DARPA Meeting
Second order methods and reinitialization

Cheng-Han Huang

2025.9.24

Methods for Solving Relaxed LP for MIS

- pCQO: Baseline for solving a relaxed continuous optimization problem with MGD and a new MIS checker to replace traditional tolerance termination condition.

What we are exploring:

- Newton: Follows the structure of pCQO but instead uses Newton to approach the relaxed optimization problem.
- L-BFGS: Follows the structure of pCQO but instead uses Scipy’s L-BFGS-B to approach the relaxed optimization problem.

Newton and L-BFGS-B

- 10 trials for each instance in $ER(500, 0.5, \textit{seed 0 to 49})$.

- The solution size of a seed is the best solution size of the 10 trials, and the average number of iteration and time for all trials.

- Newton and L-BFGS-B both show improvement on finding larger solution over baseline MGD but take more time to solve. In particular, L-BFGS-B > Newton > MGD, where MGD is the choice of solver in pCQO, the baseline method for solving a relaxed LP.

Baseline MGD with and without Third Term

- Comparison by solution size:

	MGD without Third Term	MGD with Third Term
Average solution size	10.8	10.8

- Comparison by iteration number/time:

	MGD without Third Term	MGD with Third Term
Average iteration number	85.6	86.3
Average time	0.0196	0.0260

Newton and L-BFGS-B

- Comparison by solution size:

L-BFGS-B > Newton	Tie	L-BFGS-B < Newton
13	32	5

	L-BFGS-B	Newton
Average solution size	11.26	10.96

- Comparison by iteration number/time:

	L-BFGS-B	Newton
Average iteration number	167.1	76.3
Average time	0.7411	0.1350

Methods Comparison

- Set fixed time limits for the following five methods

- Baselines: pCQO, CP-SAT, Gurobi
- Attempts: Newton, L-BFGS

On 10 instances for each of
$ER(500,0.5), ER(1000,0.5), and\ ER(2000,0.5)$

Description for Each Method

- CP-SAT: Google's open-source solver that turns a problem into Boolean SAT with CP propagators and largest neighborhood search

- Gurobi: SOTA commercial solver that uses simplex method or interior method to branch-and-cut on LP relaxations.

- Both methods are CPU-based

Description for Each Method

- pCQO: Baseline for solving a relaxed continuous optimization problem with MGD and a new MIS checker to replace traditional tolerance termination condition.

- Newton: Follows the structure of pCQO but instead uses Newton to approach the relaxed optimization problem.

- L-BFGS: Follows the structure of pCQO but instead uses Scipy’s L-BFGS-B to approach the relaxed optimization problem.

Comparison with CPU Based Baselines

Graph, Time	MGD	CP-SAT	Gurobi	Newton	L-BFGS
ER(500,0.5), 5 sec	12	12	8.7	11.2 (11.2)	11.1 (11.1)
ER(1000,0.5), 5 sec	11.5	0	9.5	11.8 (9.3)	11.2 (11.2)
ER(1000,0.5), 10 sec	11.8	11.4	9.5	12.2 (9.5)	11.6 (11.4)
ER(2000,0.5), 10 sec	10.8	0	10.4	10.5 (9.7)	11.8 (11.8)
ER(2000,0.5), 20 sec	12.1	0	10.4	10.8 (9.7)	11.8 (11.8)

- Baselines
- Preliminary, brackets indicate without using the clique informed penalty

Note that both integer programming methods uses thread-based parallelization, but the three gradient-based methods do not. The code is run on my local computer for initial results and is also able to run on advanced computational systems for future reports.

Maximum iteration for the three gradient based methods are set to 150 across all runs, leading to possible failing cases which report 0 that drags down the average solution size in for larger graphs.

Reinitialization

- The original degree-based initialization before randomizing is
  $d_{\text{init},i} = \frac{\tilde{d}_i}{\max_k \tilde{d}_k}$ where $\tilde{d}_i = 1 - \frac{\deg(i)}{\max_j \deg(j)}$

- We propose to used instead $\frac{1}{1+\lambda p_j^{(i)}} d_{init_j}$ where $p_j^{(i)} = \begin{cases} 0, i = 0 \\ \frac{A_j^{(i)}}{i}, i > 0 \end{cases}$

  where $A_j^{(i)}$ denotes how many times node j is chosen in the first i trials.

Reinitialization

- Initial experiments show that $\lambda = 5$ is an acceptable choice, and that denser graphs are more likely to benefit from this technique.

Next, we consider nonconstant schedules for λ.

- Experiments are run on $ER([200, 400, 600, 800], 0.5, \textit{seed 0 to 10})$. Each instance is given 50 trials, and the solution size reported is the best among the 50 trials.

Reinitialization

- Different choices of lambda schedules. $C = 3$.

	mean_best_MIS
linear	11.500
fast_plateau	11.700
jump	11.675
exp_decay	11.625
quad_convex	11.450
quad_concave	11.575
cyclic	11.650

Fast drop + plateau

$$\lambda(i) = \begin{cases} 15 - \frac{10}{9}\, i, & i < 10, \\ 5, & i \geq 10. \end{cases}$$

Jump schedule

Let $b = \lfloor i/5 \rfloor$.

$$\lambda(i) = \begin{cases} 10 - b, & i \text{ even}, \\ b, & i \text{ odd}. \end{cases}$$

Exponential decay toward 5

Define $t = \frac{i}{N-1}$.

$$\lambda(i) = 5 + (15 - 5)\, e^{-5\,t}.$$

Quadratic (convex) drop

With $t = \frac{i}{N-1}$:

$$\lambda(i) = 15 - 10\, t^2.$$

Quadratic (concave) drop

With $t = \frac{i}{N-1}$:

$$\lambda(i) = 5 + 10\big[1 - (1 - t)^2\big].$$

Cyclical (sine-wave)

With $t = \frac{i}{N-1}$ and C cycles:

$$\lambda(i) = 5 + 10\, \frac{1 + \sin(2\pi C\, t)}{2}.$$

For next meeting

- Newton and L-BFGS refine for time issue
- How do we benefit from thread parallelization (Newton and L-BFGS)
- Run on
- Maybe some time later: Comparison with GPU Based Baselines (CuOPT) and iSCO