MASSIVELY PARALLEL STOCHASTIC LOCAL SEARCH FOR THE GRAPH COLORING PROBLEM

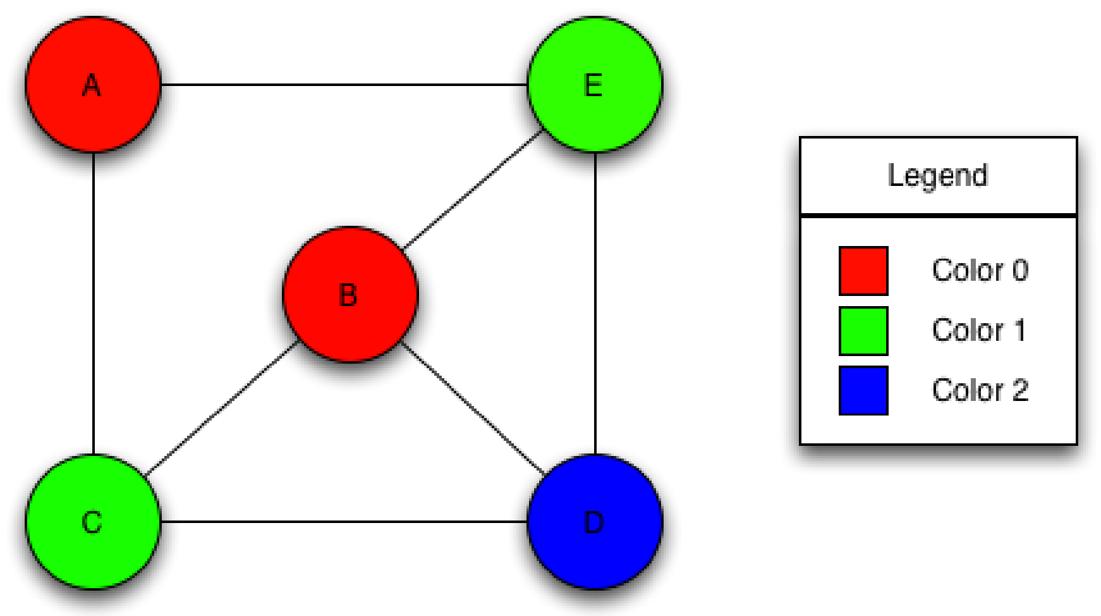
Junan Zhao <jz2723@rit.edu>   Advisor: Prof. Alan Kaminsky <ark@cs.rit.edu>

Objective

To propose a new parallel heuristic search algorithm to solve a classical NP combinatorial optimization problem—the graph coloring problem. The algorithm is expected to be faster than exact search algorithms.

Graph Coloring Problem

Given an undirected graph G = (V, E), each vertex should be assigned a color. On the constraint that adjacent vertices have to be colored differently, what's the minimum number of colors required to cover the graph?

