MA-ABC: A Memetic Algorithm Optimizing Attractiveness, Balance, and Cost for Capacitated Arc Routing Problems

Muhilan Ramamoorthy
Computing, Informatics, and Decision Systems Engineering
Tempe, Arizona
mramamoo@asu.edu

Stephanie Forrest
Biodesign Center for Biocomputing, Security and Society
Tempe, Arizona
steph@asu.edu

Violet R. Syrotiuk
Computing, Informatics, and Decision Systems Engineering
Tempe, Arizona
syrotiuk@asu.edu

ABSTRACT

Services such as garbage collection, road gritting, street sweeping, and power line inspection can each be formulated as a *capacitated arc routing problem* (CARP). The traditional formulation of CARP has the goal of minimizing the total cost of the routes making up a solution. Recently, operators of such services require routes that are balanced and visually attractive in addition to low cost. Routes that are *balanced* are about equal in length and provide fair work assignments. *Visually attractive* routes are subjective, but they usually involve non-crossing routes that provide well defined service areas. These additional features are important because they address operational complexities that arise from using the routes in practice. This paper presents MA-ABC, a memetic algorithm to find solutions for CARP that maximize route attractiveness and balance, while minimizing total cost. A novel fitness function combines route overlap with route contiguity to assess route attractiveness. MA-ABC is the first to incorporate attractiveness in a three-objective search for heuristic solutions for CARP. Experimental results on CARP benchmark instances show that MA-ABC finds a diverse set of heuristic solutions at the Pareto front, providing a wide choice for service operators to tradeoff design objectives.

CCS CONCEPTS

• **Applied computing** → **Transportation**; • **Computing methodologies** → *Heuristic function construction.*

KEYWORDS

Genetic algorithm, Evolutionary computation, Memetic algorithm, Multi-objective optimization, Capacitated arc routing problem (CARP), Visual attractiveness.

ACM Reference Format:
Muhilan Ramamoorthy, Stephanie Forrest, and Violet R. Syrotiuk. 2021. MA-ABC: A Memetic Algorithm Optimizing Attractiveness, Balance, and Cost for Capacitated Arc Routing Problems. In *2021 Genetic and Evolutionary Computation Conference (GECCO '21), July 10–14, 2021, Lille, France.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3449639.3459268

GECCO '21, July 10–14, 2021, Lille, France

ACM ISBN 978-1-4503-8350-9/21/07…$15.00
https://doi.org/10.1145/3449639.3459268

1 INTRODUCTION

Routing problems are among the most widely researched topics in operations research. In arc routing the goal is to cover, i.e., visit, each edge of the graph representing the problem. Applications of arc routing include services such as street sweeping, snow plowing, mail delivery, garbage collection, gritting roads with sand or salt, and the inspection of power lines. Expenditures on such services by public and private entities exceed billions of dollars annually in the U.S., emphasizing their economic importance [10, 11, 13, 16].

The *capacitated arc routing problem* (CARP) [21] is defined on a weighted undirected graph $G = (V, E)$. The streets in a city correspond to the edges E in G, and the vertices V to their intersections. There is a depot $D \in V$ used to store a fleet of k homogeneous vehicles, each with capacity C. *Tasks* correspond to a service required on a subset $T \subseteq E$ of the streets. Each task has a traversal cost $c(t)$ and a demand $d(t)$. Streets not requiring service have zero demand. All edges can be traversed any number of times. The goal of CARP is to find a set of closed routes, one for each vehicle, starting and ending at the depot, of minimum total cost such that: All tasks are serviced, the sum of the demands of serviced edges of each route does not exceed the vehicle capacity, and every serviced edge is in exactly one route. The cost of a route corresponds to the cost of its serviced edges and the cost of *deadheading*, i.e., the traversal cost of any intermediate connecting paths.

CARP is NP-hard [21]. As a result, an optimum solution for most practical instance sizes is intractable. Therefore a number of heuristics and meta-heuristics have been proposed [42]. Many focus on finding a least cost solution for a service. Recently, operators of such services require routes that are balanced and visually attractive in addition to low cost. Routes that are *balanced* are about equal in length and contribute to workload equity and employee satisfaction [34, 35]. The *visual attractiveness* of routes is subjective, but non-crossing routes that provide well-defined service areas are clearly preferred. These additional objectives are important because they help address operational complexities associatd with using the routes in practice.

Because it is often impossible to optimize all objectives simultaneously, a solution is called Pareto optimal if no objective can be improved without deteriorating another. *Multi-objective evolutionary algorithms* (MOEAs) use *evolutionary computation* (EC) methods to search for solutions at the Pareto front. These offer service operators a diverse set of solutions that represent different tradeoffs among design objectives.

This paper presents MA-ABC, a memetic algorithm to provide a heuristic solution for CARP to maximize route attractiveness and balance, and to minimize total cost. A memetic algorithm is

a hybrid MOEA that incorporates local search. In MA-ABC local search controls the scope of the diversification emphasizing total cost. Non-dominated sorting is used to sort solutions into different ranked fronts. Within the same front, solutions are ranked by crowding distance. We use fitness functions for total cost and for balance, and introduce route continuity and route overlap to assess route attractiveness. To the best of our knowledge, MA-ABC is the first multi-objective optimization approach for CARP that includes attractiveness as one of the objectives. We conduct a thorough experimental evaluation of MA-ABC on CARP benchmark instances, comparing to the *path scanning with random task* (PSRT) heuristic [3]. MA-ABC provides heuristic solutions at the Pareto front that have a wide diversity in attractiveness and balance without deviating in the objective space of total cost.

The rest of the paper is organized as follows. We review related work in §2. The design of MA-ABC is provided in §3. A description of the benchmark instances, experimental set-up, and an analysis of the results are provided in §4, followed by a discussion in §5. Finally, conclusions and future work are found in §6.

2 RELATED WORK

Since the introduction of CARP by Golden and Wong [21], many variants have emerged; see [10] for a comprehensive presentation of applications, resulting CARP variants, and solution methods.

Poot et al. [41] first reported that operators of services considered the heuristic solutions generated by the ORTEC vehicle routing software [40] to be poor, despite ranking highly on the traditional metric of total cost. The reason given was that the routes were not visually attractive. Operators desired routes with more subjective features such as compactness, no crossings, and fewer turns, even at the expense of cost [8]. Operationally, this assigns a route in which each driver has one geographically distinct area of responsibility. Visual preferences may serve as a source for new metrics that can help quantify the visual appeal of a route; Rossit et al. [46] review proposed metrics. Such new metrics may be used as alternatives to, or penalties in, the overall objective function.

A few publications consider visual attractiveness in variants of arc routing problems. Constantino et al. [8] propose a method to bound the number of nodes in more than one route. In partitioning street networks, Lum et al. [32] define a similar *route overlapping index* (ROI) and develop a heuristic for an *uncapacitated* arc routing problem with compact, balanced, and visually appealing routes. Corberan et al. [9] provide a heuristic solution to the same problem also using a multi-objective approach.

Route balance is another important objective [1] because it contributes to workload equity [34, 35]. Measures of balance have included *range*, the difference between the maximum and minimum route length, and *makespan*, the maximum length route. Several works consider the dual objectives of route balance and total cost in variants of routing problems [15, 25, 26].

