9th International Conference on Structural Health Monitoring of Intelligent Infrastructure
2019 ST. LOUIS SHMII-9
AUGUST 4-7, 2019 | St. Louis, Missouri USA
Organized by MISSOURI S&T

A Genetic Algorithm for MinMax k-Chinese Postman Problem with Applications to Bridge Inspection

S. Liu[1], S. Louis [2], N. Harris [2], H. La[2]
[1] *Missouri State University – USA, Email:simingliu@missouristate.edu*
[2] *University of Nevada, Reno - USA,*

Abstract

Bridge inspections ensure transportation infrastructure safety and save lives but current manual bridge inspections can be slow and costly. Automated bridge inspection research mitigates these problems and we investigate a multirobot approach to automated bridge inspection by mapping the inspection problem to the well known k-Chinese Postman problem and using multiple robots for faster, more cost-effective, and more standardized bridge inspections. We first show that a genetic algorithm quickly approaches the optimal solution to the 1-postman problem corresponding to a single robot inspecting a bridge. Then, we use the same genetic algorithm to efficiently solve the more difficult, multi-robot, NP-hard, MinMax k-postman problem (k > 1). The genetic algorithm's solutions to this problem represent robot paths that traverse (and thus inspect) every truss at least once and that minimize the length of the longest path traversed by any of the k robots - thus minimizing time and distributing the workload. These simulation results from our immersive bridge inspection simulation and training system built with the Unity3D game engine, show that our genetic algorithm quickly and efficiently produces good paths, and in addition, achieves approximately linear speedup for each robot added to the inspection task.

1. Introduction

According to ACSE's 2017 Infrastructure Report Card, the U.S. has over 614,387 bridges, forty percent of which are over 50 years old, and nine percent of which are classified as structurally deficient. Additionally, while smaller bridges can be inspected visually, larger bridges require cranes, boats, or other equipment to do a thorough inspection. This type of inspection consumes a large portion of states' budgets that could be significantly reduced by automating the inspection process. Current generation inspection robots require an operator to control their movements. The next generation of more autonomous robots will do much of the inspection without operator control and with only occasional need for operators to intervene. Human operators will manage (not tele-operate) five to ten autonomous robots at a time, intervening only to fix occasional errors. Thus, we will only need to train and use one human operator to oversee and manage multiple robots that cooperatively and simultaneously inspect the bridge more quickly, more effectively, and at significantly less cost. The problem then becomes one of finding paths for every robot such that all bridge trusses are inspected at least once and that the workload is split equally among all robots to minimize inspection time. This problem maps well to the MinMax k-Chinese Postman Problem (MM k-CPP). This problem is known to be a difficult, NP-Hard problem to solve to optimality as described by Edmonds (1973). In other words, the space of possible solutions to the MM k-CPP is too large to exhaustively search for the optimal solution within a reasonable amount of time. We thus use a Genetic Algorithm (GA) to efficiently search this poorly-understood space and quickly find near-optimal solutions. Much empirical evidence shows that GAs are good at searching poorly-understood spaces and quickly finding optimal or near-optimal solutions. Our results bear this out and show that our GA finds near-optimal solutions within a few minutes and that these solutions scale well with the number of robots. We begin the process of solving the k-robot bridge inspection problem with GAs by representing the bridge truss as a graph, where nodes represent truss intersections and edges represent truss beams. This maps the k-robot bridge inspection problem to the MM k-CPP. The GA works with a population of individuals representing routes and starts by randomly generating k routes for each individual in the population. Each route is represented by a sequence of edges and we use Djikstra's optimal algorithm to find routes between non-adjacent edges. Shorter paths have higher fitness and are likely to be reused to generate better solutions. The GA seeks to maximize fitness by perturbing individuals using genetic operators and evolves better routes over many generations (iterations). Experimental results show that GAs obtain near-optimal, equal length routes. We get linear speedup in inspection time as we add more robots - five robots are approximately five times as fast as one robot. Using a handful of autonomous robots to inspect a bridge can lead to significant savings in time and in the number of operators (one) needed to manage the robots.

The remainder of this paper is organized as follows. Section II discusses related work in research on Chinese Postman Problems (CPP) and GAs. Section IV describes route representation, genetic operations, and route splitting used in our experiments. Section V presents preliminary results and compares the generated routes with the global optimal route obtained with the assumption of using one robot. Finally, the last section draws conclusions and discusses future work.

