Synapse Optimizer documentation

Riccardo Loti

Version 1.1 (20120304)

Chapter 1

Preparation

1.1 Source compilation

Source files are accompanied by a Makefile so that is possible to use *make* to compile:

```
make
```

Once compiled the binary, called *synopt*, is found under the *bin/* subdirectory.
The makefile also con taints targets for cleaning and installing the binary:

```
make clean
make install
```

1.2 Installation

The makefile *install* directive for now only copies the binary in the source's parent directory (*../source/*):

```
make install
```

Yet it is possible to both modify the *cp* command in the Makefile file or to manually copy the executables to specific locations, like */usr/local/bin* with command *mv*:

```
mv bin/synopt <destination_dir>
```

The binary executable can anyway be run from any location as long as it is marked as executable.

Chapter 2

Usage

2.1 Passed parameters

Optimizer needs some command line parameters, while others can be optionally defined or left to default values.
Syntax is:

```
./synopt -d <degree_file> -w <result_file> [<parameter_tag> <parameter_value>]
```

Where mandatory option are:

- **-d** or –**degree** specifying the path for the degree distribution file, expressed as a list of couples of values where the second is the probability of having the number of neighbors specified in the first values and such probabilities are normalized to one (see "File format definition" document for details on the file internals).
- **-w** or –**write_file** specifying the path for the produced result file (see Chapter 3 for details and format).

And optional parameters tags and meanings are:

- **-o** or –**overlays** is the number of interconnected overlays which also represents the maximal degree of synapse nodes. The default value is 10.
- **-t** or –**ttl** is the TTL (Time To Live) of the sent queries, meaning the number of jumps after which each packet is discarded to avoid

traffic and too late replies. It is to be noticed that having an infinite statistical model in the optimizer, as opposed to a finite simulative or real implementation, a TTL over a certain limit shows results that, while correct, tends to deviate from real life scenario, where the model expand its explored horizon and the real life implementation covers repeatedly the graph, without generating any new information and only adding traffic. For networks in the order of a million nodes a TTL of 4 or less is strongly suggested for such reason, also the default value is 3.

- **-a** or −**alpha** is the α of the requested resource, meaning the probability that a visited node owns it. Also intuitively represents the distribution density of the resource, where for example an α of 0.001 means that on average a single node every 1000 has the requested resource. The default value is 0.0001.

- **-x** or −**dx** is the dx for the numerical calculation of the prime derivative, default value is $10^{-}9$ and the parameter is passable also as scientific notation format (eg.: 1e-9).

- **-g** or −**granularity** TODO default value 20

- **-r** or −**routing** TODO. Currently accepts only two values, 0 for TODO, which is also the default value, and 1 for TODO.

- **-v** or −**verbose** is a flag setting the level of console messages for information and debug purposes. Currently unused and ignored if passed.

gran_dist indica la granularit con il quale esplorare lo spazio combinatorio. Non un parametro del grafo o della rete in se, ma utilizzato nel generare le varie combinazioni tra gli overlay che verranno esplorate. Un valore elevato velocizza l'esecuzione dell'ottimizzatore, ma rischia di perdere di precisione nel rappresentare i comportamenti del sistema, mentre al contrario un valore minore migliora il campionamento dei possibili stati del sistema, ma rallenta l'esecuzione dell'ottimizzatore. Il valore consigliato di 20, eventualmente scendendo da esso se si sospetta il sistema oscilli notevolmente.

Chapter 3

Returned results

3.1 Result file

Il file prodotto da ogni esecuzione di SynOptimizer formato da pi tuple, ognuna composta da k + 3 valori, dove k il numero di overlay della rete (specificato in fase di esecuzione come parametro n_overlay) e i primi k-esimi valori indicano la percentuale normalizzata a 1 di nodi appartenenti a quel numero di overlay (1, 2, ..., k) per la permutazione del sistema indicata dalla tupla.
I restanti tre valori indicano rispettivamente:
T il "costo" delle query, inteso come il numero di messaggi inviati.
p_{hit} cio la probabilit di successo di una query.
F il risultato di funzione di costo che tenuto conto sia di T che di p_{hit} stabilisce un equilibrio (secondo coefficienti arbitrari interni al codice) tra efficienza della rete e relativo costo funzionale.