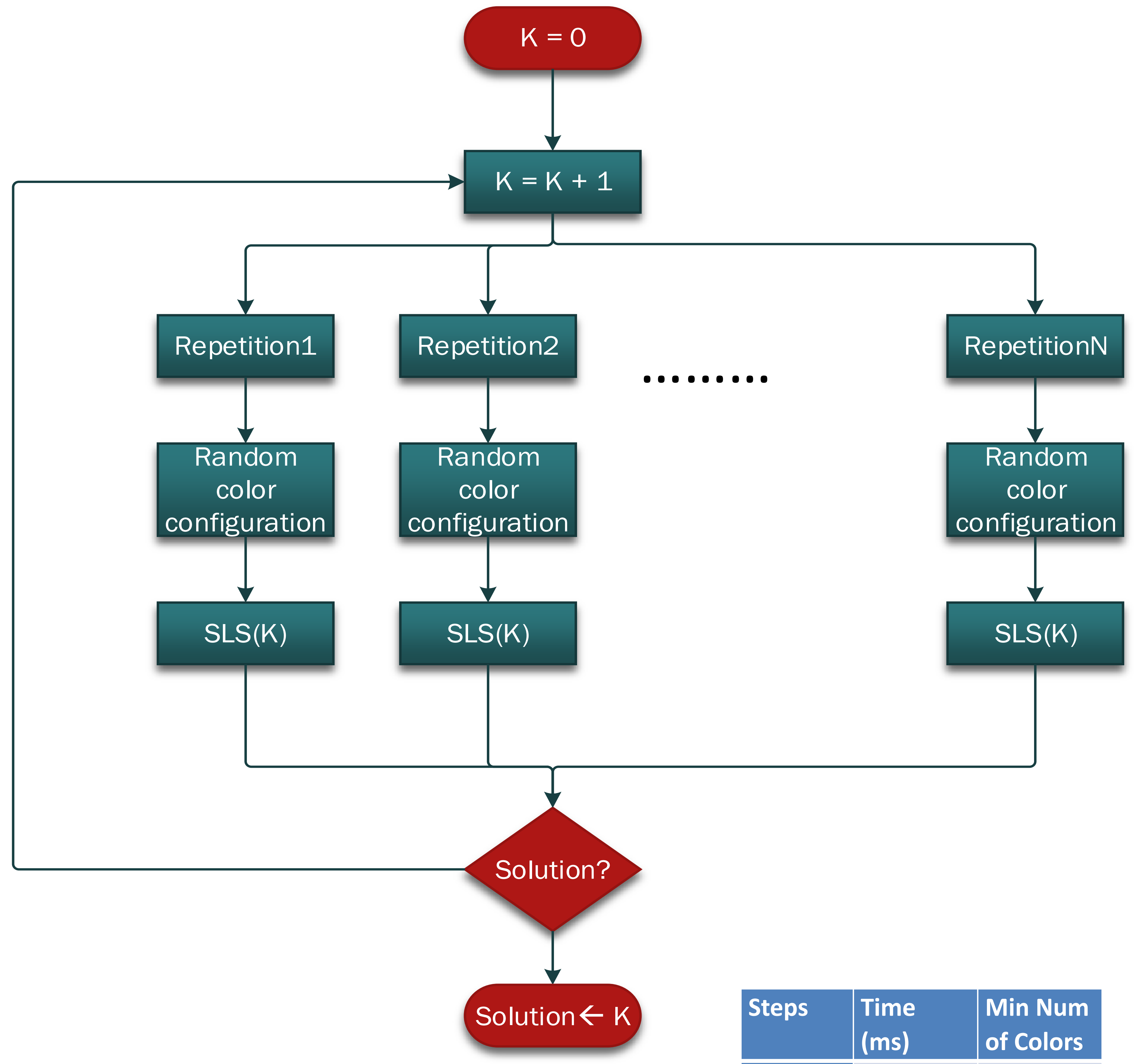


Stochastic Local Search (SLS)

SLS(K)

1. Given a color configuration with K colors, do the following 2-4 Steps S times.
2. Check the current color configuration. If there are no conflicts, stop the algorithm and report K; else continue.
3. Pick a vertex V with color conflicts randomly.
4. **If** there is a color C within K which makes no conflict for V, assign C to V;
   else generate a random fraction between 0 and 1. Compare the fraction with a constant P. **If** P is larger, assign a random color within K to V;
   else assign the color which makes least conflicts for V.
   Finally back to Step 2.
   (For all situations in Step 4, if there are multiple color choices, pick a random one of them)