Multi-objective evolutionary algorithms (MOEAs) are one of the most popular approaches for solving multi-objective optimization problems to obtain near Pareto optimal solutions [6]. Deb et al. [12] propose NSGA-II, a widely used method of MOEA due to its efficiency. It uses non-dominated sorting to obtain the Pareto fronts and for selection operations. NSGA-II's sorting method is more efficient than other popular MOEA methods such as SPEA [7].

Lacomme et al. [28–30] propose a memetic algorithm based on the *route first, split second* method to solve CARP. A single *giant tour* covering all required tasks without capacity limits is used to represent individuals. This helps avoid the need to repair routes after crossover, due to capacity overruns. Their method is effective because it replaces the mutation operation with a local search, and this approach has since been adopted for many variants of arc and node routing problems [43, 44]. An efficient splitting algorithm that splits the giant tour is used to evaluate and update the fitness values and to retrieve the final routing solution from the chromosomes.

Other multi-objective optimization methods have been applied to CARP. Mei et al. [38] use a decomposition-based framework for CARP with the dual objectives of minimizing total cost and makespan. Grandinetti et al. [22] use an ϵ-constraint method for CARP with the same objectives.

Tang et al. [50] use a giant tour and introduce a new *merge-split* operator, in addition to other move operators in the local search. This operator merges multiple routes back into a single tour and then splits it again.

Usberti et al. [2] use a memetic algorithm for a CARP variant. Three types of local search are combined with a stochastic filter to filter out solutions before applying the local search. Chen et al. [5] adopt a hybrid approach for CARP, performing only a single solution update per generation with local refinement. Martinez et al. [33] propose a genetic algorithm for CARP which incorporates local search. To solve larger-scale (> 300 tasks) CARP, Mei et al. use decomposition methods and co-operative co-evolution methods [36]. Stochastic variants of CARP have used memetic algorithms [17, 37, 54]. Wang et al. use an *estimation of distribution algorithm* and a stochastic local search for a stochastic variant of CARP [53]. Handa et al. use an EC method for dynamic route optimization in CARP [23, 24]. Liu et al. [31] use a memetic algorithm with a new splitting scheme to solve dynamic CARP that was improved in [47]. Shang et al. consider a dual objective CARP through a co-operative co-evolutionary method [48].

As we see next, our proposed memetic heuristic algorithm makes use of ideas incorporated in NGSA-II [12] and in the route first, split second method [28–30], but we believe it is the first to include attractiveness as an objective in CARP.

3 MULTI-OBJECTIVE MEMETIC ALGORITHM

We propose MA-ABC, a memetic heuristic algorithm for CARP, which maximizes route attractiveness and balance, while minimizing total cost.

MA-ABC pseudocode is given in Algorithm 1. An initial population of μ individuals is generated. Each individual is a *giant tour*. Giant tours can be generated using any heuristic for CARP [19, 42] by ignoring demands and capacity limits; we use the path scanning heuristic [20].

In each of $NGEN$ generations, λ offspring are generated. Parents are chosen using *tournament selection* in NSGA-II with the ranking determined by dominance and crowding distance. With probability $CXPB$, an order crossover operation is performed. With probability $MUTPB$, a giant tour is split into routes for each of the vehicles.

A local search is then performed using a single insertion move operator that emphasizes total cost. The routes are then joined to form a new individual.

After λ offspring have been generated, the fitness metrics of attractiveness, balance, and cost are updated for each. Elitism is incorporated through the $(\mu+\lambda)$ strategy [12]. After *NGEN* generations, we obtain the Pareto front ranked by non-dominated sorting. We now describe these steps in a more detail.

Algorithm 1: MA-ABC Algorithm

Input : Graph model $G = (V, E)$ for CARP
Output: Heuristic routing solutions maximizing balance and attractiveness, and minimizing total cost

```
set parameters NGEN, CXPB, MUTPB, μ, λ
create initial population of size μ
for i ← 1 to NGEN do
    for j ← 1 to λ do
        P1, P2 = TournamentSelectionDCD ()
        if prob < CXPB then
            C1 = OrderCrossover (P1,P2)
        else
            C1 = P1
        if prob < MUTPB then
            split = SplitGiantTour (C1)
            split = MutateByLocalSearch (split)
            C1 = JoinRoutes (split)
    update three fitness values for each of the λ offspring
    population =selectNSGA (μ, μ + λ)
Pareto_front = NonDominatedSort (population, μ)
return Pareto_front
```

3.1 Selection using NSGA-II

MA-ABC selects individuals for the crossover operation. It first selects μ of $\mu + \lambda$ individuals from the current generation. It also selects individuals at the Pareto front to return as routing solutions. MA-ABC uses NSGA-II specifically its non-dominated sorting and crowding distance for ranking solutions.

NSGA-II sorts the solutions into different ranked fronts based on *non-dominance* [25, 26]. NSGA-II first finds all non-dominated solutions among the population in the current iteration. These solutions are then removed from the population and the non-dominated individuals among the remaining population are found. The process is repeated until all the solutions are ranked.

Within the same front, solutions are ranked based on *crowding distance* [12]. A solution with higher crowding distance on the same front indicates higher population diversity and is ranked more highly.

3.2 Crossover

Many crossover operators have been studied in EC, and among these, order crossover (OX) is effective for routing problems [39]. In OX, two sites p and q are randomly selected with $1 \le p \le q \le \tau$, where $\tau = |T|$. For parents P_1, P_2, child C_1 is obtained by copying tasks $P_1(p) \dots P_1(q)$ into $C_1(p) \dots C_1(q)$. $C_1(q+1) \dots C_1(\tau)$ and $C_1(1) \dots C_1(p-1)$ are filled by copying tasks from $P_2(q+1) \dots P_2(\tau)$ and $P_2(1) \dots P_2(q-1)$, taking tasks from P_2 not already in C_1. Child C_2 is created by interchanging the roles of P_1 and P_2.

3.3 Splitting Procedure

The splitting procedure plays a central role when a giant tour is used as an individual chromosome. Splitting is based on the *route first, split second* algorithm [28, 51]; see §2. The idea is to split the giant tour into routes based on the vehicle capacity limit C. Each route starts at the depot D, and deadheads to D from the vertex where the capacity is reached or prior to it being exceeded. We use this procedure to split the giant tour when reevaluating fitness values, before applying local search, and in retrieving the routes.

3.4 Local Search

As in Lacomme et al. [28], MA-ABC uses local search in place of mutation to improve exploitation of the search space and speed convergence. Once a giant tour has been split into routes, a *single insertion* move operator is applied to each pair (t, t') of tasks in the routing solution. Each task t in a route is moved after task t' in every other route. The move that minimizes the total cost is chosen.

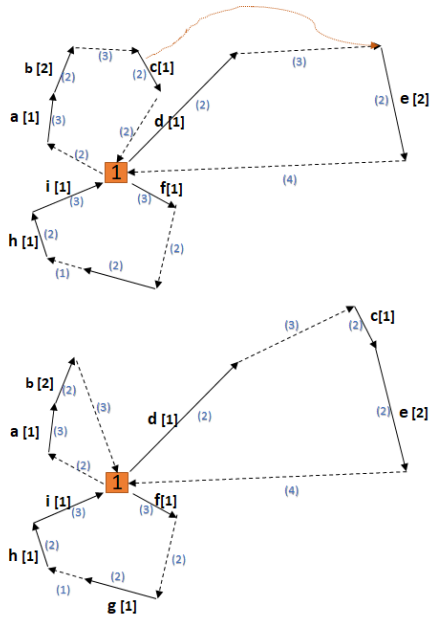


Figure 1: Example of the single insertion move operator.

