la versión Applet no existen las pestañas ejecutar, información y código; simplemente desde el navegador visualizaremos primero la interfaz que tenemos en 'ejecutar' y a continuación la información relativa a este modelo NetLogo.

La pestaña que nos interesa en este manual, a saber, la pestaña 'ejecutar' muestra el siguiente aspecto:

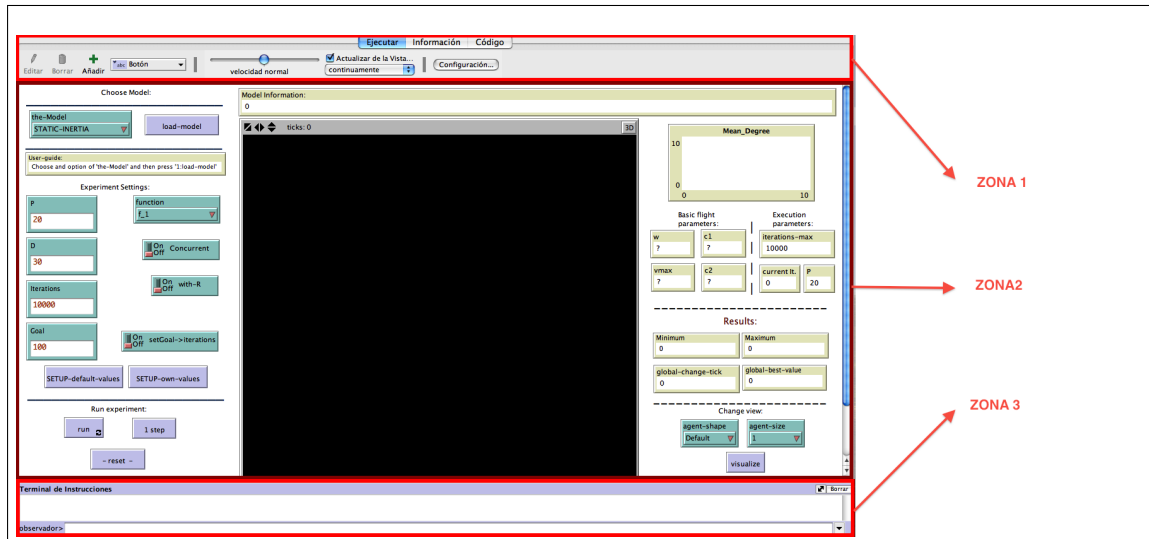


Figura 4.2: Pestaña de ejecución de Netlogo

Tal como indica la Figura 4.2 en esta pestaña de ejecución podemos diferenciar 3 zonas². La primera, situada en la parte superior de la pantalla, permite configurar la vista que más adelante tendremos en el recuadro negro central, que es conocido como 'el mundo' y también definir la velocidad y el modo de actualizar esta vista. Si la casilla 'Actualizar de la vista'³ está activada los cambios en el modelo se nos mostrarán paso a paso. En cambio, si está desactivada, el mundo sólo se actualizará al finalizar la simulación. Si se selecciona 'continuamente', el mundo, variables y gráficas que se presentan en pantalla se actualizan en tiempo real. En cambio, si seleccionamos 'manualmente(ticks)', la pantalla se actualizará al finalizar cada iteración (cada vez que aparezca la palabra tick en la ejecución del programa). Además, tal como muestra la Figura 4.3 permite añadir nuevos elementos a la interfaz del modelo NetLogo.

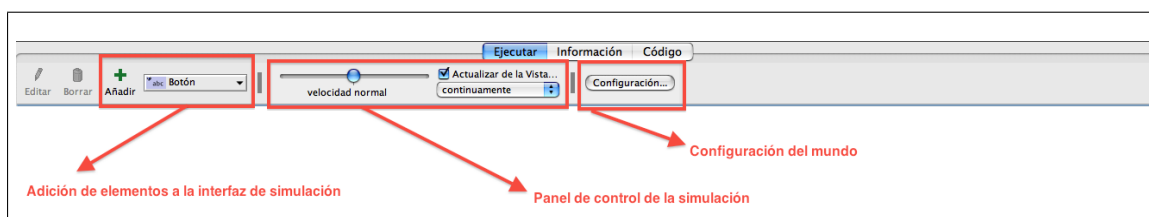


Figura 4.3: Zona 1 en la pestaña de ejecución

Esta primera zona, así como la tercera zona que se encuentra en la parte inferior de la pantalla, es común para cualquier modelo de NetLogo. En esta tercera zona⁴, que podemos ver en la Figura 4.4, disponemos de un terminal donde podemos dar todo tipo de instrucciones de NetLogo desde el *observador*. Este terminal mostraría también cualquier mensaje de salida del programa si lo hubiera.

