Homework 5
K-way Graph Partitioning Using JaBeJa

Denys Tykhoglo

Solution

As the homework description suggests, I used the scaffolding source code for JaBeJa simulation provided at https://github.com/smkniazi/id2222. The solution consists of 2 tasks:

1. The first task deals with filling in the missing code to methods *sampleAndSwap(...)* and *findPartner(...).* The inserted java code corresponds to the pseudocode from the JaBEJa paper.
2. For the second task I implemented a simulated annealing algorithm described in http://katrinaeg.com/simulated-annealing.html, which I refer as *enhanced SA* in further text. Apart from that, I implemented optional restart for simulated annealing.

I added several new command options to the solution:

- *-enhanced* – turn on usage of enhanced SA;
- *-temp_enh* – initial temperature for enhanced SA;
- *-min_temp_enh* - minimal temperature for enhanced SA;
- *-alpha_enh* – alpha factor for enhanced SA;
- *-iter_enh* - number of iterations per single temperature for enhanced SA;
- *-restart* – turn on usage of SA restarts;
- *-restart_interval* – restart SA after T reaches its min. value after this number of rounds.

Dataset

For this homework, I used graphs *3elt*, *add20* and *Twitter* that are provided in form of *.graph* text files in the homework directory.

How to run

In order to run the solution, it is necessary to run scripts *compile.sh* and *run.sh*. The solution can be run with several parameters, all of which can be displayed and described with an option *-help*. After that, the solution can be shown in form of a graph using script *plot.sh* in case if *Gnuplot* is installed.

As I am using operational system Windows, I translated script *plot.sh* into *plot.bat* so that I can plot executions solutions on my computer.

Results

I came up with several parameter sets, which do not provide the best possible results for all graphs but are still useful for demonstration. They are displayed in *Figure1*.

All executions are performed with parameter *seed = 0*. In order to get more adequate results, it is necessary to take average values after using several different seeds as sources of randomness.

I limit the number of rounds for each *run* to *1000*. If more rounds are allowed, sometimes results can become better considerably. However, it usually does not yield a very big improvement, though it requires more tome to test, especially for large graphs.

In addition, I use parameters *alpha = 2*, *node selection policy = HYBRID* and *Initial temperature for enhanced SA = 1* for all runs as they proved to be good for all graphs. Each execution tries to maximize *number of partitions* equal to *4*.

Parameter set number	ICSP	URSS	RNSS	Delta	Min. temperature (SA enh.)	Alpha (SA enh.)	Iterations per temperature (SA enh.)	Restart (in rounds)
1 (Initial params)	Round robin	6	3	0.003				
2	Batch	7	1	0.02				
3	Batch	7	1	0.02				200
4	Batch	7	1		0.01	0.85	15	
5	Batch	7	1		0.15	0.8	15	200

Figure 1: Parameter sets

Figure 2 shows the execution results for each parameter set – graph combination. The numbers in red are the worst results for a given graph. Correspondingly, the numbers in green are the best or almost the best results.

Parameter set number	Edge cut	Swaps	Migrations
3elt graph			
1 (Initial params)	2604	125651	3328
2	377	26511	254
3	345	95401	333
4	345	58818	246
5	347	45741	229
add20 graph			
1 (Initial params)	2095	599407	1751
2	1619	22456	1504
3	1467	61418	1558
4	1701	66221	1482
5	1744	72666	1469
twitter graph			
1 (Initial params)	41112	199264	2050
2	41305	12525	1312
3	41251	33239	1332
4	41333	4296	1302
5	41388	4622	1289

Figure 2: Results for several parameter sets and graphs

For *3elt.graph* we can clearly see a big impact of changing the initial parameters. One of the biggest advancements for it was changing the RNSS value from 3 to 1. For this graph, enhanced SA shows a slight improvement in terms of edge cut and a good improvement in terms of the number of swaps when restarts are used.

For *add20.graph* the enhanced SA shows a worse result comparing to the standard SA. Moreover, the enhanced SA results for the number of swaps is worse as well.

In case of *twitter.graph*, by changing the initial parameters it was only possible to improve the number of swaps but not the edge cut. The enhanced SA in this case showed same or a little worse edge cut, but a considerably better number of swaps.

One common conclusion that we notice is that in most cases restarts are useful for decreasing edge cut but they obviously require more swaps. As for the enhanced SA, it can be quite useful for some graphs as *3elt* and *twitter* (for the latter it can reduce the number of swaps) but completely lose to the standard SA as for *add20*. However, I would not exclude an option that enhanced SA can have a much better performance for *add20* if other parameters are tuned properly.

The plots for all executions are attached with this document in directory *plots*. The plots filenames are in format *<graph name>-<number of parameter set>.png*.

On the plots we see that generally the convergence time is bigger when using enhanced SA mainly because of the chosen parameters. For example, for the enhanced SA we use the same temperature for several rounds (in our case, 15) before decreasing the temperature.