Partial-ACO as a GA Mutation Operator Applied to TSP Instances

Darren M. Chitty
Aston Lab for Intelligent Collectives Engineering (ALICE)
Aston University, Birmingham, UK
darrenchitty@googlemail.com

ABSTRACT

A recent novel modification to Ant Colony Optimisation (ACO) known as Partial-ACO can be successfully used to solve Travelling Salesman Problems (TSP) by making partial modifications. The approach also dispenses with a pheromone matrix using the population to build pheromone levels on edges enabling scaling to large problems. Consequently, being population based the approach can be also used within a Genetic Algorithm as a mutation operator. Results demonstrate significant improvements when using Partial-ACO as a mutation operator with a range of crossover operators.

CCS CONCEPTS

• **Mathematics of computing → Evolutionary algorithms**.

KEYWORDS

Ant Colony Optimisation, Genetic Algorithm, TSP

ACM Reference Format:
Darren M. Chitty. 2021. Partial-ACO as a GA Mutation Operator Applied to TSP Instances. In *2021 Genetic and Evolutionary Computation Conference Companion (GECCO '21 Companion), July 10–14, 2021, Lille, France.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3449726.3459424

1 INTRODUCTION

Meta-heuristic approaches are popular for solving Travelling Salesman Problems (TSP) such as Genetic Algorithms (GAs) [4] or Ant Colony Optimisation (ACO) [2]. The goal of the TSP is to visit all cities once minimising traversal. The symmetric TSP is represented as a complete weighted graph $G = (V, E, d)$ where $V = \{1, 2, .., n\}$ is a set of vertices defining cities and $E = \{(i, j)|(i, j) \in V \times V\}$ the edges with distance d between pairs of cities such that $d_{ij} = d_{ji}$. The objective is to find a Hamiltonian cycle in G of minimal length.

ACO has two issues for scaling to large problems, a memory overhead from a pheromone matrix and a computational cost of simulating ants. A novel variant to ACO known as Partial-ACO [1] dispenses with a pheromone matrix using an ant population to construct pheromone levels. Moreover, only *partial* changes are made to ants best found solutions enabling Partial-ACO to scale to TSPs of over 100k cities. As Partial-ACO constructs pheromone levels via a solution population these can be ants or in fact GA solutions. Hence, the approach could be embedded inside a GA. This paper presents Partial-ACO within a GA as a mutation operator.

ACM ISBN 978-1-4503-8351-6/21/07.
https://doi.org/10.1145/3449726.3459424

2 PARTIAL-ACO INSPIRED GA MUTATION

ACO applied to the TSP involves simulated ants moving through graph G visiting each city depositing pheromone defined by the quality of the ant's tour. Ants construct tours by probabilistically deciding cities to visit next using this pheromone and heuristic information of edge length using the *random proportional rule*, the probability of ant k at city i visiting city $j \in N^k$ defined as:

$$p_{ij}^k = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N^k} [\tau_{il}]^\alpha [\eta_{il}]^\beta} \quad (1)$$

where $[\tau_{il}]$ is the pheromone on edge of city i to city l; $[\eta_{il}]$ is heuristic information, $1/d_{il}$; α and β are tuning parameters.

An alternative meta-heuristic is a GA which uses the principles of Darwinian evolution to find optimal solutions. A population of solutions is iteratively improved using natural selection, genetic crossover and mutation. With the TSP genetic material represents a tour of cities to be visited. As cities can only occur once specialist crossover operators prevent a given city existing twice in a solution. To generate offspring genetic material from one parent between two crossover points is copied. Genetic material is then copied from a second parent unless a city is already present. Remaining unvisited cities are resolved using a range of crossover methods such as cyclic (CX), partially mapped (PMX) or order based (OX) crossover. Mutation is then probabilistically performed such as swapping or inserting cities or inverting a set.