²En la versión Applet no podemos modificar el modelo NetLogo, por tanto no veremos la zona 1, pero sí podremos modificar la velocidad de la simulación mediante una barra que tendremos en la parte superior de 'el mundo'. Por defecto la velocidad figura como *velocidad normal*. Puede variarse desplazando la barra hacia la derecha (más deprisa) o hacia la izquierda (más despacio). El ordenador procesa los datos a la misma velocidad independientemente de lo elegido en esta barra, pero varía la velocidad con la que estos datos se muestran en la pantalla.

³Para nuestro modelo nos interesa que la vista se actualice continuamente

⁴Esta tercera zona tampoco existe en la versión Applet del modelo

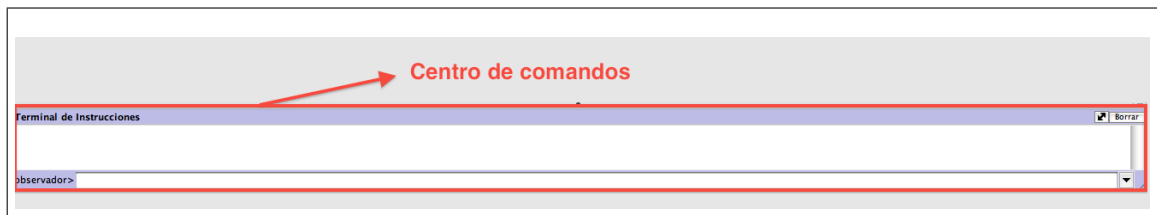


Figura 4.4: Zona 3 en la pestaña de ejecución

La segunda, y más importante de todas las zonas, es la zona central que abarca todo el espacio existente entre la primera y la tercera zona y contiene todos los elementos específicos del programa, a saber, este nuestro modelo *PSOmodels*. Tal como se muestra en la Figura 4.5, hemos agrupado los elementos en las siguientes secciones:

- 1- Choose model
- 2- Experiment settings
- 3- Run experiment
- 4- Parameters
- 5- Results
- 6- Change view
- 7- User guide
- 8- Model Information
- 9- El mundo
- 10- Mean Degree