2. Related Work

The Chinese Postman Problem, defined by Mei-Ko Kwan in 1962 attempts to determine the most effective route for a postman to distribute the mail received from the post office and to return to the post office with the shortest walking distance during mail distribution by Eiselt (1995). Algorithms for traditional CPP problems have been well studied and the CPP on completely undirected graphs or completely directed graphs can be solved in polynomial time. Currently researchers are more interested in a variety of extensions of traditional CPP problems. Papadimitriou (1976) worked on a CPP based on a mixed graph with directed and undirected edges which turns the CPP for finding the least-cost route into an NP-hard problem. Minieka (1979) further extended the mix graph to a windy postman problem which assigns different costs for traveling an edge from different direction. Since the real world CPP problems can be very large, we can speed up total time needed to traverse all the edges by increasing the number of postmen or vehicles. The k-CPP was first introduced in Frederickson (1976) and it is NP-hard by a reduction from the k-partition problem. Comparing to the common objective of minimizing the total distance traveled by the k vehicles (k-CPP), MinMax k-Chinese postman problem (MM k-CPP) aims to to minimize the length of the longest of the k tours in order to balance the distance of each tour for k vehicles. Heuristic algorithms for the MM k-CPP were developed by Ahr and Reinelt (2002). A tabu search algorithm was also presented by Ahr and Reinelt (2006). Chen (2018) described a related MinMax Multiple-Depot Rural Postman Problem (MMMDRPP) and developed an efficient tabu-search-based algorithm and proposed three novel lower bounds to evaluate the routes. Gendreau and Hertz (1994) described a new tabu search heuristic for the vehicle routing problem with capacity and route length restrictions. Salhi and Sari (1997) proposed a multilevel composite heuristic to address the problem of simultaneously allocating customers to depots, finding the delivery routes and determining the vehicle fleet composition. GAs are a class of stochastic optimization algorithms that use the principles inspired from natural evolution for solving optimization problems. Sumichrast and Markham (1995) used the GA based method presented by Clarke and Wright (1964) for solving a problem where raw materials were transported from multiple depots to a set of plants. Thangiah and Salhi (2001) proposed a generalized clustering method based on a GA and applied a Genetic Clustering (GenClust) method for solving the multidepot vehicle routing problem.Wink and Back (2012) presented a Hybrid GA which incorporates problem-specific heuristics and domain knowledge into the algorithm for solving the Capacitated Vehicle Routing Problem. Choi and Seong (2003) presented a GA to solve the asymmetric traveling salesman problem. In this paper, we present a genetic algorithm for the MM k-CPP problem with unfixed depots.

3. Methodology

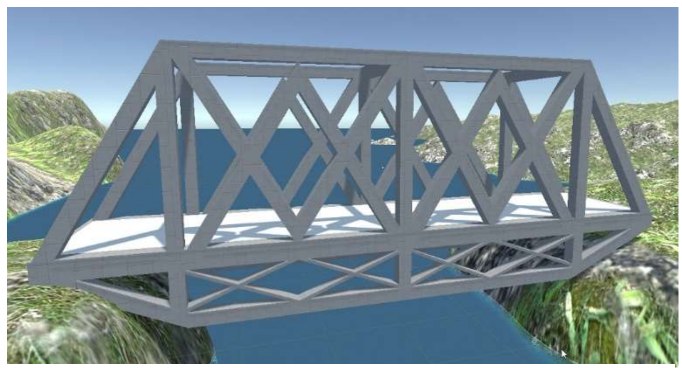

Figure 1. A bridge to be inspected.

In this section we introduce our method for route generation for multiple robots during a bridge inspection. To create a path for k > 1 robots such that each edge on a bridge is covered at least once (k-CPP) and the workload is evenly distributed is difficult for a human expert, especially as k grows. Since the problem is also NP-Hard, we cannot use exhaustive search to find the optimal solution and we need to use some kind of heuristic search method such as a GA. Any heuristic search method requires us to compute the cost of a solution so that we can search for the minimal cost solution. The next two subsections describe our simulation environment and our cost computation. We used Unity3D as our simulation engine for simulating and visualizing bridges and robots. Fig. 1 shows a visualization of a test bridge used in our experiments. Once a final solution has been generated for k robots, we can observe their route in our simulation as they traverse the bridge. Each robot will leave a unique colored trail behind it, so that each robots path can be easily identified and it can be seen that the bridge has been completely covered. Furthermore, a human operator can be easily trained to manage multiple robots as they traverse truss members within this simulation, watching inspection progress, and intervening if robots run into trouble. The next subsection describes how this simulated bridge maps to the more abstract MM k-CPP.

