Soft Comput
DOI 10.1007/s00500-016-2112-3

METHODOLOGIES AND APPLICATION

A bi-criteria evolutionary algorithm for a constrained multi-depot vehicle routing problem

Vikas Agrawal[1] · **Constance Lightner**[2] · **Carin Lightner-Laws**[3] · **Neal Wagner**[4]

Abstract Most research about the vehicle routing problem (VRP) does not collectively address many of the constraints that real-world transportation companies have regarding route assignments. Consequently, our primary objective is to explore solutions for real-world VRPs with a heterogeneous fleet of vehicles, multi-depot subcontractors (drivers), and pickup/delivery time window and location constraints. We use a nested bi-criteria genetic algorithm (GA) to minimize the total time to complete all jobs with the fewest number of route drivers. Our model will explore the issue of weighting the objectives (total time vs. number of drivers) and provide Pareto front solutions that can be used to make decisions on a case-by-case basis. Three different real-world data sets were used to compare the results of our GA vs. transportation field experts' job assignments. For the three data sets, all 21 Pareto efficient solutions yielded improved overall job completion times. In 57 % (12/21) of the cases, the Pareto efficient solutions also utilized fewer drivers than the field experts' job allocation strategies.

Keywords Vehicle routing problem · Bi-criteria genetic algorithm · Pareto front · Multi-depot transportation problem · Hard and soft time windows

1 Introduction and background of BSL

Amid significant shifts in the socioeconomic landscape, the growing complexity of global supply chains and innovative technological advances, it has become increasingly challenging for transportation companies to maintain their competitive advantage. Transportation-based companies are incessantly looking for ways to cut logistics costs, reduce waste, utilize information technologies, improve operational efficiency and increase overall productivity levels. Transportation companies are ultimately tasked with delivering packages efficiently and exceeding customer expectations.

After extensively researching logistics companies and existing methodologies for vehicle routing problems (VRP), it became apparent that the literature focused on more generalized problems that did not collectively address many of the variants that are common to transportation companies. Specifically, we perused the literature for research about solving VRPs that involved a heterogeneous fleet of vehicles, multi-depot subcontractors (drivers), and pickup/delivery time window and location constraints. Since we were unable to find existing literature that collectively addressed the aforementioned variants in a VRP, the initial phase of our research involved developing an evolutionary algorithm to minimize the total time to complete all jobs with the fewest number of route drivers given pickup/delivery times, locations, and vehicle capacity constraints (Lightner-Laws et al. 2016).

Communicated by V. Loia.

✉ Carin Lightner-Laws
CarinLightner-Laws@clayton.edu

Vikas Agrawal
avikas@gmail.com

Constance Lightner
clightner@uncfsu.edu

Neal Wagner
neal.f.wagner@gmail.com

[1] Jacksonville University, 2800 University Blvd N, Jacksonville, FL, USA

[2] Fayetteville State University, 1200 Murchison Rd., Fayetteville, NC, USA

[3] Clayton State University, 2000 Clayton State Boulevard, Morrow, GA, USA

[4] MIT Lincoln Laboratory, 244 Wood St., Lexington, MA, USA

Published online: 08 April 2016

Our results revealed an interesting dichotomy between the objectives of minimizing the overall completion time vs. the number of route drivers needed to complete all jobs. There was an apparent trade-off between improving either the total job completion time or number of drivers. Consequently, in this phase of our research, our primary objective is to minimize this bi-criteria problem by exploring varying weights that prioritize the time vs. the number of drivers needed.

We plan to use a nested bi-criteria genetic algorithm (GA) that computes a Pareto frontier to explore this VRP, with a heterogeneous fleet of 4 vehicle types, multi-depot subcontractors (drivers), and pickup/delivery time window and location constraints. Our model will explore the issues of weighting the objectives and provide Pareto front solutions that can be used to make decisions on a case-by-case basis. Three data sets from BSL, a mid-sized transportation company, will be used in this paper. BSL has customarily used in-house field specialists to assign jobs to each driver based on the pickup/delivery times, location, number of packages and weight of the load. Although this is a common practice, management wants to automate their process and find more efficient methods to allocate jobs for route drivers as a means of improving operations and competitiveness. Ultimately, our model will provide the Pareto front solutions (route assignments) that BSL can use on a case-by-case basis to assign jobs to route drivers.