Massively Parallel Stochastic Local Search



Results

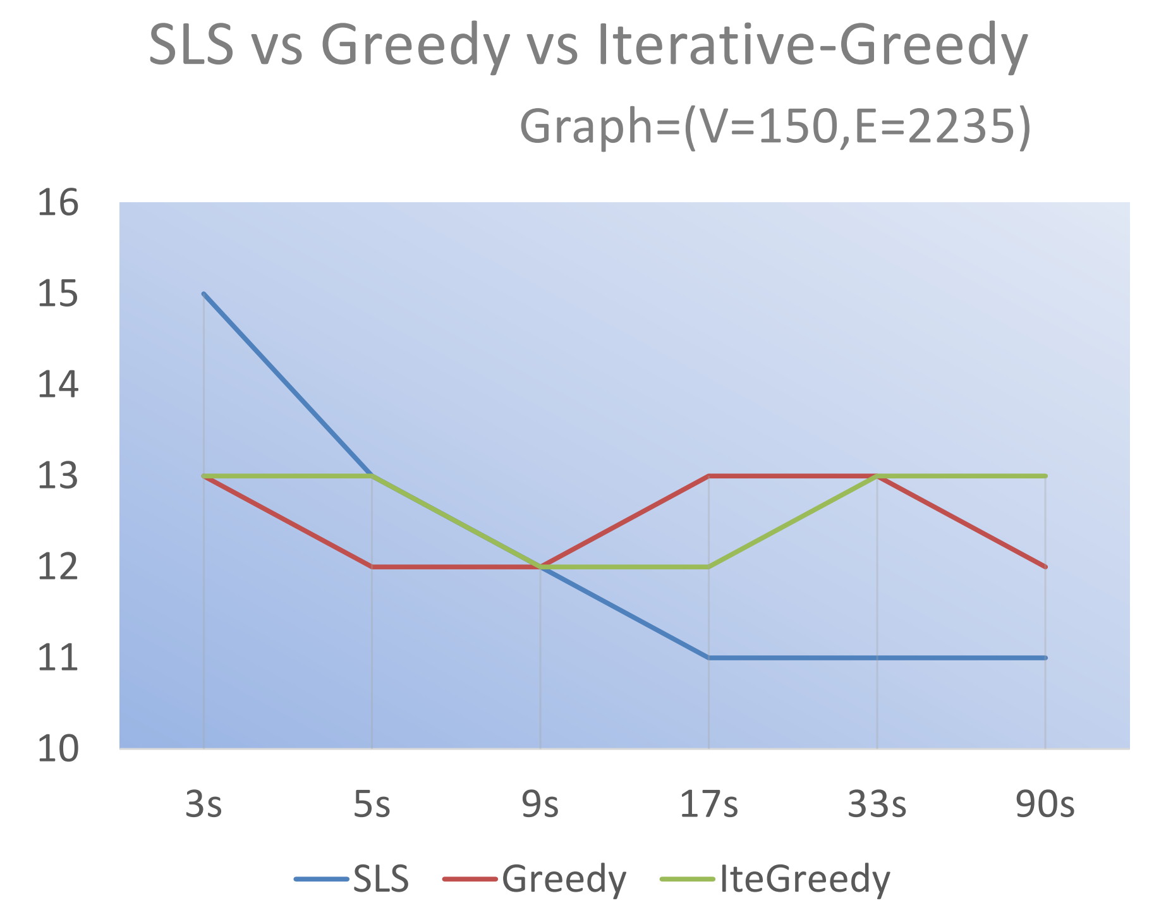
Time (ms)	Graph1	Graph2	Graph3	Graph4	Graph5
Exhaustive	126831	34854	N/A	N/A	N/A
Backtracking	205	212	467164	132888	878331
MPSLS	296	329	559	653	2193