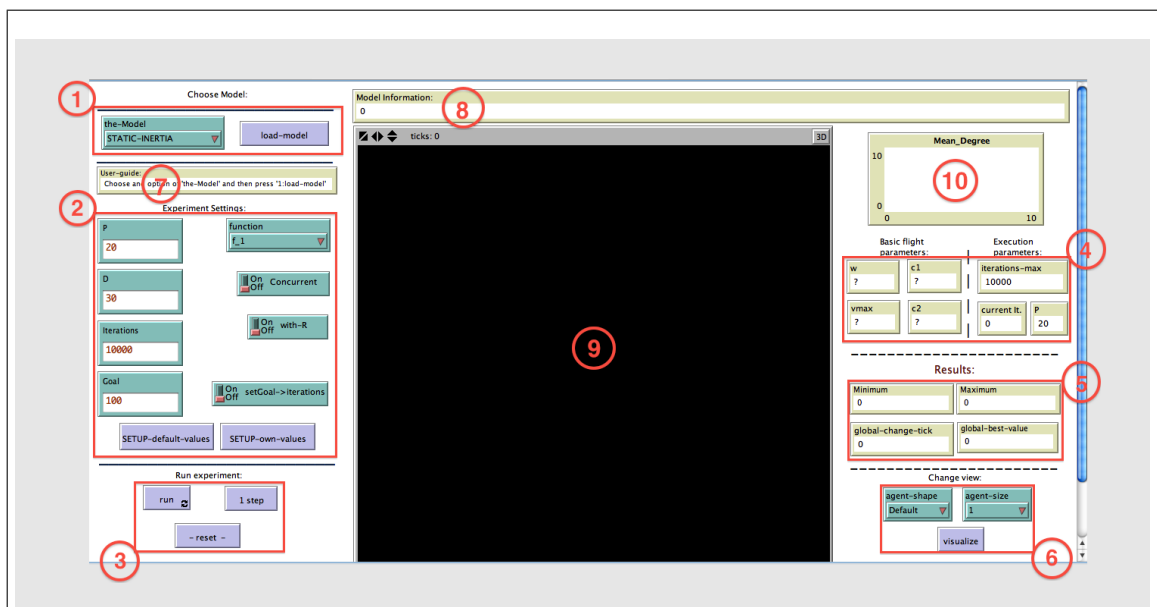


Figura 4.5: Zona 2 en la pestaña de ejecución

1- Choose Model

Permite al usuario seleccionar y cargar uno de los modelos PSO implementados con la configuración de los parámetros de vuelo y ejecución que a este le correspondan, de acuerdo con numerosos artículos científicos sobre PSO.

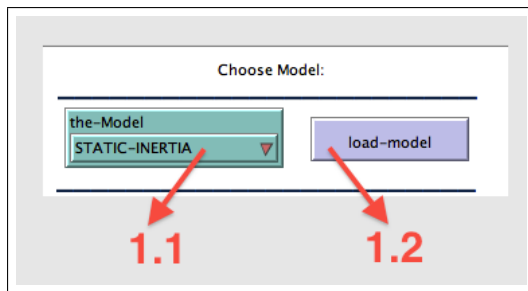


Figura 4.6: Sección 1 de la zona 2: choose model

1.1- the-model:

Se trata de un *seleccionador* que despliega varias opciones (Static-Inertia, Without-Inertia, Randiw, Ldiw, PSO-G, PSO-VG y PSO-PG) para que el usuario seleccione una de ellas. Cada opción se corresponde con un modelo diferente de PSO. Su valor por defecto es el modelo: Static-Inertia.

1.2- load-model

Pulsando este *botón* el programa carga la configuración correspondiente al modelo seleccionado en el elemento 1.1.

2- Experiment Settings

Permite al usuario configurar algunos parámetros del experimento que desea realizar. Encontramos los siguientes elementos:

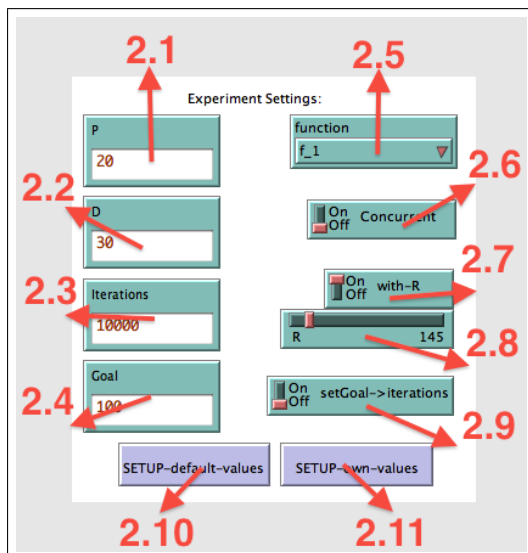


Figura 4.7: Sección 2 de la zona 2: run experiment

2.1- P:

Se trata de un elemento de *entrada* que muestra una variable de entrada del programa y permite al usuario modificarla. Esta variable P representa el tamaño de la población, es decir, la cantidad de partículas de nuestro enjambre. Posee un valor por defecto $P=20$.

2.2- D:

Se trata de un elemento de *entrada* que muestra una variable de entrada del programa y permite al usuario modificarla. Esta variable D representa el número de dimensiones con las que queremos realizar nuestro experimento, si bien gráficamente solo veremos representadas 2 dimensiones. Su valor por defecto depende de la selección del elemento 2.5.

2.3- Iterations:

Elemento de entrada que muestra la variable "Iterations" y permite modificarla. Esta variable define el número de iteraciones máximas permitidas en el experimento. Posee un valor

por defecto de 10000.

2.4- Goal:

Se trata de un elemento de *entrada* que muestra una variable de entrada del programa y permite al usuario modificarla. Esta variable *Goal* determina el objetivo, es decir, el criterio de parada del experimento. En este caso el valor por defecto dependerá de la selección que hagamos en el elemento 2.5, pero en todos los casos estará relacionado con el valor que debe alcanzar el '*global-best-value*', a menos que le indiquemos lo contrario mediante el elemento 2.9.

2.5- function:

Se trata de un *seleccionador* que despliega las opciones: f_0, f_1, f_2, f_3 y f_6. Estas opciones se corresponden con las funciones objetivo que usa la aplicación para realizar los experimentos. Por defecto utiliza la función f_0 si el usuario no selecciona ninguna.