This paper is organized as follows: Sect. 2 provides a review of the literature about VRPs. In Sect. 3, we describe our VRP and the associated constraints. Our GA framework and its specific components are presented in Sect. 4. The experimental results and discussion are given in Sect. 5. Finally, the conclusion and future research directions are presented in Sect. 6.

2 Literature review

The vehicle routing problem is a combinatorial problem that seeks to find the optimal route that minimizes the total travel distance required to deliver packages for a given set of customers. Typically customer demand is satisfied by a homogeneous fleet of vehicles from a central depot. Dantzig and Ramser (1959) capacitated vehicle routing problem (CVRP) is a VRP in which the homogeneous fleet of vehicles servicing the demand has a limited capacity which cannot be exceeded. The CVRP precipitated variant streams of research about multiple depots (Baldacci and Mingozzi 2009; Lau et al. 2010; Ombuki-Berman and Hanshar 2009; Vidal et al. 2011a), heterogeneous fleets (Baldacci and Mingozzi 2009; Brandao 2011; Choi and Tcha 2007), time windows for making deliveries (VRPTW), and designated pickup/delivery times (Dumas et al. 1991; Ropke and Cordeau 2009; Ropke et al. 2007; Baldacci et al. 2011).

A wide variety of closed form techniques have been used to solve VRPs. However, as more complex constraints are considered, finding an explicit optimal solution becomes computationally expensive and virtually impossible to ascertain; thus researchers have explored a variety of heuristics such as data mining (Chen et al. 2012), evolutionary algorithms (Vidal et al. 2011b; Lau et al. 2010), tabu searches (Cordeau and Maichberger 2011; Brandao 2011), graph theory (Likaj et al. 2013) and simulated annealing (1993) to solve these complex transportation problems. All of these approaches seek to find near optimal solutions to address routing problem variants.

There has been extensive research on VRPs; however, Pisinger and Ropke (2007) are among the few who have developed a single robust heuristic that can be used to solve five different variants of this transportation problem. Their approach will solve the vehicle routing problem with time windows (VRPTW), capacitated vehicle routing problem (CVRP), multi-depot vehicle routing problem (MDVRP), site-dependent vehicle routing problem (SDVRP) and the open vehicle routing problem (OVRP). While this heuristic has been applied to a variety of VRPs individually, it was not applied to multiple variants simultaneously in a single problem instance (i.e., a CVRP with multi-depots, time window constraints and a heterogeneous fleet) or multi-objective problems. Research has been conducted on general multi-objective transportation route problems using linear programming parametrics (Aneja and Nair 1979; Gal 1975; Zeleny 1974) and adjacent efficient methods (Evans and Steuer 1973; Yu and Zeleny 1974). Researchers have also used heuristic techniques and Pareto front solutions to explore bi-criteria transportation problems (Muller 2010; Prakash et al. 2014; Konak et al. 2006; Abounacer et al. 2012).

Recently, an increasing number of studies have emerged to solve a multi-criteria VRPTW. Tan et al. (2006) implemented a hybrid multi-objective evolutionary algorithm to minimize the total travel distance and the number of vehicles used to meet customer demand. They incorporated Pareto's optimality concepts for determining the best solutions to solve their multi-objective problem. Muller (2010) examined a VRPTW that utilized a homogeneous fleet of vehicles from a central depot. The objective of their research was to minimize overall costs and penalties for not adhering to soft time constraints. Their optimization problem had two competing objectives that made simultaneously optimizing both cost and penalties challenging. The dual objectives formed a Pareto front that the decision maker used to determine which solution best fits their needs. Their multi-objective optimization problem was solved using the ε-constraint method (Coello et al. 2002; Miettinen 1999), graph theory, Solomon's 11 insertion heuristic, ejection chain theory, or-opt and 2-opt procedures (Potvin and Rousseau 1995). Zou et al. (2013)

proposed a hybrid swarm optimization model for minimizing the number of vehicles utilized to serve customer demand, the total travel distance and the total waiting times. Their model assumed that there were an unlimited number of service vehicles. Banos et al. (2013) presented a Pareto-based simulated annealing model for solving a VRPTW that minimized the travel distance and the imbalance (travel distance and vehicle loads) of the individual routes.

Most single and multi-objective VRPTW research has been tested using Solomon's benchmark data (1987). This data set consists of 56 different problems with varying fleet sizes, vehicle capacities, and extensive information regarding the location of customer demand and travel time/distances between different location sites. This data set has been used extensively to compare different heuristic methods for solving a VRPTW. The research presented in this paper includes the additional variants of using a heterogeneous fleet and multiple depots, with a multi-objective VRPTW. Although these additional variants are common for a variety of real-world delivery/logistics scenarios, an exhaustive exploration of the literature did not reveal existing research about a VRPTW with all of the variants presented in this paper. As a result, no standardized benchmark datasets are available to test and compare with the results of this research.