Figure 1 gives an example of a single insertion move. In this example, the red square is the depot, solid edges are tasks, dashed edges are not required, vehicle capacity $C = 4$, integers in square brackets are demands, and integers in parentheses are costs.

The three routes before the move are $((a, b, c)(d, e)(f, g, h, i))$ with costs $(14, 12, 13)$ that sum to a total of 39. After moving c after d the routes are $((a, b)(d, c, e)(f, g, h, i))$ with costs of $(10, 14, 13)$ that sum to 37, a cost savings of 2.

3.5 Fitness Functions

Let $R = \{r_1, r_2, \ldots, r_k\}$ be a set of routes for the k vehicles found by splitting a giant tour. Each route r_i is a sequence of streets (tasks) serviced $t_{i1} = (v_{i1}, v_{i2}), t_{i2} = (v_{i2}, v_{i3}), \ldots, t_{ij-1} = (v_{ij-1}, v_{ij})$, that make a closed walk starting and ending at the depot.

The *total cost* is the sum of the route cost for all routes in R. The cost of a route is the cost of its serviced tasks including any deadheading via shortest paths.

$$\text{total cost} = \sum_{r_i \in R} \left(\sum_{t \in r_i} c(t) + \sum_{\ell=1}^{j-1} DeadHeading(v_{i\ell}, v_{i\ell+1}) \right) \quad (1)$$

We use *makespan* as a proxy for route balance. The makespan is the maximum route cost among the routes $r_i \in R$.

A novel contribution of this paper is our definition of route attractiveness, a combination of two metrics: The *common node count* (CNC), a measure of route overlap, and the *discontinuity count* (DC), a measure of route contiguity. That is, the preference is for routes of different vehicles to have no overlap, and for sequences of tasks in a route not to be disconnected.

More formally, CNC is a count of vertices $v \in |V \setminus D|$ that exist in each route $r_i \in R$ with one endpoint of a task serviced by r_i.

$$CNC = \sum_{v \in |V \setminus D|} \left(\sum_{i=1}^{k} exists(v, r_i) - |V| - 1 \right) \quad (2)$$

This definition differs from the node count used in [8] because we count the nodes that only belong to the tasks serviced by the route. Nodes that belong to the tasks that are not serviced by the route or nodes that belong to non-required edges are not counted. We also count each endpoint of a serviced task unlike [8].

Given two consecutive tasks $t_{i\ell}, t_{i\ell+1}$ serviced by route r_i it is possible that the tasks are not contiguous, i.e., the vehicle must deadhead from the end of $t_{i\ell}$ to the start of $t_{i\ell+1}$. If any edge along the deadheading path has *at least one task that is serviced by another route*, we consider it a *discontinuity* in route r_i and increment DC.

$$DC = \sum_{r_i \in R} \left(\sum_{\ell=1}^{j-1} discontinuity(t_{i\ell}, t_{i\ell+1}) \right) \quad (3)$$

Both $exists(\cdot)$ and $discontinuity(\cdot)$ are simple indicator functions.

If the discontinuity count is less than $k = |R|$ then we set the *attractiveness* equal to the CNC, otherwise we set it to infinity:

$$\text{attractiveness} = \begin{cases} \text{CNC}, & \text{if } DC < k \\ \infty, & \text{if } DC \geq k \end{cases} \quad (4)$$

This definition of attractiveness forces MA-ABC to reject any solution that has high discontinuity in the selection of parents for offspring of a generation, and the selection of the next generation.

3.6 Elitism

Elitism is essential for the convergence of MOEAs [6]. We incorporated elitism in MA-ABC as implemented in NGSA-II [12].

4 EVALUATION OF MA-ABC

MA-ABC was coded in `python`; source code is provided [49] for reproducibility. We use the DEAP [18] EC framework, which includes an implementation of NSGA-II ranking and selection operations.

We evaluate two classic benchmark CARP instances: `val`, and `egl`. `val` [4] is a collection of 34 instances based on ten randomly generated graphs with 24-50 vertices and 34-97 edges, where all edges are required. Instances of the same graph differ by vehicle capacity. `egl` [14] is a collection of 24 instances generated based on a winter gritting application in Lancashire county in the U.K. It contains two graphs, one with 77 vertices and 98 edges (`egl-s`) and the other with 140 vertices and 190 edges (`egl-e`). Each graph consists of four sets of three instances each with the sets differing by number of required edges. The instances within a set (named with suffix A, B, C) differ by capacity limit of the vehicles.

We report the results of MA-ABC, using the parameter settings in Table 1, from a single run to show that our algorithm is efficient, i.e., that operators can use MA-ABC in a real-world scenario where solutions need to be generated in real time such as in a vehicle breakdown scenario [45]. For completeness, we also present a statistical analysis of 30 runs of our algorithm in §4.3.

Table 1: MA-ABC parameter settings

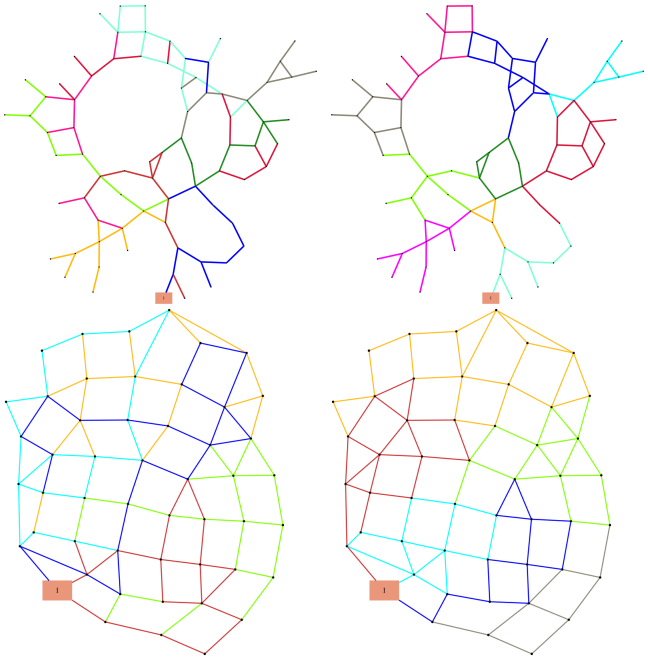
Parameter	Value
Number of generations ($NGEN$)	300
Population size (μ)	100
Offspring produced (λ)	100
Crossover rate ($CXPB$)	1
Mutation rate ($MUTPB$)	1

4.1 Results

We compare the results of MA-ABC with the solution generated by 1000 iterations of the *path scanning with random task* (PSRT) heuristic. PSRT was chosen because it seeks to optimize total cost. We present results for the objectives of attractiveness, total cost, and balance, in turn. Due to space limitations, see [49] for a presentation of all results, including the data used to generate figures.

