

# A Genetic Algorithm for Multi-Robot Routing in Automated Bridge Inspection

Nicholas Harris  
University of Nevada, Reno  
nicholasharris@nevada.unr.edu

Siming Liu  
Missouri State University  
simingliu@missouristate.edu

Sushil J. Louis, Jim Hung La  
University of Nevada, Reno  
sushil.hla@unr.edu

## ABSTRACT

We attack the problem of generating balanced and efficient routing for automated inspection by using genetic algorithms to solve the equivalent Min-Max  $k$  Windy Chinese Postman Problem. Specifically, we use  $k$  robots to collectively inspect every member of a steel truss bridge. Experimental results show that the genetic algorithm produces efficient routes that are well-balanced among the robots. Additionally, we demonstrate that with our novel representation, as the number of robots increases, the generated routes exhibit near-linear speedup in the time needed to complete the inspection task -  $k$  robots take  $\frac{1}{k}$ <sup>th</sup> the time needed by one robot. Finally, our genetic algorithm produces similar results on a set of benchmark arc routing problem instances from the literature.

## CCS CONCEPTS

• **Computing methodologies** → **Robotic planning**; • **Theory of computation** → *Discrete optimization*; Graph algorithms analysis;

## KEYWORDS

Genetic algorithms, Arc routing, Robot inspection, Bridge inspection

### ACM Reference format:

Nicholas Harris, Siming Liu, and Sushil J. Louis, Jim Hung La. 2019. A Genetic Algorithm for Multi-Robot Routing in Automated Bridge Inspection. In *Proceedings of Genetic and Evolutionary Computation Conference Companion, Prague, Czech Republic, July 13–17, 2019 (GECCO '19 Companion)*, 2 pages. <https://doi.org/10.1145/3319619.3321917>

## 1 INTRODUCTION

Arc Routing Problems (ARPs) are a broad range of problems that include the Chinese Postman Problem (CPP) and its many variants, some of which are NP-hard, such as the Windy Postman Problem (WPP) and Min-Max  $k$ -Chinese Postman Problem (MM  $k$ -CPP). In each of these, we work with a graph  $G = (V, E)$ , which includes a set of vertices  $V$ , a set of edges  $E$ , and a set of weights,  $w : E \rightarrow \mathbb{R}^+$ , or costs associated with each edge. The task in the traditional CPP is to find the optimal closed tour for one vehicle that covers every edge

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

GECCO '19 Companion, July 13–17, 2019, Prague, Czech Republic

© 2019 Association for Computing Machinery.

ACM ISBN 978-1-4503-6748-6/19/07...\$15.00

<https://doi.org/10.1145/3319619.3321917>

of the graph at least once. The Windy Postman Problem includes an additional constraint, where the cost of traversing an edge in one direction may be different from doing so in the opposite direction. The Min-Max  $k$ -Chinese Postman, meanwhile, extends the CPP by requiring that the work of covering the graph be divided among a group of  $k$  vehicles. Moreover, the task in this problem is to minimize the length of the longest route traveled by any of the  $k$  vehicles. In this paper we examine the Min-Max  $k$  Windy Chinese Postman Problem (MM  $k$ -WCPP), which combines the constraints of the WPP and MM  $k$ -CPP. This problem arises in the context of planning routes for a team of  $k$  wheeled robots to collectively inspect every member of a steel truss bridge. Modelling this problem as the MM  $k$ -CPP is insufficient because the robot's traversal speed is affected by the incline of the beam. Therefore, we incorporate the windy constraint and assign each member two weights: one for uphill traversal and one for downhill. These weights will be equal if the member is level. A slight variation on the MM  $k$ -WCPP is made, however, in that we do not require the routes taken by the  $k$  robots to be closed tours, but may begin an end at any vertex. This is done because we assume our robots may be deployed and retrieved at any juncture on the bridge.

We examine the use of a genetic algorithm (GA) on this problem. GAs have been shown to be competitive with existing algorithms at solving other arc routing problems [5]. Other practical applications of ARPs have been studied, such as garbage collection [1] [4] and road salting [6]. We apply the MM  $k$ -WCPP to the practical situation of planning bridge inspection routes for multiple robots.

## 2 METHODOLOGY

### 2.1 Bridge Representation

To model bridge inspection as an MM  $k$ -CCP, we convert a bridge instance to a corresponding weighted graph. We created a virtual model bridge containing 83 steel members for testing and converted this to a corresponding graph with 83 edges and 34 vertices. Edge weights (or costs) are proportional to edge length and direction with double costs for uphill travel. With this we have defined a full MM  $k$ -CPP problem instance from the model bridge.

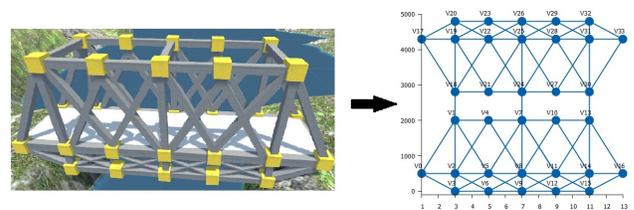


Figure 1: Model bridge converted into graph form.