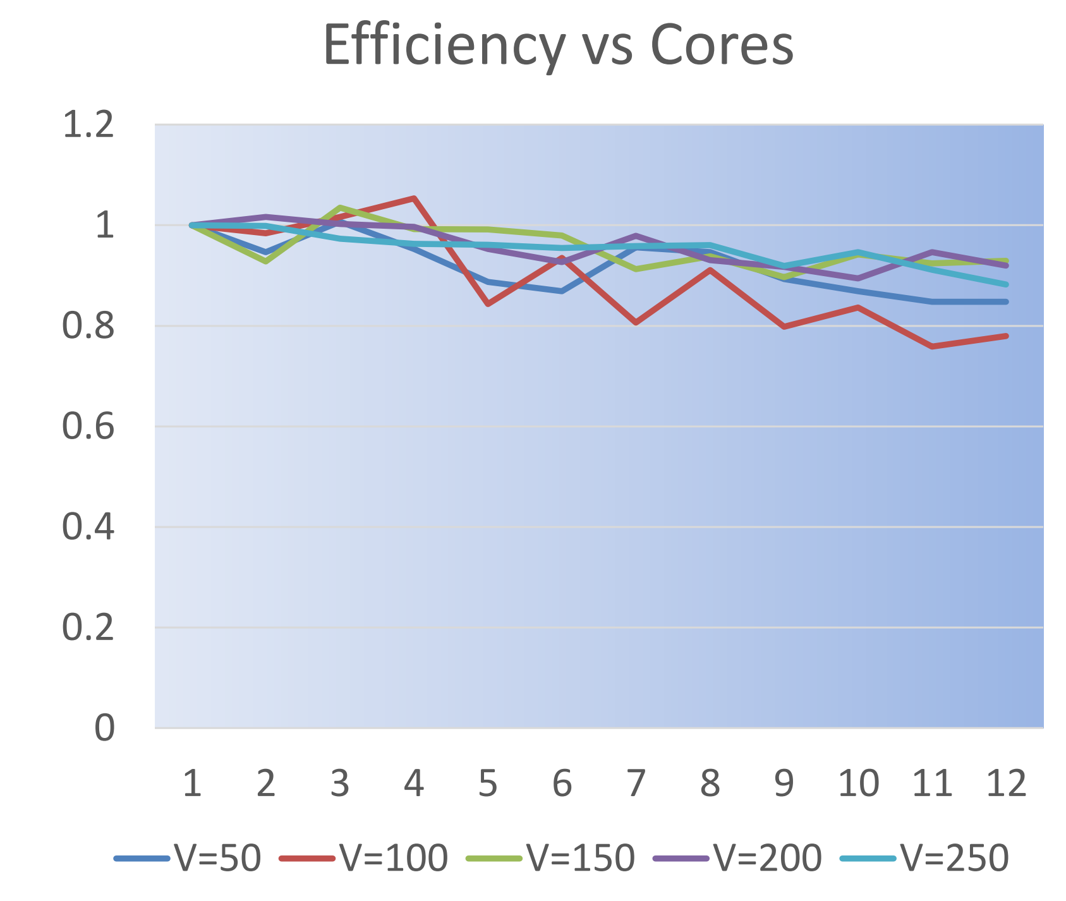
Graph1=(V=12,E=60), Graph2=(V=20,E=65), Graph3=(V=35,E=500), Graph4=(V=45,E=400), Graph5=(V=60,E=400)

Steps	Time (ms)	Min Num of Colors
100	13455	32
200	15823	20
500	35605	18
1000	64409	17
2000	128985	17
5000	300344	16
10000	607300	16
20000	1257179	16

Graph=(V=250, E=5976)

The left table shows its runtime comparison with exact algorithms on small/medium graphs. With the same optimum solutions (num of colors) found, the MPSLS ran at least 100 times faster than parallel exhaustive search on small graphs, 200 times faster than parallel backtracking on medium graphs. The right table shows its ability of approaching optimum solutions on large graphs: with given more runtime, the MPSLS algorithm will gradually approach/reach the optimum solution.





The left figure presents its comparison with inexact algorithms on large graphs. When the runtime is long enough (>10s), the MPSLS usually got better solutions than the other two algorithms. The right figure reveals its efficiency as a parallel algorithm. With cores increasing from 1 to 12, it's always efficient better than 78%.

Future Work

- To change the initial configuration algorithm, instead of random color configuration
- To study the interference of different P values
- To compare it with more other inexact algorithms

Conclusion

The MPSLS algorithm

- is able to get the optimum solutions
- much faster than exact search algorithms
- with runtime long enough (>10s), it can usually get better solutions than parallel greedy and parallel iterative greedy algorithms
- has a good weak scaling performance

References

1. R. M. R. Lewis, *A Guide to Graph Colouring: Algorithms and Applications*. 2016.
2. A. Kaminsky, *Big CPU, Big Data: Solving the Worlds Toughest Computational Problems with Parallel Computing*. 2nd ed., pages 197-216, Barnes & Noble Press, 2019.