2.6- Concurrent:

Mediante este *interruptor* podemos solicitar que los agentes que representan las partículas ejecuten el código de manera "concurrente". En versiones muy antiguas de NetLogo este tipo de concurrencia

venía por defecto, pero desde NetLogo 4.0 (2007) las instrucciones de los agentes se ejecutan en serie por defecto, es decir, los agentes ejecutan sus comandos de uno en uno teniendo que esperar cada agente a que el anterior finalice todas sus instrucciones. Mediante el interruptor que activa la “conurrencia” se produce una concurrencia simulada a través de un mecanismo de turnos. En nuestro modelo, por defecto este interruptor se mostrará encendido para que la ejecución sea “concurrente”.

2.7- with-R

Este interruptor activa la topología espacial que define el vecindario de una partícula mediante un radio.

2.8- R:

Este *deslizador* permite dar un valor a la variable de entrada R , que representa el radio dentro de unos límites establecidos.

2.9- setGoal->iterations:

Este interruptor activa una opción mediante la cual el *Goal* pasa a ser el número máximo de iteraciones. Por defecto este interruptor esta en modo *off*.

2.10- Setup-default-values

Este botón inicializa el experimento para su posterior ejecución. Los parámetros de inicialización serán los que la función objetivo seleccionada en el elemento 2.5 tenga definidas como tal.

2.11- Setup-own-values

Este botón inicializa el experimento para su posterior ejecución con los parámetros propios del modelo y los que el usuario haya definido en los elementos del 2.1 al 2.9.

3- Run Experiment

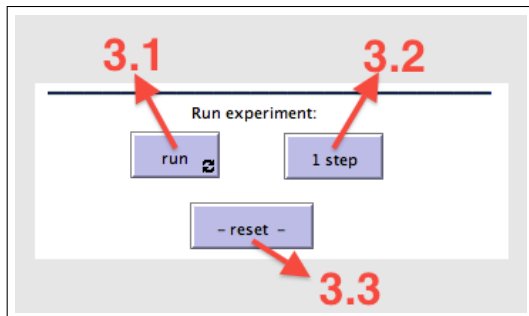


Figura 4.8: Sección 3 de la zona 2: run experiment

3.1- run:

Botón que ejecuta de manera continua todas las iteraciones del experimento configurado en las secciones *Choose Model* y *Experiment Settings* hasta que se satisfaga el criterio de parada.

3.2- 1 step:

Análogo al botón 3.1 pero ejecutando tan solo 1 iteración cada vez que se pulsa el botón.

3.3- reset:

Botón que deja el programa en su estado inicial, eliminando la configuración de los modelos y parámetros anteriores. Es importante pulsar

siempre este botón antes de seleccionar un nuevo modelo de PSO.

4- Parameters

Los elementos del 4.1 al 4.4 muestran los parámetros de control que afectan a la ecuación de vuelo mientras que los del 4.5 al 4.7 muestran los parámetros relacionados con la configuración del experimento.

4.1- w:

Monitor que va mostrando el valor de la variable w en tiempo real. Esta variable representa el peso inercial en la ecuación de vuelo. Su valor por defecto depende del modelo PSO.

4.2- c1:

Monitor que va mostrando el valor de la variable $c1$ en tiempo real. Esta variable representa el coeficiente de la componente cognitiva. Su valor por defecto depende del modelo PSO.

4.3- c2:

Monitor que va mostrando el valor de la variable *c2* en tiempo real. Esta variable representa el coeficiente de la componente social. Su valor por defecto depende del modelo PSO.

4.4- vmax:

Monitor que va mostrando el valor de la variable *vmax* en tiempo real. Esta variable limita la velocidad a un valor máximo definido por el modelo PSO.

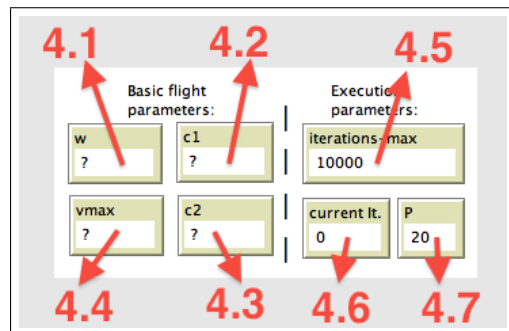


Figura 4.9: Sección 4 de la zona 2: parameters

duos.

