

Taller de Modelos y Métodos Cuantitativos

Instrucciones Laboratorio Sintonización Automática

Leslie Pérez Cáceres

26 de octubre de 2010

1. Indicaciones generales

El objetivo de este laboratorio es el aprendizaje en el uso de ciertos algoritmos de sintonización automática, de modo que se puedan aplicar al proyecto desarrollado en el transcurso de este taller.

Se dispone de dos algoritmos de prueba a sintonizar: AIS para TTP y PSO para NSP. La descripción y ejecutables de éstos se encuentra disponible en el sitio web del curso.

2. REVAC

En la página del curso encontrará un archivo comprimido con nombre `revac.zip`, el cual se utilizará para este laboratorio. En este encontrará:

- Fuentes REVAC:
 - `Main.c`: Procedimiento central de REVAC
 - `includes.h`: Aquí se setean las iteraciones de REVAC (`MAX_ITER`), tamaño población (`M`), cantidad de padres (`N`), intervalo de mutación (`H`).
 - `globales.h`
- `configuracion.config`: Descripción de parámetros a sintonizar.
- `ejecutable_algoritmo.sh`: Rutina de ejecución del algoritmo a sintonizar.
- directorio `instancias`: Se deben dejar en éste las instancias a sintonizar.
- `Makefile`
- `ToDo.sh`: Rutina para ejecutar REVAC (Opcional)

2.1. Características ejecutable algoritmo

EL algoritmo para ser ejecutado por REVAC debe tener la siguientes características:

- Venir en un ejecutable, el cual reciba por parámetros: la instancia con la cual se ejecutará, los parámetros a sintonizar y el nombre del archivo de salida. Ejemplo:
`ejecutable #iteraciones #particulas factor_social factor_cognitivo inercia archivo_salida.txt`
- El resultado a optimizar (fitness final) del algoritmo debe ser entregado en un archivo de texto que lleve por nombre el indicado en la ejecución, en donde la primera línea tenga el mejor valor encontrando.

En caso de necesitar ejecutar la calibración en varias instancias a la vez, el archivo de resultados debe ser incremental, por lo que mostrará en cada línea un resultado.

2.2. Preparando la ejecución de REVAC

2.2.1. Rango de parámetros

El archivo configuración.config describe los rango de los parámetros a sintonizar. Ej.

```
2
TASA_CL 0 100 100
TASA_REEM 0 100 100
```

La primera línea indica cuantos parámetros se sintonizarán, en este caso 2. Luego por cada uno de estos parámetros se indica el nombre, rango de valores (0 100) y la segmentación de este rango (100). En este caso indica que los valores irán desde 0.0 hasta 1.0, con precisión de un decimal (definida directamente en el código de REVAC para la ejecución y en base al rango para la generación).

2.2.2. Preparación de la ejecución

En el archivo ejecutable_algoritmo.sh se realiza la ejecución del algoritmo a sintonizar, por lo que ella debe contener la ejecución del algoritmo en base a los parámetros que REVAC indique (lectura al inicio del procedimiento). Finalmente se realiza la ejecución que sea indicada por REVAC.

Se deben dejar en la carpeta instancias la(s) instancia(s) para la cual se sintonizará el algoritmo.

2.3. Ejecución

Una vez definida la búsqueda a realizar se procede a ejecutar REVAC:

- Ejecutar `make REVAC`
- Se puede ejecutar con el siguiente comando `./REVAC ejecutable_algoritmo.sh configuracion.conf instancias [semilla]`
- O con la siguiente línea `./ToDo.sh`

2.4. Resultados

Los resultados de REVAC son entregados por pantalla (archivo Salida-REVAC.out si se usa ToDo.h), en ellos se puede ver al final de cada iteración un resumen de cada parámetro (nombre_parametro media rango).