3.1 Translating a Bridge to a Graph

The problem of generating k inspection routes for k robots maps to the MM k-CPP where the set of edges, E, maps to the set of bridge members and the set of vertices, V, map to joints. Thus, given a graph G = (V, E) corresponding to the bridge under consideration, solving the MM k-CPP solves the k-robot routing problem for bridge inspection. Specifically, we begin by translating the bridge into an undirected weighted graph, where the beams of the bridge are the edges, the connecting areas or joints, the vertices. The weight of each edge simply represent the distance along that path. Fig. 2 shows the bridge in Fig. 1 converted into a graph as described. We use a distance matrix to represent the graph for our computation in GA by determining the shortest distance between any given two points to evaluate a possible solution. In order to solve the MM k-CPP problem we apply GA to generate possible solution paths and evolve solutions that minimize cost. GAs have been shown to find high-quality, near optimal solutions for problems with extensively large search spaces in polynomial time. Our results show that using the total length of all edges in our graph as the theoretical optimal solution, GAs can find the optimal path for one robot and consistently find a near optimal solution for k > 1 robots.

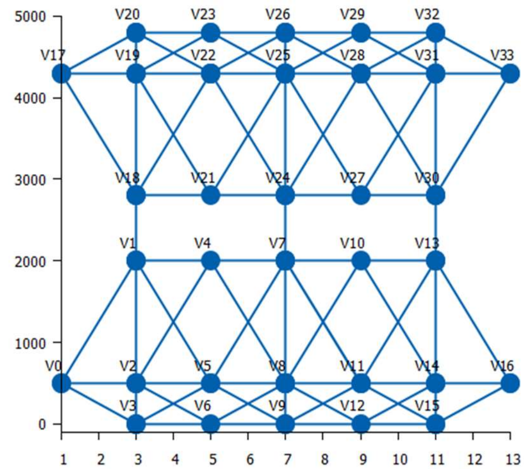

Figure 2. A graphical representation of the bridge.

3.2 Genetic Algorithms

A Genetic Algorithm is a metaheuristic for optimization and search problems that evolves solutions by mimicking the process of natural selection. The process begins with an initial population comprised of randomly generated individuals; each individual contains a genetic representation of its solution called a chromosome. The process is iterative, each iteration produces a new population, or generation, of solutions. Every generation, each individual representing a candidate solution, is evaluated using a fitness function, this assigns a numeric value to each individual based on the quality of its solution; individuals with a higher fitness represent a better solution than those with a lower fitness. To form the next generation of solutions, individuals can be chosen using a variety of semi-random methods where the probability of any given individual being chosen is proportional to its fitness, so that high quality solutions are more likely to pass onto the next generation. Once a group of solutions has been selected, their chromosomes are combined, altered, or possibly randomly mutated to form the next generation using methods described further in the next subsection. This process will continue for a set number of generations or until a target fitness level is reached and the highest quality solution is returned.

3.3 Chromosome Representation and Evaluation

For a GA to work, it requires two things: a genetic representation of a candidate solution to the problem (usually encoded in a string like structure called a chromosome), and a fitness function to evaluate the quality of those solutions. In our GA an individual's chromosome is stored as a string of numbers representing paths for k robots. Each number, read from left to right, represents an edge from the graph of our bridge to be traveled in that order; separating these numbers are robot identifiers, one unique identifier for each robot. An individual robots route is represented as the path of each edge starting from its identifier up until the next robots identifier. In the case where the end of the string is reached before another identifier is reached it will simply loop around to the beginning and continue until an identifier is found. Such a string like