4.1.1 Attractiveness. We first provide a qualitative evaluation of MA-ABC for a few CARP instances. Figure 2 shows a visualization of the routes generated by PSRT on the left and those produced by MA-ABC on the right, using a different colour for each route, for the `egl-e4-A` and `val10C` instances, respectively. As the figure shows, the routes generated by MA-ABC are compact, do not cross, and have better defined service areas; therefore they considered more attractive than the routes generated by PSRT.

We also provide a quantitative comparison of the algorithms using three metrics: The *connectivity index* (CI), the *average task distance* (ATD), and the *route overlapping index* (ROI) [8]. CI measures the average number of connected components within a route. ATD is a measure of average deadheading cost between task pairs within the same route. ROI measures the node overlap of the current solution with the overlapping of an "ideal" solution. For each metric, smaller is considered more attractive.



(a) PSRT (b) MA-ABC

Figure 2: Heuristic routing solution produced by PSRT (left) and MA-ABC (right) on two instances: `egl-e4-A` (top row) and `val10C` (bottom row). The rectangle represents the depot. Vehicles on some routes need to deadhead on the shortest path from the depot to the first task on the route.

We compute these three metrics for the most attractive solution in the Pareto front found by MA-ABC and compare with PSRT. Table 2 shows the results for the `egl` instances. For each metric the smaller value is shown in bold. We see that CI and ROI for MA-ABC are lower than those of PSRT for all instances. ATD is also lower for MA-ABC for all but six instances. Hence our routing solutions are considered attractive by external metrics not used in the algorithm.

4.1.2 Total Cost. The first row of Figure 3 plots the total cost of `egl` (a) and `val` (b) instances, produced by MA-ABC and PSRT; the optimal or current best known total cost solution is also plotted.

In comparison to PSRT, MA-ABC found a lower total cost solution for all 24 instances of `egl`. The percentage of reduction ranges from 8.81-19.64% with an average reduction of 11.45%. The total cost found by MA-ABC ranges from 0.36-6.18% higher than the optimum (or best known solution), with an average of only 3.31% higher. MA-ABC is able to come quite close to the optimum (or best known) routing solutions with respect to total cost.

4.1.3 Balance. We use makespan as a proxy for balance; see §3.5. The second row of Figure 3 plots the makespan of `egl` (a) and `val` (b) instances, produced by MA-ABC and PSRT. Maximizing balance corresponds to minimizing the makespan. The best makespan solutions of MA-ABC are smaller than those of PSRT for all benchmark instances. For `egl` the average reduction is 13.68%, but for `val` the average reduction is much larger, namely 28.27%.

Table 2: egl: Attractiveness metrics for PSRT v. MA-ABC

Instance	PSRT CI	PSRT ROI	PSRT ATD	MA-ABC CI	MA-ABC ROI	MA-ABC ATD
egl-e1-A	2.60	0.65	61.19	**2.17**	**0.54**	**43.14**
egl-e1-B	2.71	0.38	**45.35**	**2.29**	**0.38**	57.05
egl-e1-C	2.70	0.35	44.28	**1.30**	**0.35**	**23.40**
egl-e2-A	2.71	0.35	58.67	**2.00**	**0.28**	**58.22**
egl-e2-B	2.60	0.21	60.23	**1.64**	**0.20**	**43.44**
egl-e2-C	2.21	0.23	37.97	**1.40**	**0.22**	**28.91**
egl-e3-A	2.88	0.73	**67.31**	**2.30**	**0.61**	71.95
egl-e3-B	2.25	0.53	51.26	**1.64**	**0.47**	**32.64**
egl-e3-C	2.18	0.62	**33.40**	**1.83**	**0.59**	47.48
egl-e4-A	2.22	0.92	63.86	**1.92**	**0.73**	**67.48**
egl-e4-B	2.36	0.75	41.70	**1.75**	**0.68**	**40.56**
egl-e4-C	1.90	0.59	32.51	**1.29**	**0.57**	**26.79**
egl-s1-A	3.29	1.18	**83.33**	**2.57**	**1.18**	96.80
egl-s1-B	2.90	0.77	**76.66**	**2.18**	**0.71**	83.50
egl-s1-C	2.29	0.59	49.09	**1.60**	**0.56**	**45.60**
egl-s2-A	2.79	0.41	60.95	**2.18**	**0.36**	**44.17**
egl-s2-B	2.85	0.41	44.98	**1.79**	**0.36**	**36.71**
egl-s2-C	2.33	0.50	43.87	**2.03**	**0.48**	**29.16**
egl-s3-A	2.67	0.38	46.59	**1.94**	**0.36**	**35.94**
egl-s3-B	2.27	0.54	**41.67**	**1.96**	**0.51**	56.17
egl-s3-C	2.21	0.38	40.51	**1.44**	**0.36**	**19.95**
egl-s4-A	2.47	0.69	49.65	**1.43**	**0.65**	**39.04**
egl-s4-B	2.04	0.62	34.70	**1.38**	**0.59**	**23.53**
egl-s4-C	2.31	0.59	34.26	**1.28**	**0.54**	**19.97**

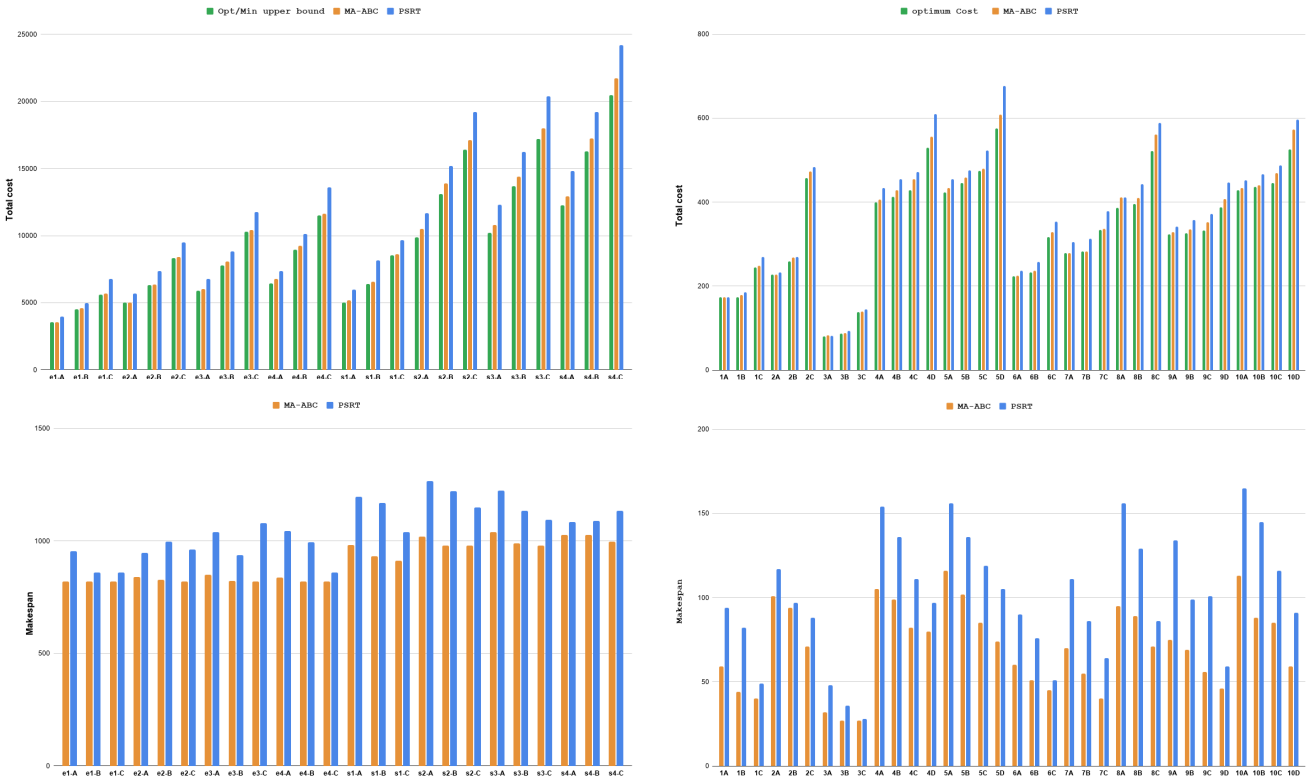
4.2 Pareto Efficiency, Spread, and Convergence