4.5- iterations max:

Monitor que muestra el valor de la variable *Iterations*. Esta variable indica el número máximo de iteraciones que ejecutará el experimento. Su valor por defecto es 10000 iteraciones.

4.6- current It.:

Monitor que va mostrando el valor de la variable *T* en tiempo real. Indica la iteración actual. Su valor puede incrementar uno a uno desde 0 hasta iteration max.

4.7- P:

Monitor que muestra el valor de la variable *P*. Esta variable representa el tamaño de la población y tiene asignado un valor por defecto de 20 individuos.

5- Results

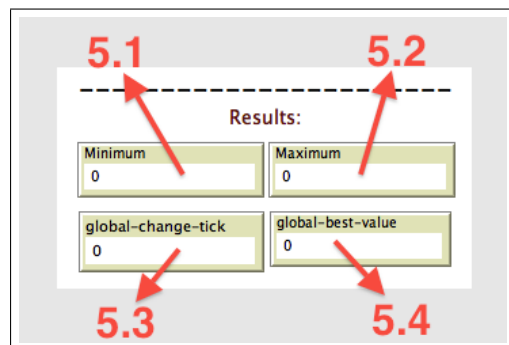


Figura 4.10: Sección 5 de la zona 2: parameters

5.1- Minimum:

Monitor que va mostrando en tiempo real el valor de la variable *Minimum*. Esta variable representa el mínimo valor de la función hallado hasta el momento.

5.2- Maximum:

Monitor que va mostrando en tiempo real el valor de la variable *Maximum*. Esta variable representa el máximo valor de la función hallado hasta el momento.

5.3- global-change-tick:

Monitor que va mostrando el valor de la variable *global-change-tick* en tiempo real. Esta variable indica el número de cambios que se realizan en el gbest position y gbest value.

5.4- global-best-value:

Monitor que va mostrando en tiempo real el valor de la variable *global-best-value*, es decir, el óptimo global hasta el momento.

6- Change View

En esta sección se puede cambiar la vista de las partículas que se encuentran o se verán en *el mundo* (sección 9)

6.1- agent-shape:

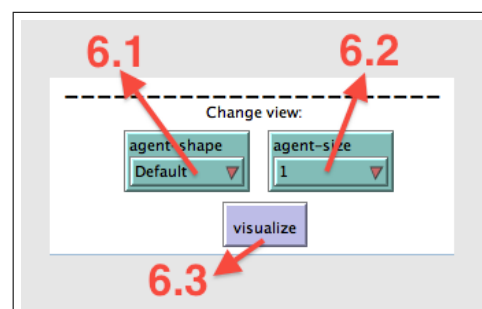


Figura 4.11: Sección 6 de la zona 2: change view

Es un seleccionador que permite escoger la apariencia de las partículas dentro del mundo, pudiendo elegir entre *default* (triángulos) y *pájaros*.

6.2- agent-size:

Es un seleccionador que permite escoger el tamaño de las partículas dentro del mundo, pudiendo elegir entre 1 (por defecto), 3 y 5.

6.3- visualize:

Este botón cambia el aspecto de la vista de las partículas en *el mundo* de acuerdo con lo que el usuario elige en los seleccionadores 6.1 y 6.2.

7- User guide

Este monitor va mostrando al usuario una pequeña guía de cómo proceder.

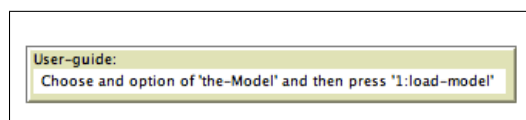


Figura 4.12: Sección 7 de la zona 2: user guide

8- Model Information

Este monitor ofrece en una línea información sobre el modelo seleccionado en el elemento 1.1 y cargado mediante el elemento 1.2

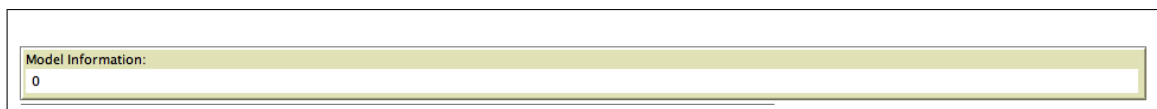


Figura 4.13: Sección 8 de la zona 2: model information

9- El mundo

El “mundo” es el lugar donde se visualiza el comportamiento de los agentes que creamos en nuestro modelo. Está compuesto por una serie de patches o celdas. Se puede elegir el tamaño de cada celda y el número de celdas que hay en el mundo desde el botón “settings” de la zona 1.

