Scaling ACO to Large-scale Vehicle Fleet Optimisation Via Partial-ACO

Darren M. Chitty, Elizabeth Wanner, Rakhi Parmar, Peter R. Lewis
Aston Lab for Intelligent Collectives Engineering (ALICE)
Aston University, Birmingham. B4 7ET UK
{d.chitty,e.wanner,r.parmar8,p.lewis}@aston.ac.uk

ABSTRACT

The benefits of optimising fleets of vehicles regards scheduling tasks are threefold; reduced costs, reduced road use, and most importantly, reduced emissions. However, optimisation methods, both exact and meta-heuristic, scale poorly. This issue is addressed with *Partial-ACO*, a novel variant of ACO that scales by ants only partially modifying good solutions. For real-world fleet optimisation problems supplied by a Birmingham company of up to 298 jobs and 32 vehicles, *Partial-ACO* demonstrates better scalability than ACO and GAs reducing the company's fleet traversal by over 40%.

CCS CONCEPTS

• **Theory of computation → Evolutionary algorithms**;

KEYWORDS

Fleet Optimisation, Multi-Depot Vehicle Routing Problem, ACO

ACM Reference Format:

Darren M. Chitty, Elizabeth Wanner, Rakhi Parmar, Peter R. Lewis. 2019. Scaling ACO to Large-scale Vehicle Fleet Optimisation Via Partial-ACO. In *Genetic and Evolutionary Computation Conference Companion (GECCO '19 Companion), July 13–17, 2019, Prague, Czech Republic.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3319619.3322048

1 INTRODUCTION

Fleet optimisation is a common problem faced by organisations from delivery companies, maintenance firms and medical professionals performing care in the community. They have many tasks over a geographical area and a set of vehicles to carry them out with. The problem is to assign tasks to vehicles and their ordering to minimise the traversal time of the vehicle fleet. The primary gain is reduced fuel and labour costs but an important added benefit is a reduction in emission levels. The World Health Organisation (WHO) reports levels of particulates such as nitrogen oxides (NO_x) in major cities are increasing[1] causing breathing problems and linked to increased cardiovascular disease. Many cities must maintain low levels of particulates using clean air policies like Birmingham City Council[2], and fleet optimisation can assist in this goal. However, optimising fleets of vehicles is NP-hard and heuristic techniques fail to scale well with problem size. This paper profiles a recent advance in Ant Colony Optimisation (ACO) [4] known as *Partial-ACO* [1] and demonstrates its ability to scale better when applied to real-world fleet optimisation problems.

2 PARTIAL-ACO FOR FLEET OPTIMISATION

ACO simulates ants traversing a fully connected graph G probabilistically visiting vertices once depositing pheromone on edges E defined by the solution quality. Ants probabilistically decide which vertex to visit using pheromone levels on the edges of graph G plus heuristic information. An *evaporation* effect limits pheromone levels. However, ACO has scalability issues, first the requirement for a pheromone matrix, a 100,000 vertex problem requires 37GB of memory. Second, the probabilistic nature of ants deciding vertices to visit using pheromone on the edges E. Eventually, an ant will probabilistically make a poor decision even with high pheromone levels on the optimal edge. It can be hypothesized that ACO by its probabilistic nature becomes less likely to reach optimality as problem sizes increase with quadratically rising computational costs.

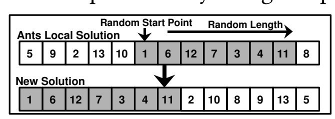

Figure 1: An illustration of the *Partial-ACO* methodology.

Table 1: Birmingham maintenance company scenarios.

Problem	Number of Vehicles	Number of Jobs	Job Servicing Time (hh:mm)	Fleet Traversal Time (hh:mm)
Week_A	8	77	47:09	31:12
Week_B	8	79	48:24	22:49
2Week_A	16	156	95:33	54:01
2Week_B	16	138	102:01	57:07
3Week_A	24	219	150:34	77:01
3Week_B	24	221	151:49	68:38
Month_A	32	298	198:58	99:50

To address this scalability, the derivative *Partial-ACO* has been proposed enabling ACO to be applied to TSP instances up to 200,000 cities [1]. *Partial-ACO* operates similarly to ACO but without a pheromone matrix, addressing the first scalability issue. Instead, pheromone is calculated from a population of ants and their respective solutions. *Partial-ACO* maintains a population of ants each with a *local memory* (l_{best}) of the best solution it has found operating in a *steady state* manner. Pheromone deposit of ant k on an edge E of graph G is related to the quality of solution l^k_{best} compared to the global best solution, g_{best} ensuring consistent pheromone level deposits as improved solutions are found negating any scalability issues. An ant reconstructs edge pheromone levels by iterating through all l_{best} solutions of the ant population finding edges taken from and arriving at the current location. The second component of *Partial-ACO* addresses the probability of poor decisions

[1]Air pollution levels rising in many of the worlds poorest cities. http://www.who.int/mediacentre/news/releases/2016/air-pollution-rising/en
[2]A Clean Air Zone for Birmingham https://www.birmingham.gov.uk/caz

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

ACM ISBN 978-1-4503-6748-6/19/07…$15.00
https://doi.org/10.1145/3319619.3322048

Table 2: The results from each meta-heuristic approach regards optimising the schedules of the real-world scenarios.