Lightner-Laws et al. (2016) developed a heuristic solution for a VRP where multiple variants are presented in a single problem instance. Their approach utilized a GA to solve a VRP with a heterogeneous fleet of vehicles, multi-depot subcontractors, and constraints on pickup/delivery times and locations. The solution aimed to minimize the overall driving time to complete all job assignments while utilizing the fewest number of drivers. While their approach addressed multiple variants of the classic VRP, it only marginally explored the trade-off between overall driving time and the total number of drivers.

This paper further develops the solution proposed by Lightner-Laws et al. (2016) to optimize a multi-objective VRP where multiple variants of the problem exist in a single problem instance. We develop a nested GA, which explores the VRP for a heterogeneous fleet of vehicles with multi-depot subcontractors and constraints on pickup/delivery times and locations. We provide Pareto front solutions that allow BSL management to fully explore the trade-off between these two objectives.

3 Problem formulation

The multiple depot VRP presented in this research is concerned with the execution of a set of jobs where each job represents a package to be delivered. Each job is associated with a set of requirements that specify how the job is to be fulfilled. The following list gives the six requirements specified for each job:

- Pickup time (PT)
- Pickup location (PL)
- Delivery time (DT)
- Delivery location (DL)
- Vehicle type (VT)
- Job weight (JW)

PT is the earliest possible time the package can be picked up, while DT is the latest possible time that the package can be delivered. Specialists use their expertise to assign the appropriate vehicle types (based on customer estimates of the weight, size, and number of packages) for each job. Although cars (C), SUVs (S), box trucks (B), and tractor trailers (T) are the vehicle types used in this problem formulation, alternative vehicle types could also be easily incorporated as well. Finally, JW is the sum of the weight of all packages for a given job.

All jobs, J_i, (where $i = 1 \ldots N$ and N is the total number of jobs) to be fulfilled are known in advance. A potential driver list is created for each job, J_i, (based on customer input, estimated weight, size, and capacity constraints) from a set of available drivers, D_k, (where $k = 1 \ldots M$ and M is the total number of available drivers). A driver's job completion time is calculated from the time they leave their home and pickup their first job until the time their last job is delivered. The overall total time for all route deliveries is equivalent to the sum of the job completion times for each individual driver. Both mapping software and input from field experts are used to determine the travel time between any home, pickup or delivery location (Lightner-Laws et al. 2016).

BSL wishes to automate the process of allocating jobs to drivers and designating the order in which jobs should be picked up/delivered. The driver assignments should be made in a manner that minimizes total travel time and the total number of drivers required to complete all jobs.

4 Proposed framework

Holland (1975) developed a stochastic search technique, a genetic algorithm (GA), which incorporates features of natural selection to solve optimization problems. The main components in the design of a GA include:

- Candidate solution representation or chromosomes,
- Selection strategy,
- Genetic operators,
- Fitness evaluation,
- Termination requirement.

A GA starts with a population of initial solutions. Genetic operations, (mutations and/or crossover events) alter initial parent solutions to create new children solutions. The next generation of solutions is selected from the set of parent and children solutions, in such a way that the best solutions have a higher probability of continuing to the next generation. The goal is to produce improved solutions at each generation by ultimately allowing natural selection to filter out the weaker candidates.

Our optimization model features a nested GA. Our main GA seeks to minimize the total distance traveled and the number of drivers required to complete all jobs. This genetic algorithm employs a secondary GA to help compute the fitness of a candidate solution. The secondary GA minimizes the total travel time for an individual driver by finding the optimal ordering of assigned pickups and deliveries. Our primary GA will be referred to as the Outer GA, and the secondary GA will be referred to as the Inner GA.

4.1 Evolutionary algorithm for outer GA

4.1.1 Candidate solution representation/initial population generation

Candidate solution representation refers to the encoding scheme that is used to represent a candidate solution. Suppose we were given the five jobs and potential driver list for completing each job shown in Table 1.

As mentioned in a previous section, available driver lists for each job are determined by BSL experts based upon the availability of BSL drivers with vehicle types that meet the established requirements. For this example, job J_1 can be serviced by driver D_1 or D_2; however, driver D_4 is the only driver available to service job J_5.