10- Mean Degree

Este gráfico representa el valor de una variable respecto al tiempo (ticks). En nuestro caso, la variable degree que indica el número medio de vecinos que tienen las tortugas (partículas de PSO). Como por defecto el elemento 2.7 (with-R) está apagado, el degree será constante a lo largo de toda la ejecución. Si, por el contrario, activamos el interruptor 2.7 la variable degree cambiará a lo largo de los ticks.

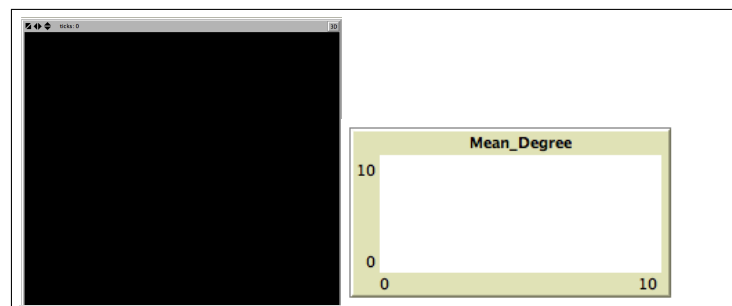


Figura 4.14: Sección 9(i) y 10 (d) de la zona 2

Capítulo 5

¿Cómo usar PSOModels?

Para realizar un experimento con uno de los modelos de la PSO debemos seguir los siguientes pasos: 1- Reset, 2- Load, 3- Setup, 4- Run y 5- Ver resultados.

5.1 Reset

Lo único que debemos hacer es pulsar el botón “reset” que se encuentra en la sección “Run experiment”

5.2 Load

Primero, desde el seleccionador llamado “the model” escogemos el modelo que queremos ejecutar y después pulsamos el botón “load-model”. Inmediatamente podremos ver los nuevos valores que toman algunos parámetros en la sección “Fly parameters”

- Para el caso del modelo PSO, conocido como STATIC PSO, al pulsar el botón “load-model” se mostrarán por pantalla algunas preguntas que nos permiten configurar los parámetros básicos de la ecuación de vuelo. En este caso, cada vez que queremos modificar esos parámetros, podremos hacerlo volviendo a pulsar “load-model”.

- Si queremos cambiar de modelo debemos pulsar Reset, antes de seleccionar y cargar un nuevo modelo.

5.3 Setup

Podemos elegir entre 2 tipos de Setup: 1 - “Setup-default-value” y 2- “Setup-own-value”.

1 – Si utilizamos el “Setup-default-value” se cargará la configuración por defecto de los parámetros del experimento. Esta configuración por defecto es distinta para cada función objetivo. Por defecto se carga la función `f_0` y sus parámetros correspondientes. Si deseamos otra función debemos seleccionarla desde “function” y después pulsar “Setup-default-value”.

2 – Si utilizamos “Setup-own-value” se cargará el experimento con los parámetros que hayamos definido en la sección “Experiment Settings”.

Nota: Los parámetros definidos por defecto provienen del estudio de diversos científicos. Si deseamos un buen funcionamiento del algoritmo no es recomendable introducir valores que se alejen demasiado de la configuración por defecto ya que la PSO es muy sensible a la inicialización de éstos. Es aconsejable pulsar siempre “Setup-default-values” para ver los valores recomendados, después hacer ligeras modificaciones y, entonces, pulsar “setup-own-values”. A continuación, se muestra la tabla con todos los valores que toma el programa por defecto:

		W					C1					C2					VMAX					P				
		F_0	F_1	F_2	F_3	F_6	F_0	F_1	F_2	F_3	F_6	F_0	F_1	F_2	F_3	F_6	F_0	F_1	F_2	F_3	F_6	F_0	F_1	F_2	F_3	F_6
STATIC INERTIA	user	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	max-pxcor	max-pxcor	max-pxcor	max-pxcor	max-pxcor	20	20	20	20	20
	basic	0,9	0,9	0,9	0,9	0,9	2	2	2	2	2	2	2	2	2	2	max-pxcor	max-pxcor	max-pxcor	max-pxcor	max-pxcor	20	20	20	20	20
	clerc	0,729	0,729	0,729	0,729	0,729	1,494	1,494	1,494	1,494	1,494	1,494	1,494	1,494	1,494	1,494	max-pxcor	max-pxcor	max-pxcor	max-pxcor	max-pxcor	20	20	20	20	20
	telea	0,6	0,6	0,6	0,6	0,6	1,7	1,7	1,7	1,7	1,7	1,7	1,7	1,7	1,7	1,7	10000	10000	10000	10000	10000	20	20	20	20	20
WUTHOUT INERTIA		0	0	0	0	0	1,494	1,494	1,494	1,494	1,494	1,494	1,494	1,494	1,494	1,494	max-pxcor	max-pxcor	max-pxcor	max-pxcor	max-pxcor	20	20	20	20	20
RANDIW		eq1*	eq1*	eq1*	eq1*	eq1*	1,494	1,494	1,494	1,494	1,494	1,494	1,494	1,494	1,494	1,494	max-pxcor	max-pxcor	max-pxcor	max-pxcor	max-pxcor	20	20	20	20	20
LDIW		eq2*	eq2*	eq2*	eq2*	eq2*	1,494	1,494	1,494	1,494	1,494	1,494	1,494	1,494	1,494	1,494	2	2	2	2	2	20	20	20	20	20
PSO-G/PSO-VG/PSO-PG		0,729	0,729	0,729	0,729	0,729	1,494	1,494	1,494	1,494	1,494	1,494	1,494	1,494	1,494	1,494	max-pxcor	max-pxcor	max-pxcor	max-pxcor	max-pxcor	20	20	20	20	20
		eq1* -> w= 0.5 + rand()/2					eq2* -> www_end+(w_start - w_end)(1 - t/t_max)																			
		ITERATIONS					D					GOAL					CONCURRENT					WITH-R				
		F_0	F_1	F_2	F_3	F_6	F_0	F_1	F_2	F_3	F_6	F_0	F_1	F_2	F_3	F_6	F_0	F_1	F_2	F_3	F_6	F_0	F_1	F_2	F_3	F_6
STATIC PSO	user	10000	10000	10000	10000	10000	30	30	30	30	2	0,01	100	100	0,1	0,0001	on	on	on	on	on	off	off	off	off	off
	basic	10000	10000	10000	10000	10000	30	30	30	30	2	0,01	100	100	0,1	0,0001	on	on	on	on	on	off	off	off	off	off
	clerc	10000	10000	10000	10000	10000	30	30	30	30	2	0,01	100	100	0,1	0,0001	on	on	on	on	on	off	off	off	off	off
	telea	10000	10000	10000	10000	10000	30	30	30	30	2	0,01	100	100	0,1	0,0001	on	on	on	on	on	off	off	off	off	off
WUTHOUT INERTIA		10000	10000	10000	10000	10000	30	30	30	30	2	0,01	100	100	0,1	0,0001	on	on	on	on	on	off	off	off	off	off
RANDIW		10000	10000	10000	10000	10000	30	30	30	30	2	0,01	100	100	0,1	0,0001	on	on	on	on	on	off	off	off	off	off
LDIW		10000	10000	10000	10000	10000	30	30	30	30	2	0,01	100	100	0,1	0,0001	on	on	on	on	on	off	off	off	off	off
PSO-V/PSO...		10000	10000	10000	10000	10000	30	30	30	30	2	0,01	100	100	0,1	0,0001	on	on	on	on	on	off	off	off	off	off

Cuadro 5.1: valores por defecto de la aplicación

NOTA: En la versión Applet debemos pulsar “intro” cuando introducimos valores en los elementos de entrada de la sección “Experiment Settings”

5.4 Run

Para ejecutar el experimento podemos usar 2 botones: “Run” y “1 Step”.

Si pulsamos el botón “Run” se ejecutarán todas las iteraciones del experimento hasta que el criterio de parada sea alcanzado y ponga fin a la ejecución.

Si pulsamos “1 Step” solo se ejecutará una iteración del experimento cada vez que pulsemos este botón. Podemos volver a repetir el mismo experimento pulsando setup y de nuevo Run o también podemos cambiar la configuración del experimento en “Experiment Setting” y después pulsar Setup y run.

Nota: La sección Change View permite cambiar la forma y tamaño de las partículas PSO que visualizamos en el mundo en cualquier momento

5.5 Ver resultados

En el elemento “el mundo” veremos durante la ejecución el vuelo de las partículas PSO en tiempo real. En la sección “Result” veremos, también en tiempo real, como cambian de valor los parámetros de interés para el experimento.