Problem	Genetic Algorithm		Max Min Ant System		Partial-ACO		Partial-ACO^{PH}	
	Job Time Serviced (%)	Traversal Time Reduction (%)	Job Time Serviced (%)	Traversal Time Reduction (%)	Job Time Serviced (%)	Traversal Time Reduction (%)	Job Time Serviced (%)	Traversal Time Reduction (%)
Week_A	100.00 ± 0.00	28.10 ± 7.15	100.00 ± 0.00	33.62 ± 3.39	100.00 ± 0.00	32.29 ± 3.77	100.00 ± 0.00	37.14 ± 2.15
Week_B	100.00 ± 0.00	28.06 ± 10.60	100.00 ± 0.00	30.70 ± 4.85	100.00 ± 0.00	22.39 ± 7.84	100.00 ± 0.00	38.20 ± 2.88
2Week_A	100.00 ± 0.00	25.38 ± 5.93	100.00 ± 0.00	23.84 ± 7.46	99.98 ± 0.10	23.12 ± 6.30	100.00 ± 0.00	36.43 ± 1.37
2Week_B	100.00 ± 0.00	27.33 ± 4.32	100.00 ± 0.00	22.55 ± 5.01	99.98 ± 0.09	25.81 ± 4.72	100.00 ± 0.00	30.56 ± 1.91
3Week_A	100.00 ± 0.00	29.62 ± 4.21	99.95 ± 0.11	7.33 ± 6.56	99.91 ± 0.21	20.64 ± 2.36	100.00 ± 0.00	20.01 ± 2.71
3Week_B	100.00 ± 0.00	26.26 ± 4.13	99.86 ± 0.18	−2.36 ± 5.92	99.84 ± 0.27	18.00 ± 7.84	100.00 ± 0.00	13.74 ± 3.84
Month_A	100.00 ± 0.00	29.23 ± 4.18	99.76 ± 0.18	−17.85 ± 3.75	99.82 ± 0.18	19.60 ± 4.09	98.04 ± 0.49	−26.59 ± 8.48

during solution construction and computational costs. Each decision point has a given probability of a poor decision occurring with decision choices for a 100,000 vertex problem requiring five billion pheromone comparisons. Consequently, *Partial-ACO* proposes an ant when building a solution takes its l_{best} tour and retains part of this tour and completes the rest in the same probabilistic manner as normal. A random point is first selected in the l_{best} tour and then a random length of the tour to be retained with this section copied into the new solution. The remaining part is then constructed as normal (see Figure 1). This partially modified solution replaces an ant's l_{best} solution if an improvement. Reducing probabilistic decision reduces the probability of error and pheromone comparisons.

Fleet optimisation is in effect the Multi-Depot Vehicle Routing Problem (MDVRP) [3], vehicles operate from depots with jobs to complete with vehicle fleet traversal time minimised. Solutions consist of the vehicle set and their tasked jobs in the order they need to be completed. To build solutions a vehicle is randomly selected from which ants probabilistically select unfulfilled jobs or another vehicle whereby the current vehicle returns to its depot. Solution quality is measured using two objectives: maximise the number of jobs performed within their time windows and minimise the total traversal time of the vehicle fleet. See Chitty et al. [2] for further details on *Partial-ACO* applied to this fleet optimisation problem.

3 RESULTS

To evaluate *Partial-ACO* for fleet optimisation, a real-world problem is used from a maintenance company based in Birmingham with multiple vehicles and geographical customers. Vehicles have depots to return to when finished. Customers are defined by a location, job duration, and often, a time window. The working day is defined as between 08:00 and 19:00 hours. Problems range in time periods, jobs and vehicle availability with current company schedules enabling real-world reductions to be ascertained (see Table 1).