Figure 1 depicts our encoding scheme for three possible candidate solutions to this problem. Our encoding scheme includes the pickup and drop off locations for each job since optimal driver routes may include consecutive (nested) pickups and/or drop offs when multiple jobs are assigned to a single driver, in addition to the maximum amount of weight that the driver is carrying at any given time. In the first solution, driver D_2 was assigned job J_1; thus the pickup location J_1PL and drop off location J_1DL are assigned to driver D_2. Since driver D_2 only carried job J_1, the maximum weight that the driver carried at any given time is 300 lbs (the total package weight for job J_1). Driver D_3 was assigned jobs J_2, J_3, and J_4, where they picked up and delivered each job consecutively, and the maximum weight that the driver carried at any given time was 850 lbs. Additionally, driver D_4 was assigned job J_5(600 lbs). In the second solution, driver D_2 was assigned jobs $J_1$and J_3; driver D_3 was assigned jobs J_2 and J_4, and driver D_4 was assigned job J_5. The last candidate solution assigned jobs J_1, J_2, J_3, and J_4 to driver D_2, and job J_5 to driver D_4. The maximum weight carried by a driver is recorded at all times.

To create our initial population, we use the aforementioned encoding scheme to randomly assign job pickup and delivery locations to drivers, based upon the potential drivers list for each job. We impose large time penalties in the fitness function for each candidate solution if time conflicts make it impossible for the assigned driver to complete all jobs within the given time constraints, or if the known vehicle capacity was exceeded. These penalties effectively prevent infeasible candidates from being considered when other feasible candidates have been found.

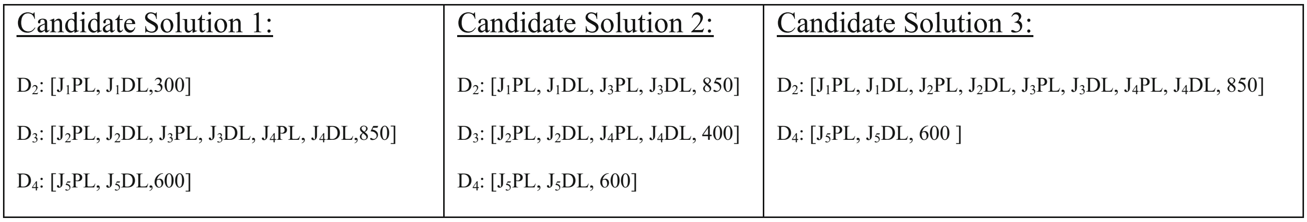
4.1.2 Selection strategy

We adopt the tournament selection strategy to select candidate solutions for breeding. Our "tournament" is conducted by randomly selecting two candidate solutions from the current population, comparing their fitness values, and then declaring the strongest and weakest candidate of the pair (i.e. stronger candidates have high fitness values). Since weaker candidates can sometimes produce strong progenies, we set a selection probability of 0.9 that the strongest candidate solution from each pair will be selected for genetic operation.

Additionally, we employ the elitism strategy, where the best candidate solution from each generation is directly copied into the next population, unaltered. For our model, we select the top three candidate solutions to directly move on to the next generation. We then conduct tournaments, as described above, until our next generation is fully populated.

4.1.3 Genetic operator-mutation

Once a candidate solution is identified via the tournament selection process, our model utilizes mutation as the genetic operator. A candidate solution is mutated as follows:



Candidate Solution 1:	Candidate Solution 2:	Candidate Solution 3:
D_2: [J_1PL, J_1DL,300]	D_2: [J_1PL, J_1DL, J_3PL, J_3DL, 850]	D_2: [J_1PL, J_1DL, J_2PL, J_2DL, J_3PL, J_3DL, J_4PL, J_4DL, 850]
D_3: [J_2PL, J_2DL, J_3PL, J_3DL, J_4PL, J_4DL,850]	D_3: [J_2PL, J_2DL, J_4PL, J_4DL, 400]	D_4: [J_5PL, J_5DL, 600]
D_4: [J_5PL, J_5DL,600]	D_4: [J_5PL, J_5DL, 600]	

Fig. 1 Sample candidate solutions

Table 1 Sample potential drivers list

Job	Drivers available to service job	Job weight (lbs)
J_1	D_1, D_2	300
J_2	D_1, D_2, D_3	400
J_3	D_2, D_3	850
J_4	D_1, D_2, D_3	400
J_5	D_4	600