Both ACO and GAs have disadvantages. ACO in terms of a large pheromone matrix memory overhead and computational cost of simulating ants. A GA in terms of using only two parents rather than whole population to generate two new solutions and also no domain specific heuristic information. A recent ACO variant, Partial-ACO [1], has two differences to ACO. First, no pheromone matrix instead using a population based approach (P-ACO) [3] with pheromone levels calculated based on the population of ant solutions. As pheromone cannot *build up* on edges of graph G, the pheromone deposit of ant k on an edge E of graph G is related to the quality of the given solution compared to the global best, g_{best}. Hence, ant k pheromone deposit, $\Delta\tau_{ij}^k$, is defined by:

$$\Delta\tau_{ij}^k = \begin{cases} (g_{best}/l_{best}^k)^\alpha, & \text{if edge } (i,j) \text{ belongs to } T^k \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

where (g_{best}/l_{best}^k) is the quality of ant k's local best solution in relation to the global best and α a parameter controlling pheromone influence. An ant k at each decision point reconstructs pheromone levels on edges from its current location to those unvisited.

The second difference with Partial-ACO is that ants only *partially* modify their own best found solution, l_{best}, in the same probabilistic manner as ACO. A random section is selected in the l_{best} solution and retained. The remaining part of the solution is reconstructed probabilistically as previously described (see Figure 1).

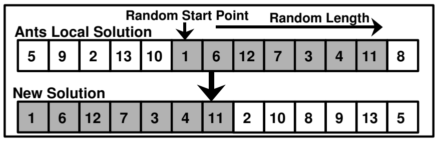

Figure 1: An illustration of the *Partial-ACO* methodology.

Table 1: GA and Partial-ACO mutation parameters

GA Parameters		Partial-ACO Mutation Parameters	
Population Size -	128	α -	10.0
Maximum Iterations -	100k	β -	4.0
Mutation Probability -	33%	τ_{init} -	0.5
Tournament Size -	3		

Table 2: Average run-times and error from optimal solutions when using standard GA crossover and mutation operators

TSP Instance		Crossover Operator		
		CX	OX	PMX
lin318	Error (%)	5.13 ± 1.20	11.24 ± 2.12	5.84 ± 1.39
	Time (secs)	99.83 ± 2.98	90.64 ± 2.44	90.88 ± 2.28
pcb442	Error (%)	5.83 ± 1.22	19.70 ± 2.34	6.35 ± 1.52
	Time (secs)	99.54 ± 2.08	93.91 ± 2.42	94.56 ± 1.83
rat783	Error (%)	9.32 ± 0.71	31.82 ± 1.21	8.87 ± 0.69
	Time (secs)	107.86 ± 1.49	107.92 ± 1.78	110.89 ± 0.95
pr1002	Error (%)	12.12 ± 1.02	35.67 ± 1.23	10.63 ± 1.07
	Time (secs)	116.43 ± 0.95	112.45 ± 0.54	117.02 ± 1.43
fl1400	Error (%)	14.22 ± 2.17	31.22 ± 2.12	11.24 ± 2.22
	Time (secs)	123.86 ± 0.40	124.56 ± 0.31	131.99 ± 2.80
u2192	Error (%)	31.09 ± 1.15	47.97 ± 1.13	25.80 ± 1.33
	Time (secs)	155.40 ± 0.55	156.10 ± 0.44	194.83 ± 5.10
pr2392	Error (%)	30.50 ± 1.01	45.74 ± 1.04	25.07 ± 1.08
	Time (secs)	165.86 ± 0.86	167.84 ± 0.55	224.33 ± 6.76
fl3975	Error (%)	45.34 ± 1.85	56.57 ± 1.78	39.62 ± 3.51
	Time (secs)	250.48 ± 0.79	249.69 ± 1.38	387.73 ± 27.20

Given the population based nature of Partial-ACO it can equally use a population of solutions generated by a GA. Consequently, the Partial-ACO methodology could be utilised as a mutation operator within a GA. Indeed, the *partial* modification of a solution can be described as similar to mutation. Hence Partial-ACO mutation selects two points in a solution similar to crossover. However, in this case all the cites between these points are marked as unvisited. An ant then reconstructs this section starting from the city prior to the first crossover point making choices using pheromone constructed from the population of GA solutions and heuristic information.