An MOEA that produces a diverse set of solutions at the Pareto front is useful for operators by providing a wide set of heuristic solutions to meet design objectives. The approximate Pareto front generated by MA-ABC for the instance `egl-e4-A` is shown in Figure 4 with total cost, attractiveness, and balance as axes. The figure shows that the Pareto front is wide and diverse, providing choice for operators.

To further illustrate the range of choice that MA-ABC offers, in Figure 5 we use a box-and-whiskers plot for each metric, for each heuristic solution on the Pareto front. In general, these plots show that the boxes for balance and attractiveness are larger than those of total cost. This is not surprising because our algorithm uses only the total cost for the move operator in the local search; this helps to achieve total cost values closer to optimum and maintain the heuristic solutions in the population closer to the optimum total cost (balance and attractiveness considered secondary objectives).

Early convergence is a concern in EC [27]. Convergence can be inferred from the change in the population from generation to generation. A figure that plots of the objective values as a function of generation is also helpful. For example, Figure 6 plots the total cost versus generations for the `egl-s4-C` instance. This figure shows that, towards the end of the number of generations, the total cost of the best heuristic solution flattened indicating that it has attained stability and is near convergence. At the same time, the mean fitness varies indicating that the method maintains good diversity.

We also computed the number of unique non-dominated solutions, number of unique dominated solutions, and the number of duplicate solutions in the final iteration of MA-ABC for `egl` and `val`. The number of unique non-dominated solutions ranges from 12-63 for the `egl` instances. For the `val` instances the range is from 7-60, except for `val1-val3` which range from 2-17 because the graph is smaller. The larger the number of unique non-dominated solutions, the higher the number of choices for an operator; it confirms that diversity is maintained and the early convergence is avoided.



(a) egl instances (b) val instances

Figure 3: Comparison of total cost (top); optimum/best known total cost results in green, MA-ABC in orange, and PSRT in blue. Comparison of makespan (bottom); MA-ABC results in orange, and PSRT in blue. Both objectives are minimized.

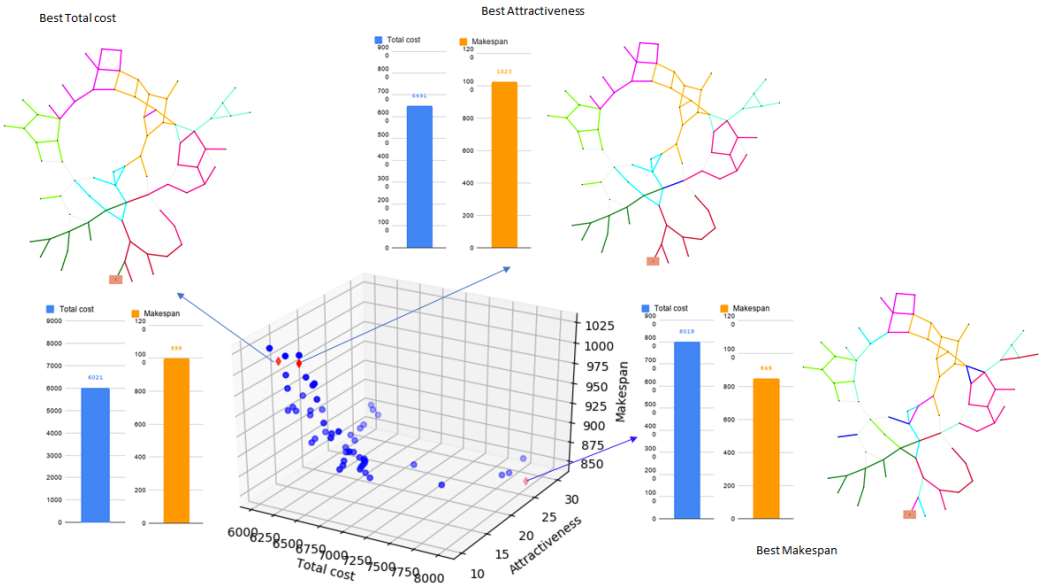
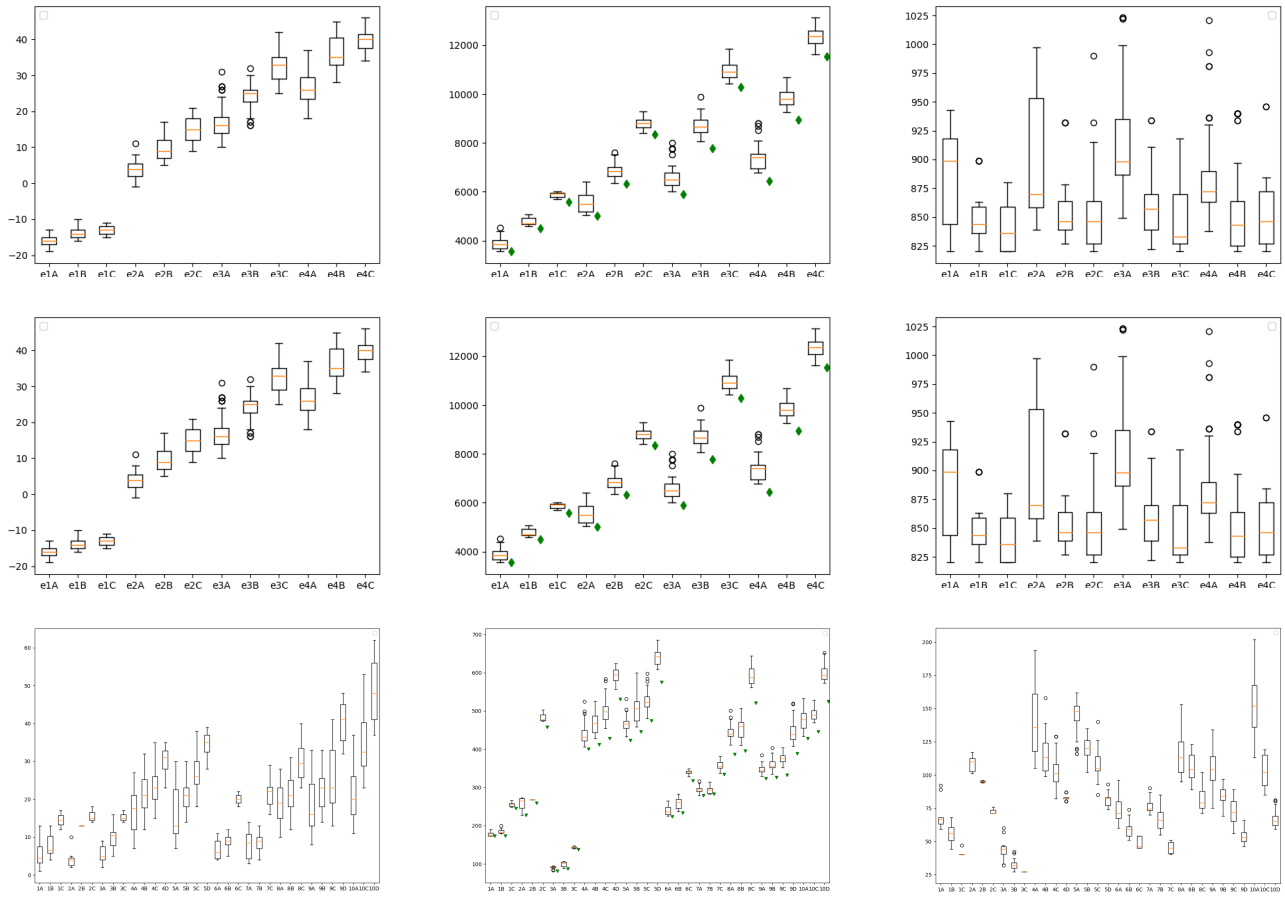