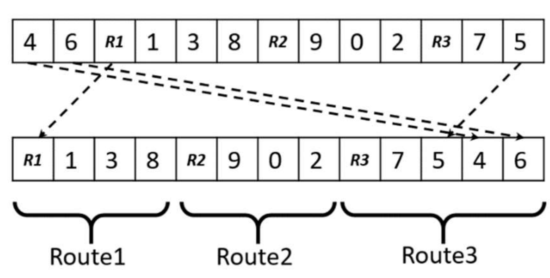

Figure 3. Chromosome and route splitting.

representation makes it easy for the genetic operators of crossover and mutation to generate new solutions. The GAs fitness function evaluates the individuals solution based on the distance of the path. There are two parts to this: the total path length, which is the combined length of all robots individual routes, and the length of the longest individual route. The fitness of a solution is equal to the longest route within the k routes. This way the fitness function is minimizing the longest route as well as minimizing the total length of k routes. When evaluating a member of the GA population, we convert the list of edges in the chromosome into a waypoint list as follows. If edges in the chromosome are adjacent nothing needs to be done and we add to the waypoint list. When two edges in the list are not adjacent, we uses Dijkstra's algorithm to calculate the shortest path between the two edges and construct the waypoint list for that route. Individuals containing disconnected edges will have longer distance routes than individuals containing connected edges. Therefore, the GA will favor the individuals with more connected edges and less disconnected edges. To calculate the distance of a path, the GA translates the path representation from a series of edges into a series of waypoints, this process is done two edges at a time. The four points involved, each edges start and finish, are analyzed by running Dijkstra's algorithm to find the shortest path between each point. Once all the edges have been converted to points on our graph, we can simply sum up the distance between them to get the routes length. In our GA, chosen individuals chromosomes are segmented and swapped using ordered crossover (OX) to form the next generation of solutions.

4. Results and Discussion

The experiments were designed to apply GAs to evolve routes for multiple inspection robots on a bridge inspection task. Each experiment is allowed to run for 106 fitness evaluations on the graph which is generated from the simulated bridge with 35 nodes and 81 arcs in total. As the global optima of the multiple robots path planning is unknown, the global optima of one robot path planning is used as a reference value to estimate the solution quality of different number of robots.

Algorithm 1 Finding an Optimal Chinese Postman Route, Eiselt (1995)

Step 1: List all odd nodes.
Step 2: List all possible pairings of odd nodes.
Step 3: For each pairing find the edges that connect the nodes with the minimum weight.
Step 4: Find the pairings such that the sum of the weights is minimized.
Step 5: On the original graph add the edges that have been found in Step 4.
Step 6: The length of an optimal Chinese Postman route is the sum of all the edges added to the total found in Step 4.
Step 7: A route corresponding to this minimum weight can then be easily found.

In a preliminary experiment the goal was to evaluate whether the GA is a suitable search algorithm for searching multiple routes for MM k-CPP problem. Therefore, we compare the quality of solutions produced by GAs with the solution using traditional method on a CPP with a single inspecting robot. To find a minimum Chinese Postman route we must go through each edge at least once and in addition we must also go through the least pairings of odd nodes on one extra occasion. We apply the Algo. 1 on the graph described in Fig. 2 which is derived from the bridge shown in Fig. 1 and found the optimal solution to be 51,268 which will be used as our upper bound baseline for our solutions found by GAs and for searching routes for multiple inspecting robots. Fig. 4 shows the result of finding a near optimal route for one robot. The result indicate that the best solution found by GA is 51,974 which is only 1.377% extra distance compared to the optimal solution derived from Algo. 1.

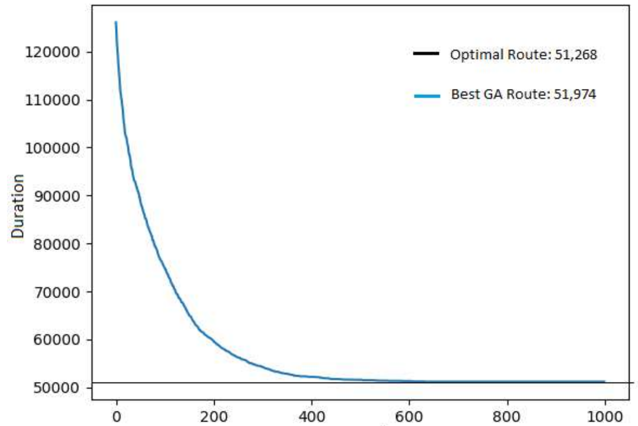

Figure 4. The average over 10 runs of GA on one robot

Although this experiment shows that GA finds near optimal routes close to the global optima for one robot, we are more interested in using multiple robots. Algo. 1 is no longer suitable for evenly splitting the optimal route into multiple routes. Therefore, we continue with our GA and investigate performance on the MM 2-CPP version of the problem with two robots. Fig. 5 shows the evolutionary progress of GA on searching near optimal solutions for two robots. The best solution found by the GA for two robots is 52,968 which is 3.316% over the global optima on the one robot scenario.