3. ParamILS

El código completo de este algoritmo se puede descargar en <http://www.cs.ubc.ca/labs/beta/Projects/ParamILS/>, se encuentra programado en el lenguaje Ruby y hay una guía de uso disponible en <http://www.cs.ubc.ca/labs/beta/Projects/ParamILS/ParamILS2.3-Quickstart.pdf>, me diante la cual se puede aprender a adaptar el algoritmo a nuestra medida. En la página del curso encontrará un archivo comprimido con nombre paramils.zip, el cual se utilizará para este laboratorio. En este encontrará:

- paramils (ejecutable)

- carpeta files
 - carpeta instancias
 - carpeta salida
 - archivo instancia.cnf: Contiene ruta a la instancia a sintonizar.
 - archivo escenario-instancia.scn: Contiene información para la ejecución de la instancia a sintonizar.
 - params.params: Archivo que describe el set de valores de un parámetro y su valor inicial.
 - sh-algoritmo.sh: Rutina de ejecución del algoritmo a sintonizar.
- ToDo.sh: Rutina para ejecutar ParamILS

3.1. Características ejecutable algoritmo

EL algoritmo para ser ejecutado por ParamILS debe tener la siguientes características:

- Venir en un ejecutable, el cual reciba por parámetros: la instancia con la cual se ejecutará, los parámetros a sintonizar y el nombre del archivo de salida. Ejemplo:
ejecutable #iteraciones #particulas factor_social factor_cognitivo inercia archivo_salida.txt
- El resultado a optimizar (fitness final) del algoritmo debe ser entregado en una archivo de texto que lleve por nombre el indicado en la ejecución, en donde la primera línea tenga el mejor valor encontrando.

3.2. Preparando la ejecución de ParamILS

3.2.1. Rango de parámetros

El archivo files/params.params describe los set de valores de los parámetros y su valor inicial. Ej.

```
TASA_CL {0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0} [0.5]
FACT_CL {0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0} [0.5]
```

Los valores en son los valores posibles para el parámetro y al final se adjunta su configuración inicial, que en este caso es 0.5 para ambos.

3.2.2. Preparación de la ejecución

Cada una de las instancias a utilizar en la sintonización deben ubicarse en la carpeta instancias. Se define un archivo del tipo instancia.cnf para mostrar la ruta a las instancias que participarán en la sintonización. Puede ser solo una o muchas mas, además se pueden definir dos sets de instancias, uno de entrenamiento y otro de ejecución (para mas información ver el documento oficial).

Por cada instancia o grupo de ellas a ejecutar se define un archivo del tipo escenario-instancia.scn, el cual entrega a ParamILS las características de la búsqueda que deseamos que haga. Los elementos que se encontrarán en este archivo son:

- algo: señala la rutina de ejecución del algoritmo a sintonizar.
- execdir: directorio en el cual se realiza la ejecución

- **deterministic**: determina como realizar la búsqueda 0 para randomizada, 1 para determinista.
- **run_obj**: determina el objetivo a optimizar en cada corrida del algoritmo a sintonizar. ParamILS trae varios ya incorporados como runtime para el tiempo de ejecución, runlength para que tan larga es la búsqueda, entre otros.
- **overall_obj**: determina como son comparados dos resultados obtenidos con el criterio anterior (run_obj). Algunos de estos procedimientos incluyen mean (media), median (mediana), entre otros.
- **cutoff_time**: señala el tiempo máximo de ejecución de una corrida del algoritmo a sintonizar.
- **cutoff_length**: señala el largo máximo de la búsqueda (ej. cantidad de nodos revisados, evaluaciones, etc.)
- **tunerTimeout**: tiempo máximo de ejecución del sintonizador.
- **paramfile**: archivo de descripción de los parámetros.
- **outdir**: directorio donde se escribirán los resultados.
- **instance_file**: archivo que contiene las rutas a las instancias de prueba de la sintonización.
- **test_instance_file**: archivo que contiene las rutas a las instancias de entrenamiento de la sintonización.

En el archivo `sh_algoritmo.sh` se realiza la ejecución del algoritmo a sintonizar, por lo que ella debe contener la ejecución del algoritmo en base a los parámetros que ParamILS indique (lectura al inicio del procedimiento). Finalmente se realiza la ejecución que sea indicada.

3.3. Ejecución

Una vez definida la búsqueda a realizar se procede a ejecutar ParamILS:

- Ejecutar `./ToDo.sh`
- O se puede ejecutar con el siguiente comando:
`./paramils numRun <#corridas> -maxIts <#iteraciones> -maxEvals <#evaluaciones> -approach
<tipo:basic,focused,random> -N <#numero de runs para evaluar conf. de param> -userunlog <1
| 0> -scenariofile <archivo.scn>`

3.4. Resultados

Los resultados de ParamILS se pueden encontrar en el directorio de salida, los resultados finales se encuentran en el archivo `-result`. Además ParamILS provee varios archivos que muestran en el estado de la búsqueda paso a paso (para mas información leer el documento oficial)