Figure 4: Approximate Pareto front (in blue) for `egl-e4-A`. A visualization of the routing solution, total cost and makespan are given for three solutions on the front: From left to right, the best heuristic solution with respect to total cost, attractiveness, and makespan.



(a) Attractiveness (b) Total cost (c) Balance

Figure 5: Box-and-whiskers plots illustrating spread of the heuristic solutions found by MA-ABC for (a) attractiveness, (b) total cost, and (c) balance for the `egl-e` (row 1), `egl-s` (row 2), and `val` (row 3) instances. The green diamond next to each instance of total cost indicates the optimum (or best known) solution.

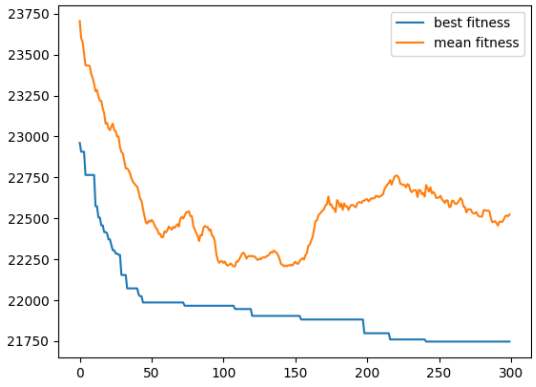


Figure 6: Total cost v. generations for `egl-s4-C`.

4.3 Statistical Analysis

All results presented so far are for a single run of MA-ABC. To measure the repeatability and precision of MA-ABC we use the *coefficient of variation* (CV). It is defined as the ratio of the standard deviation σ over the absolute mean μ. A CV value that is more than 100% is considered highly variant, while a CV less than 100% is considered to show low variance.

Table 3 presents μ, σ^2, and CV for each of total cost, attractiveness, and makespan, computed from 30 runs of MA-ABC. The table shows that the CV is greater than 100% for only one instance in `egl-e` and none in `egl-s`. All `val` instances (see [49]), have CV less than 100%; 10 had more than 33% and only four were more than 50%. This suggests that MA-ABC gives consistent results.

Table 3: Mean, variance, and CV for egl instances

Instance	Total cost			Atttractiveness			Makespan		
	μ	σ^2	$\frac{\sigma}{\lvert\mu\rvert}$	μ	σ^2	$\frac{\sigma}{\lvert\mu\rvert}$	μ	σ^2	$\frac{\sigma}{\lvert\mu\rvert}$
egl-e1-A	3589.73	3988.20	1.76	-19.00	0.69	4.37	819.77	1.63	0.16
egl-e1-B	4566.00	618.90	0.54	-16.63	0.52	4.32	815.77	115.77	1.32
egl-e1-C	5725.57	2497.84	0.87	-14.20	0.51	5.03	799.03	425.90	2.58
egl-e2-A	5122.97	1943.76	0.86	-0.30	2.15	488.57	837.63	124.65	1.33
egl-e2-B	6437.53	1848.53	0.67	3.70	1.11	28.52	826.77	16.94	0.50
egl-e2-C	8529.60	3400.25	0.68	10.27	0.41	6.23	818.40	12.52	0.43
egl-e3-A	6048.33	2804.02	0.88	10.70	1.11	9.86	856.70	87.04	1.09
egl-e3-B	8033.47	5288.26	0.91	17.63	1.14	6.05	826.67	29.26	0.65
egl-e3-C	10422.67	8846.16	0.90	25.37	1.27	4.45	819.73	0.75	0.11
egl-e4-A	6716.30	3202.15	0.84	18.13	0.67	4.52	854.30	150.56	1.44
egl-e4-B	9404.47	6731.50	0.87	28.33	1.61	4.48	822.20	4.65	0.26
egl-e4-C	11775.20	16557.13	1.09	35.33	2.16	4.16	819.80	0.37	0.07
egl-s1-A	5168.37	154.17	0.24	-54.67	0.37	1.11	953.83	260.76	1.69
egl-s1-B	6585.53	1697.57	0.63	-52.00	0.69	1.60	922.70	18.08	0.46
egl-s1-C	8570.50	2717.71	0.61	-47.33	0.37	1.28	914.00	20.69	0.50
egl-s2-A	10454.63	7436.65	0.82	11.07	2.89	15.37	1030.53	187.64	1.33
egl-s2-B	13749.00	15646.07	0.91	21.90	4.58	9.77	992.87	61.64	0.79
egl-s2-C	17058.07	18664.41	0.80	31.17	3.25	5.78	978.70	0.22	0.05
egl-s3-A	10684.20	5807.27	0.71	18.60	3.35	9.84	1031.23	258.74	1.56
egl-s3-B	14368.47	14492.81	0.84	33.20	3.96	5.99	989.40	66.04	0.82
egl-s3-C	17979.63	20183.76	0.79	40.00	3.72	4.82	979.00	4.55	0.22
egl-s4-A	13169.83	13510.83	0.88	52.90	9.33	t 5.78	1031.57	36.19	0.58
egl-s4-B	17076.97	20326.93	0.83	66.77	5.56	3.53	1026.80	0.58	0.07
egl-s4-C	21500.30	41947.25	0.95	80.33	4.71	2.70	1018.50	173.02	1.29

4.4 Run Time Performance

MA-ABC was run on a desktop system with Intel Core i7-4770S CPU @ 3.10 GHz × 8, with 7.7 GiB of memory, running the Ubuntu 18.04 LTS operating system. Figures 7 shows the running time for egl instances, plotted as a function of instance size, i.e., the number of required tasks. The running time for egl instances ranges from 123.60-1291.12 seconds. As the figure shows, the run time is approximately linear. This indicates that MA-ABC appears to scale well and may be appropriate for real-world instances.