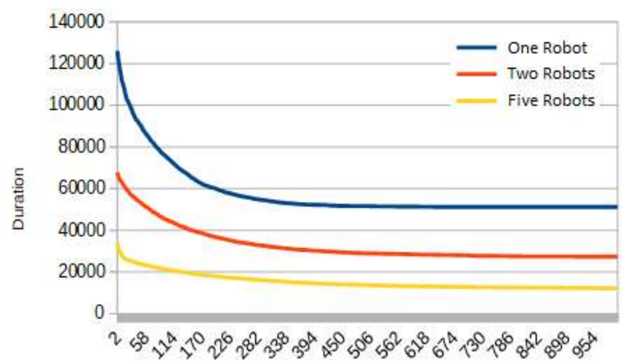

Figure 5. Average over 10 runs of GAs on 2 and 5

The two routes are visualized on Fig. 5 with route 1 to be 26,334 and route 2 to be 26,634 which is split closely by GAs. Comparing to the one robot scenario, the two robots are able to share the similar amount of the distance with the total amount of the two routes to be close to the best solution found by GAs for one robot scenario. This indicates that our total amount of time of using two robots to conduct the bridge inspection is only half of the time of using one robot. The best solution for two robots is shown in Table I . We noticed that the best solution found by GAs for two robots is slightly worse than the best solution found by GAs for one robot. This is

TABLE I: Best Solution for two Routes

Name	Waypoints	Distance
R1	26→25→29→31→27→25→23→20→17→19→18→17→18→21→19→22→24→7→11→9→8→9→6→3→0→2→5→8→12→15→14→13→30→27→24→21→25→1→22→20→19→23→26→22→18→1→0→2→3→5→1	26,334
R2	24→25→28→26→29→32→28→30→31→28→24→7→4→1→2→6→8→5→7→10→13→11→15→14→12→9→5→2→4→8→11→14→16→15→16→13→30→33→32→31→33→31→27→24→7→8→10→14	26,634

because we encoded robots into the chromosome along with the edges which means the more robots we have in the experiments, the larger search space our GAs need to search. This is also indicated by that at the same generation, the GAs found the distance for one robot is better than two robots.

We further extend our experiments from two robots to five robots for evaluating the generalization of our approach. Fig. 5 shows the best solution found by GAs when using five robots to inspecting the same bridge described in Fig. 1. The results shows that the best solution for five robots is 55,480 which is 8.216% extra distance comparing to the global optima. The best solution for five robots is shown in Table II. Comparing to the one robot scenario, the five robots are able to share the similar amount of the distance with the total amount of the five routes to be close to the best solution found by GAs for one robot scenario. This indicates that our total amount of time of using five robots to conduct the bridge inspection is only one fifth of the time of using one robot. The duration of a bridge inspection project linearly decreases based on the number of robots used in the project.

TABLE II: Best Solution for five Routes

Name	Waypoints	Distance
R1	14→15→16→14→11→9→6→3→2→5→2→0→1→5→3→2→4→1→18→22→25→23→20→22	11,376
R2	30→28→25→29→31→28→32→31→27→24→7→11→8→9→12→14→13→30→33	9,776
R3	16→13→10→7→24→25→22→26→28→25→26→29→32→33→31→29→26→23→19→21→25→27	11,176
R4	17→20→19→18→21→18→1→2→0→3→6→2→6→8→5→9→11→14→10→13→11→8→10	11,682
R5	11→15→12→8→5→7→4→8→7→24→28→31→30→27→24→22→19→17→18→21→24→7	11,470