3 RESULTS

To measure the effectiveness of the Partial-ACO GA mutation operator it will be tested against eight TSP instances from the TSPLIB library. To provide a baseline to compare Partial-ACO mutation several standard crossover and mutation operators will be tested first, crossover operators OX, CX and PMX with swap, inversion and insertion mutation. Experiments were conducted over 25 random runs using a parallel implementation executing on an AMD Ryzen 2700 processor. Table 1 provides the GA parameters used.

The baseline results are shown in Table 2 whereby it can be observed that for small problems error is only a few percent from optimal but for larger problems considerable performance loss occurs.

Table 3: Average run-times and error from optimal solutions when using additional Partial-ACO mutation within a GA

TSP Instance		Crossover Operator		
		CX	OX	PMX
lin318	Error (%)	2.49 ± 0.61	2.88 ± 0.79	2.98 ± 0.98
	Time (secs)	112.22 ± 1.75	112.65 ± 1.58	112.94 ± 1.56
pcb442	Error (%)	2.83 ± 0.68	6.05 ± 1.35	3.05 ± 0.93
	Time (secs)	126.37 ± 1.27	124.54 ± 0.73	125.10 ± 0.95
rat783	Error (%)	3.53 ± 0.52	9.60 ± 1.77	4.07 ± 0.72
	Time (secs)	191.09 ± 0.89	187.92 ± 0.53	193.12 ± 1.68
pr1002	Error (%)	3.71 ± 0.52	10.77 ± 1.57	4.16 ± 0.78
	Time (secs)	239.97 ± 0.81	239.02 ± 0.48	243.05 ± 2.43
fl1400	Error (%)	3.82 ± 1.50	8.86 ± 1.92	5.12 ± 1.72
	Time (secs)	354.27 ± 3.02	357.01 ± 0.90	360.21 ± 4.84
u2192	Error (%)	5.17 ± 0.73	14.37 ± 1.35	6.44 ± 1.37
	Time (secs)	621.97 ± 2.64	624.81 ± 1.16	640.35 ± 11.17
pr2392	Error (%)	5.40 ± 0.70	17.14 ± 1.31	6.05 ± 0.62
	Time (secs)	713.06 ± 1.61	714.87 ± 1.23	728.75 ± 5.21
fl3975	Error (%)	4.96 ± 1.91	12.67 ± 1.21	6.99 ± 2.76
	Time (secs)	1486.92 ± 8.36	1483.63 ± 2.72	1511.48 ± 37.20

Local search such as Edge Assembly (EAX) or 2-opt would improve results but for these experiments a pure evolutionary approach is desired. The PMX crossover operator offers the best performance.

To test the effectiveness of Partial-ACO mutation within a GA it is added as a fourth mutation operator. The Partial-ACO parameters used are described in Table 1. The results from a GA with Partial-ACO mutation are shown in Table 3. Contrasting to Table 2 it is clear that a significant performance gain has been achieved especially for larger problems with up to a ten fold reduction in relative error. CX crossover demonstrates the best results when using Partial-ACO mutation. However, it should be noted that Partial-ACO mutation incurs a computational cost especially as problem sizes increase, a quadratic increase. This is due to quadratically greater edges for ants to consider at each step as noted in [1].

4 CONCLUSIONS

The performance of a GA applied to TSP instances is significantly improved using the population based ACO technique Partial-ACO as a mutation operator. The GA population provides pheromone influence for simulated ants *partially* mutating solutions combining Darwinian principles with ant search. This enables considerable error reduction compared to standard GAs with only small errors from optimality *without* resorting to local search. These gains are due to ants considering all GA solutions via pheromone and heuristic information which GAs do not. However, simulating ants incurs a large computational cost. Further work will investigate methods to reduce this cost, other crossover operators and local search.

REFERENCES

[1] Darren M Chitty. 2017. Applying ACO to Large Scale TSP Instances. In *UK Workshop on Computational Intelligence*. Springer, 104–118.
[2] 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.
[3] Michael Guntsch and Martin Middendorf. 2002. A population based approach for ACO. In *Workshops on Applications of Evolutionary Computation*. Springer, 72–81.
[4] John H Holland. 1975. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.* U Michigan Press.