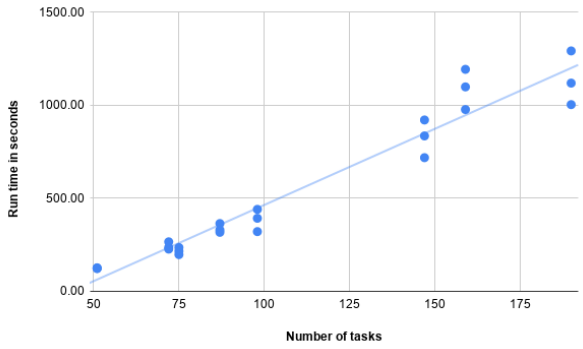


Figure 7: Run time v. instance size for egl instances.

5 DISCUSSION

The purpose of the current work is to show that the attractiveness of a heuristic routing solution may be improved using EC algorithms, without compromising total cost. This provides operators of services broad and diverse choice among heuristic solutions to meet their design objectives. Furthermore, MA-ABC produces these heuristic solutions with only one run which is important in case of service interruption which requires a real time response, e.g., rerouting in the event of a vehicle breakdown.

Our new metric of attractiveness could find other application domains, e.g., in node routing problems such as in air route networks, among others.

We want to keep MA-ABC simple to enable easy adoption and reproducibility. Hence no parameter tuning methods are currently integrated into MA-ABC. To enable MA-ABC to be more general so that it can perform well across more types of instances, some form of automatic tuning can be integrated. Recall that the source code is provided at [49].

We also did not use pool update or population restart strategies in order to keep our design simple. The current population size of 100 and number of generations of 300 can be reduced by using some form of pool update strategy which may help in reducing the running time for large-sized instances.

We implemented local search based only on the total cost objective to keep the solutions close to the optimum total cost; this is because total cost is generally considered of high importance.

We used only one move operator as the local search is expensive and we want to limit the running time. Multi-objective local search and sophisticated move operators such as K-opt may help to improve solution quality in real-world instances in spite of an increase in computation time. The computation time of local search may be reduced by using acceleration mechanisms and by using statistical filters [52].

6 CONCLUSIONS AND FUTURE WORK

In this paper we developed MA-ABC, a multi-objective EC algorithm based on NSGA-II for CARP with three objectives: To seek to minimize total route cost and balance, and to maximize route attractiveness. We defined a novel fitness function for attractiveness combining a measure of route overlap and a measure of route contiguity. We evaluated MA-ABC on two benchmark CARP instances. The resulting heuristic solutions are not only visually attractive but also obtain a Pareto front that is diverse with broad choices for the attractiveness and makespan objectives without deviating much from the best total cost objective.

As future work, we plan to evaluate MA-ABC on larger, more realistic, data sets in order to understand the implications to running time. Comparison to other heuristics that seek to optimize total cost and balance, not just total cost as in PSRT, is also planned. Incorporating either more or different metrics for attractiveness are worthy of study. In addition, we plan to implement automatic tuning of parameters for each instance so as to improve the accuracy and speed. We will also include additional objectives, such as minimizing the number of vehicles, considering the use of a *many objective optimization* method such as in NSGA-III.

ACKNOWLEDGMENTS

We thank the anonymous reviewers whose useful comments have improved the quality of our paper.

SF was supported in part by NSF (CCF 1908633), DARPA (FA8750-19C-0003, N6600120C4020), AFRL (FA8750-19-1-0501), and the Santa Fe Institute. VRS was supported in part by NSF NeTS 1813451.

REFERENCES

[1] David Applegate, William Cook, Sanjeeb Dash, and André Rohe. 2002. Solution of a min-max vehicle routing problem. *INFORMS Journal on computing* 14, 2 (2002), 132–143.

[2] Rafael Kendy Arakaki and Fábio Luiz Usberti. 2018. Hybrid genetic algorithm for the open capacitated arc routing problem. *Computers & Operations Research* 90 (2018), 221–231.

[3] José-Manuel Belenguer, Enrique Benavent, Philippe Lacomme, and Christian Prins. 2006. Lower and upper bounds for the mixed capacitated arc routing problem. *Computers & Operations Research* 33, 12 (2006), 3363–3383.

[4] Enrique Benavent, Vicente Campos, Angel Corberán, and Enrique Mota. 1992. The capacitated arc routing problem: lower bounds. *Networks* 22, 7 (1992), 669–690.

[5] Yuning Chen, Jin-Kao Hao, and Fred Glover. 2016. A hybrid metaheuristic approach for the capacitated arc routing problem. *European Journal of Operational Research* 253, 1 (2016), 25–39.

[6] Carlos A. Coello. 2006. Evolutionary multi-objective optimization: a historical view of the field. *IEEE computational intelligence magazine* 1, 1 (2006), 28–36.

[7] Carlos A. Coello, Silvia González Brambila, Josué Figueroa Gamboa, Ma Tapia, Castillo Guadalupe, and Raquel Hernández Gómez. 2020. Evolutionary multi-objective optimization: open research areas and some challenges lying ahead. *Complex & Intelligent Systems* 6, 2 (2020), 221–236.

[8] Miguel Constantino, Luís Gouveia, Maria Cândida Mourão, and Ana Catarina Nunes. 2015. The mixed capacitated arc routing problem with non-overlapping routes. *European Journal of Operational Research* 244, 2 (2015), 445–456.

[9] Ángel Corberán, Bruce L. Golden, Oliver Lum, Isaac Plana, and José M. Sanchis. 2017. Aesthetic considerations for the min-max k-windy rural postman problem. *Networks* 70, 3 (2017), 216–232.

[10] Ángel Corberán and Gilbert Laporte. 2015. *Arc routing: problems, methods, and applications*. SIAM.

[11] Angel Corberán and Christian Prins. 2010. Recent results on Arc Routing Problems: An annotated bibliography. *Networks* 56, 1 (2010), 50–69.

[12] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6, 2 (2002), 182–197.

[13] Moshe Dror. 2012. *Arc routing: theory, solutions and applications*. Springer Science & Business Media.

[14] Richard W. Eglese. 1994. Routeing winter gritting vehicles. *Discrete applied mathematics* 48, 3 (1994), 231–244.

[15] Matthias Ehrgott. 2005. *Multicriteria optimization* (second ed.). Springer Science & Business Media.

[16] Horst A. Eiselt, Michel Gendreau, and Gilbert Laporte. 1995. Arc routing problems, part I: The Chinese postman problem. *Operations Research* 43, 2 (1995), 231–242.

[17] Gérard Fleury, Philippe Lacomme, and Christian Prins. 2004. Evolutionary algorithms for stochastic arc routing problems. In *Workshop on Applications of Evolutionary Computation*. 501–512.

[18] Félix-Antoine Fortin, François-Michel De Rainville, Marc-André Gardner, Marc Parizeau, and Christian Gagné. 2012. DEAP: Evolutionary Algorithms Made Easy. *Journal of Machine Learning Research* 13 (July 2012), 2171–2175.

[19] Greg N. Frederickson. 1979. Approximation algorithms for some postman problems. *J. ACM* 26, 3 (1979), 538–554.

[20] Bruce L. Golden, James S. DeArmon, and Edward K. Baker. 1983. Computational experiments with algorithms for a class of routing problems. *Computers & Operations Research* 10, 1 (1983), 47–59.