6. Conclusion and Future Work

This paper investigates using evolutionary algorithms on a MinMax k-Chinese postman problem on generating k routes each for a robot in bridge inspection projects. We encoded k routes for a group of inspecting robots as a series of edges to be traversed by the robots and used a GA to find near optimal solutions that minimize the total distance traversed while covering every edge and balancing the distances traversed by each robot. In order to put our work in context, we compare our experiment results with the global optimal solution based on one robot traverse the same graph. Then we extended our experiments to search for two balanced routes for two inspecting robots in a project. We further extended our experiments to search for five balanced routes which assumes we deploy five inspection robots. The results show that using our route representation, GAs could find a near optimal solution within 1.377% of the global optimum found using Algo. 1. For two robots, where Algo. 1 no longer applies, our GAs found two approximately equal length routes and each route is approximately half of the length comparing to the global optimal route. Extending to five robots, we get similar results which all the edges on the bridge are covered at least once and each route is approximately one fifth of the global optimal route. The results show that using our GAs we are able to linearly speedup our bridge inspection tasks for each robot added. By automating route generation for inspection robots we reduced the cost and time involved in routine bridge inspection. In future work, we are interested in applying our evolutionary approach on multiple extensions of MM k-CPP problems including robots moving with different speeds during inspection and traveling without inspecting. We are also interested in extending the undirected graph considered in MM k-CPP to a directed graph with different costs on different directions.

7. References

J. Edmonds and E. L. Johnson, "Matching, euler tours and the chinese postman," Mathematical Programming, vol. 5, no. 1, pp. 88–124, Dec 1973.

H. Eiselt, M. Gendreau, and G. Laporte, "Arc routing problems, part i: The chinese postman problem," Oper. Res., vol. 43, no. 2, pp. 231–242, Apr. 1995.

C. Papadimitriou, "On the complexity of edge traversing," J. ACM, vol. 23, no. 3, pp. 544–554, Jul. 1976.

E. Minieka, "The chinese postman problem for mixed networks," Management Science, vol. 25, no. 7, pp. 643–648, 1979.

G. N. Frederickson, M. S. Hecht, and C. E. Kim, "Approximation algorithms for some routing problems," in 17th Annual Symposium on Foundations of Computer Science (sfcs 1976), Oct 1976, pp. 216–227.

D. Ahr and G. Reinelt, "New heuristics and lower bounds for the min-max k-chinese postman problem," in Algorithms — ESA 2002, R. M¨ohring and R. Raman, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 64–74.

D. Ahr and G. Reinelt, "A tabu search algorithm for the minmax k-chinese postman problem," Computers & Operations Research, vol. 33, no. 12, pp. 3403 – 3422, 2006, part Special Issue: Recent Algorithmic Advances for Arc Routing Problems.

H. Chen, T. Cheng, and J. Shawe-Taylor, "A balanced route design for min-max multiple-depot rural postman problem (mmmdrpp): a police patrolling case," International Journal of Geographical Information Science, vol. 32, no. 1, pp. 169–190, 2018.

M. Gendreau, A. Hertz, and G. Laporte, "A tabu search heuristic for the vehicle routing problem," Management science, vol. 40, no. 10, pp. 1276–1290, 1994.

S. Salhi and M. Sari, "A multi-level composite heuristic for the multidepot vehicle fleet mix problem," European Journal of Operational Research, vol. 103, no. 1, pp. 95–112, 1997.

R. T. Sumichras and I. S. Markham, "A heuristic and lower bound for a multi-depot routing problem," Computers & Operations Research, vol. 22, no. 10, pp. 1047 – 1056, 1995.

G. Clarke and J. W. Wright, "Scheduling of vehicles from a central depot to a number of delivery points," Operations Research, vol. 12, no. 4, pp. 568–581, 1964.

S. R. Thangiah and S. Salhi, "Genetic clustering: An adaptive heuristic for the multidepot vehicle routing problem," Applied Artificial Intelligence, vol. 15, no. 4, pp. 361–383, 2001.

S. Wink, T. B¨ack, and M. Emmerich, "A meta-genetic algorithm for solving the capacitated vehicle routing problem," in Evolutionary Computation (CEC), 2012 IEEE Congress on. IEEE, 2012, pp. 1–8.

I. Choi, S. Kim, and H. Kim, "A genetic algorithm with a mixed region search for the asymmetric traveling salesman problem," Computers & Operations Research, vol. 30, no. 5, pp. 773–786, 2003.