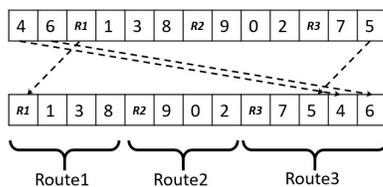
**Table 1: Performance of GA on problem instances.**

Problem	V	E	Baseline	Best $k = 1$	Percent Above Baseline	Best $k = 2$	Speedup $k = 2$	Best $k = 5$	Speedup $k = 5$
C03	28	57	101.5	96	-5.42 %	50	1.92	25	3.84
C19	33	54	255.5	252	-1.37 %	133	1.89	64	3.94
C20	50	98	698	713	2.15 %	383	1.86	177	4.03
C21	49	110	525	544	3.62 %	283	1.92	141	3.86
C24	41	125	1039	1070	2.98 %	555	1.93	255	4.20
Bridge	34	83	64120	66638	3.93 %	35674	1.87	15646	4.26
Averages	-	-	-	-	0.98%	-	1.90	-	4.02

### 2.2 Genetic Algorithm

Given  $M$  edges and  $k$  vehicles our chromosomes length becomes  $M + k$ . As shown in Figure 2, the chromosome consists of the integers 1 through  $M$ , as well as  $k$  special elements, in any order. The elements 1 through  $M$  represent the edges from the graph in the order they will be traversed, while the special elements serve as separators that split the edges among the  $k$  vehicles. The first vehicle traverses the edges between the first pair of separators, the second vehicle traverses those between the second pair, and so on. The route of the final vehicle is obtained by reading the chromosome in a loop, so that the edges that follow the final separator element wrap around to the beginning until the first separator. Figure 2 shows an example chromosome being split into separate routes. In the case where two edge elements are adjacent in the chromosome but not on the graph, we use Dijkstra’s algorithm to calculate the shortest path between the two edges so that a full route is obtained.

We use the highly elitist CHC generational scheme for selection [3] and Order Crossover (OX). We allowed two kinds of mutation: slide mutation, where a random section of the chromosome slides one position over; and invert mutation, where a random section of the chromosome is reversed in order. In all tests the probability of crossover used was 0.99 and the probability of mutation was 0.4. All tests used a population size of 1,000 and ran for 1,000 generations. For bridge inspection, we want to be able to generate efficient routes quickly; these choices of population size and duration allow a single run of the GA to be completed in a matter of minutes.



**Figure 2: Chromosome and route splitting.**

In addition to the model bridge, and to test generalizability, we ran our GA on a set of benchmark problem instances modified from [2]. These were converted from the "Rural" variation of the MM  $k$ -WCPP, where only a subset of the edges must be traversed.

### 3 RESULTS AND CONCLUSIONS

Since the optimal routes for our particular variation of this problem are unknown, a baseline value for comparison is calculated in each

problem as the sum of the cost of each edge, with the cost of an edge being taken as the average of the cost in either direction. The best results obtained in 5 runs of the GA on the model bridge and benchmark problem instances are shown in Table 1. These results are promising firstly because the GA often performs well compared to the baseline value, and secondly because the speedup of the task is close to linear as the number of vehicles increases.

These results demonstrate the effectiveness of the GA in producing efficient and balanced routes in the context of the MM  $k$ -WCPP, and in particular in the problem instance of bridge inspection conducted by multiple robots. Further study of GAs in arc routing problems, particularly with new constraints and on a broader range of problem graphs, may be rewarding.

### ACKNOWLEDGMENTS

This work was supported by grant number N00014-17-1-2558 from the Office of Naval Research. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Office of Naval Research. Dr. Louis was supported in part by the U.S. Department of Transportation, Office of the Assistant Secretary for Research and Technology (USDOT/OST-R) under Grant No. 69A3551747126 through INSPIRE University Transportation Center (<http://inspire-utc.mst.edu>) at Missouri University of Science and Technology. The views, opinions, findings and conclusions reflected in this publication are solely those of the authors and do not represent the official policy or position of the USDOT/OST-R, or any State or other entity.

### REFERENCES

- [1] S.K. Amponsah and S. Salhi. 2004. The investigation of a class of capacitated arc routing problems: The collection of garbage in developing countries. *Waste Management* 24, 7 (2004), 711–721.
- [2] Enrique Benavent, Ángel Corberán, Isaac Plana, and José María Sanchis. 2009. Min-Max  $K$ -vehicles windy rural postman problem. *Networks* 54 (2009), 216–226.
- [3] L. J. Eshelman. 1991. The CHC Adaptive Search Algorithm : How to Have Safe Search When Engaging in Nontraditional Genetic Recombination. *Foundations of Genetic Algorithms* (1991), 265–283. <http://ci.nii.ac.jp/naid/10000024547/en/>
- [4] Gianpaolo Ghiani, Francesca Guerriero, and Gennaro Improtà. 2005. Waste collection in Southern Italy: Solution of a real-life arc routing problem. *Int Trans Oper Res* 12, 2 (2005), 135–144.
- [5] Lacomme, Prins, and Sevaux. 2006. A genetic algorithm for a bi-objective capacitated arc routing problem. *Computers & Operations Research* 33, 12 (2006), 3473–3493.
- [6] L. Muyldermans, D. Cattrysse, D. Van Oudheusden, and Lotan T. 2002. Districting for salt spreading operations. *European Journal of Operational Research* 139, 3 (2002), 521–532.