[21] Bruce L. Golden and Richard T. Wong. 1981. Capacitated arc routing problems. *Networks* 11, 3 (1981), 305–315.

[22] Lucio Grandinetti, Francesca Guerriero, Demetrio Laganà, and Ornella Pisacane. 2012. An optimization-based heuristic for the multi-objective undirected capacitated arc routing problem. *Computers & Operations Research* 39, 10 (2012), 2300–2309.

[23] Hisashi Handa, Lee Chapman, and Xin Yao. 2005. Dynamic salting route optimisation using evolutionary computation. In *IEEE Congress on Evolutionary Computation*, Vol. 1. 158–165.

[24] Hisashi Handa, Lee Chapman, and Xin Yao. 2007. Robust salting route optimization using evolutionary algorithms. In *Evolutionary Computation in Dynamic and Uncertain Environments*. Springer, 497–517.

[25] Nicolas Jozefowiez, Frederic Semet, and El-Ghazali Talbi. 2008. *From single-objective to multi-objective vehicle routing problems: Motivations, case studies, and methods*. Springer, 445–471.

[26] Nicolas Jozefowiez, Frédéric Semet, and El-Ghazali Talbi. 2009. An evolutionary algorithm for the vehicle routing problem with route balancing. *European Journal of Operational Research* 195, 3 (2009), 761–769.

[27] Liu Juan, Cai Zixing, and Liu Jianqin. 2000. Premature convergence in genetic algorithm: Analysis and prevention based on chaos operator. In *IEEE World Congress on Intelligent Control and Automation*, Vol. 1. 495–499.

[28] Philippe Lacomme, Christian Prins, and Wahiba Ramdane-Cherif. 2004. Competitive Memetic Algorithms for Arc Routing Problems. *Annals of Operations Research* 131, 1 (2004), 159–185.

[29] Philippe Lacomme, Christian Prins, and Marc Sevaux. 2003. Multiobjective capacitated arc routing problem. In *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 550–564.

[30] Philippe Lacomme, Christian Prins, and Marc Sevaux. 2006. A genetic algorithm for a bi-objective capacitated arc routing problem. *Computers & Operations Research* 33, 12 (2006), 3473–3493.

[31] Min Liu, Hemant Kumar Singh, and Tapabrata Ray. 2014. A memetic algorithm with a new split scheme for solving dynamic capacitated arc routing problems. In *2014 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 595–602.

[32] Oliver Lum, Carmine Cerrone, Bruce L. Golden, and Edward Wasil. 2017. Partitioning a street network into compact, balanced, and visually appealing routes. *Networks* 69, 3 (2017), 290–303.

[33] Cristian Martinez, Irene Loiseau, Mauricio G.C. Resende, and S Rodriguez. 2011. BRKGA algorithm for the capacitated arc routing problem. *Electronic Notes in Theoretical Computer Science* 281 (2011), 69–83.

[34] Piotr Matl, Richard F. Hartl, and Thibaut Vidal. 2018. Workload equity in vehicle routing problems: A survey and analysis. *Transportation Science* 52, 2 (2018), 239–260.

[35] Piotr Matl, Richard F. Hartl, and Thibaut Vidal. 2019. Workload equity in vehicle routing: The impact of alternative workload resources. *Computers & Operations Research* 110 (2019), 116–129.

[36] Yi Mei, Xiaodong Li, and Xin Yao. 2013. Cooperative coevolution with route distance grouping for large-scale capacitated arc routing problems. *IEEE Transactions on Evolutionary Computation* 18, 3 (2013), 435–449.

[37] Yi Mei, Ke Tang, and Xin Yao. 2010. Capacitated arc routing problem in uncertain environments. In *IEEE Congress on Evolutionary Computation*. IEEE, 1–8.

[38] Yi Mei, Ke Tang, and Xin Yao. 2011. Decomposition-based memetic algorithm for multiobjective capacitated arc routing problem. *IEEE Transactions on Evolutionary Computation* 15, 2 (2011), 151–165.

[39] I. M. Oliver, D. J. Smith, and J. R. C. Holland. 1987. A Study of Permutation Crossover Operators on the Traveling Salesman Problem. In *International Conference on Genetic Algorithms on Genetic Algorithms and Their Application*. 224–230.

[40] ORTEC. 2020. http://www.ortec.com/

[41] Alexander Poot, Goossen Kant, and Albert Peter Marie Wagelmans. 2002. A savings based method for real-life vehicle routing problems. *Journal of the Operational Research Society* 53, 1 (2002), 57–68.

[42] Christian Prins. 2013. The Capacitated Arc Routing Problem: Heuristics. In *Arc routing: Problems, methods, and applications*. SIAM, Chapter 7, 131–157.

[43] Christian Prins, Nacima Labadi, and Mohamed Reghioui. 2009. Tour splitting algorithms for vehicle routing problems. *International Journal of Production Research* 47, 2 (2009), 507–535.

[44] Christian Prins, Philippe Lacomme, and Caroline Prodhon. 2014. Order-first split-second methods for vehicle routing problems: A review. *Transportation Research Part C: Emerging Technologies* 40 (2014), 179–200.

[45] Muhilan Ramamoorthy and Violet R. Syrotiuk. 2020. Online re-routing for vehicle breakdown in residential waste collection. In *IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)*. 1–5.

[46] Diego Gabriel Rossit, Daniele Vigo, Fernando Tohmé, and Mariano Frutos. 2019. Visual attractiveness in routing problems: A review. *Computers & Operations Research* 103 (March 2019), 13–34.

[47] Ronghua Shang, Jia Wang, Licheng Jiao, and Yuying Wang. 2014. An improved decomposition-based memetic algorithm for multi-objective capacitated arc routing problem. *Applied Soft Computing* 19 (2014), 343–361.

[48] Ronghua Shang, Yuying Wang, Jia Wang, Licheng Jiao, Shuo Wang, and Liping Qi. 2014. A multi-population cooperative coevolutionary algorithm for multi-objective capacitated arc routing problem. *Information Sciences* 277 (2014), 609–642.

[49] Supplementary Materials and Reproducibility Information. 2021. http://www.public.asu.edu/~syrotiuk/gecco21.html

[50] Ke Tang, Yi Mei, and Xin Yao. 2009. Memetic algorithm with extended neighborhood search for capacitated arc routing problems. *IEEE Transactions on Evolutionary Computation* 13, 5 (2009), 1151–1166.

[51] Gündüz Ulusoy. 1985. The fleet size and mix problem for capacitated arc routing. *European Journal of Operational Research* 22, 3 (1985), 329–337.

[52] Fábio Luiz Usberti, Paulo Morelato França, and André Luiz Morelato França. 2013. GRASP with evolutionary path-relinking for the capacitated arc routing problem. *Computers & Operations Research* 40, 12 (2013), 3206–3217.

[53] Juan Wang, Ke Tang, Jose A Lozano, and Xin Yao. 2015. Estimation of the distribution algorithm with a stochastic local search for uncertain capacitated arc routing problems. *IEEE Transactions on Evolutionary Computation* 20, 1 (2015), 96–109.

[54] Juan Wang, Ke Tang, and Xin Yao. 2013. A memetic algorithm for uncertain capacitated arc routing problems. In *IEEE Workshop on Memetic Computing*. 72–79.