Partial-ACO will be compared to Max Min Ant System (MMAS) [6] and Genetic Algorithm (GA) [5] approaches. The GA uses crossover operators cyclic (CX), order (OX) and partially mapped (PMX) with mutations swap, order reversal and insertion. The GA operates in a *steady state*, child solutions replace parents if their quality is better. An alternative implementation of *Partial-ACO* will also be tested which does use a pheromone matrix termed *Partial-ACO*^{PH}. MMAS, GA and *Partial-ACO*^{PH} use a population of 192 run for a million iterations and *Partial-ACO* a population of 32 for six million iterations. Experiments averaged over 25 execution runs. Results in Table 2 demonstrate both GA and MMAS approaches improve upon the given company job scheduling for smaller problem instances. However, ACO fails to scale as previously hypothesized unlike the GA. Regards *Partial-ACO*, both variants improve upon

Table 3: Results from limiting maximum modification.

Max. Mod.	Problem	Partial-ACO		Partial-ACO^{PH}	
		Job Time Serviced (%)	Traversal Reduction (%)	Job Time Serviced (%)	Traversal Reduction (%)
50%	Week_A	100.00 ± 0.00	41.17 ± 0.49	100.00 ± 0.00	41.94 ± 1.37
	Week_B	100.00 ± 0.00	42.39 ± 1.11	100.00 ± 0.00	43.48 ± 0.44
	2Week_A	100.00 ± 0.00	23.89 ± 6.54	100.00 ± 0.00	34.30 ± 2.39
	2Week_B	100.00 ± 0.00	24.41 ± 5.13	100.00 ± 0.00	34.08 ± 1.12
	3Week_A	100.00 ± 0.00	27.12 ± 4.17	100.00 ± 0.00	25.94 ± 1.87
	3Week_B	100.00 ± 0.00	22.52 ± 4.05	100.00 ± 0.00	22.29 ± 1.53
	Month_A	100.00 ± 0.00	28.19 ± 5.90	98.67 ± 0.28	−5.98 ± 3.54
25%	Week_A	100.00 ± 0.00	32.93 ± 13.07	100.00 ± 0.00	30.12 ± 2.39
	Week_B	100.00 ± 0.00	33.91 ± 5.24	100.00 ± 0.00	30.84 ± 4.91
	2Week_A	100.00 ± 0.00	28.08 ± 7.53	100.00 ± 0.00	31.38 ± 2.37
	2Week_B	100.00 ± 0.00	30.66 ± 6.14	100.00 ± 0.00	32.15 ± 2.67
	3Week_A	100.00 ± 0.00	33.27 ± 6.85	100.00 ± 0.00	26.18 ± 2.89
	3Week_B	100.00 ± 0.00	31.66 ± 6.74	100.00 ± 0.00	22.78 ± 2.42
	Month_A	100.00 ± 0.00	32.91 ± 3.73	99.41 ± 0.32	5.35 ± 5.65

ACO reinforcing this hypothesis. *Partial-ACO*^{PH} has the edge over *Partial-ACO* for the smaller problems but is less capable of scaling.

Partial-ACO can be further enhanced by limiting the degree by which an ant can modify its l_{best} solution as originally postulated [1]. Doing so further reduces ants probabilistic decision making minimising error potential and computational cost. Table 3 demonstrates the effect for both *Partial-ACO* approaches with restrictions of 50% and 25% with considerable improvements achieved. Indeed, *Partial-ACO* now scales to larger problem instances much better although *Partial-ACO*^{PH} achieves slightly better solutions for smaller problem instances. When restricting to 25% for larger problems, improved reductions in fleet traversal are achieved of 30-33%, better than a GA. However, for smaller problems results are poorer as there is little capacity for change. Reductions up to 42% are achieved over the original company schedules using *Partial-ACO*.

4 ACKNOWLEDGEMENT

Carried out under the System Analytics for Innovation project, part-funded by the European Regional Development Fund.

REFERENCES

[1] Darren M Chitty. 2017. Applying ACO to Large Scale TSP Instances. In *UK Workshop on Computational Intelligence*. Springer, 104–118.

[2] Darren M Chitty, Elizabeth Wanner, Rakhi Parmar, and Peter R Lewis. 2019. Applying Partial-ACO to Large-scale Vehicle Fleet Optimisation. *arXiv Tech. Rep. arXiv:1904.07636* (2019).

[3] George B Dantzig and John H Ramser. 1959. The truck dispatching problem. *Management science* 6, 1 (1959), 80–91.

[4] Marco Dorigo and Luca Maria Gambardella. 1997. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on evolutionary computation* 1, 1 (1997), 53–66.

[5] John H Holland. 1975. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.* U Michigan Press.

[6] Thomas Stützle and Holger H Hoos. 2000. MAX–MIN ant system. *Future generation computer systems* 16, 8 (2000), 889–914.