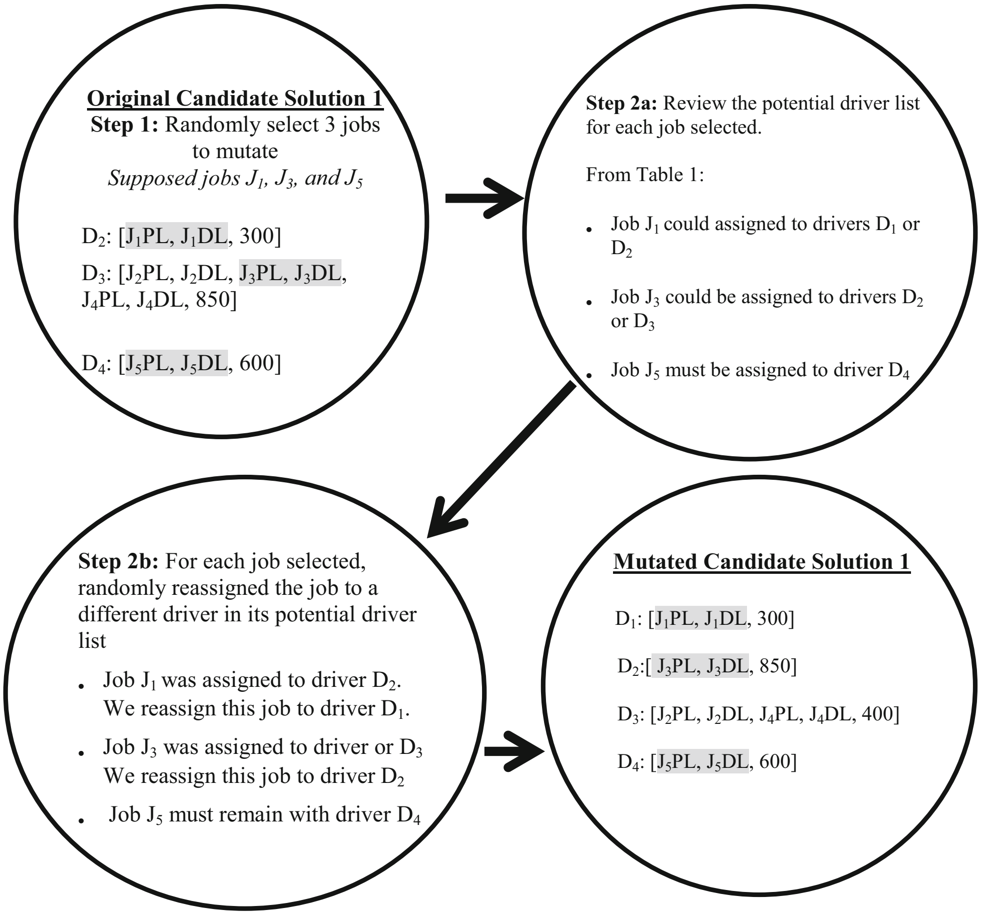
1. We randomly select three current job assignments.
2. For each job selected, we randomly reassign the job to a different driver in the potential driver list for that job. If a selected job only has one potential driver in its list, no change is made.

For example, suppose we are given the potential drivers list provided in Table 1, and our tournament selection process produced Candidate Solution 1 (from Fig. 1). Our process for mutating Candidate Solution 1 is illustrated in Fig. 2. In step 1 of this figure, we see that jobs J_1, J_3, and J_5 are randomly selected to mutate. In step 2, we review the potential driver lists for the three jobs selected, and reassign each job to a different driver from their respective list. Since driver D_4 is the only available driver that can complete job J_5, this job assignment remained unchanged.

4.1.4 Fitness evaluation

The fitness evaluation function mathematically expresses the value of a candidate solution. Genetic algorithms are primarily concerned with finding the best or "most fit" solutions for the final population. Accordingly, we start our fitness evaluation, by checking the vehicle weight capacities for all proposed driver assignment solutions. We immediately impose a large fitness penalty value to all solutions that violate capacity limits; thus reducing the likelihood of invalid solutions continuing to the next generation (Michalewicz 1995).

For all valid solutions we focus on the multi-objective goal of minimizing the travel time and the number of drivers to complete all jobs. Like many multi-criteria problems, it is challenging to optimize both objectives simultaneously. In



Original Candidate Solution 1
Step 1: Randomly select 3 jobs to mutate
Supposed jobs J_1, J_3, and J_5

D_2: [J_1PL, J_1DL, 300]
D_3: [J_2PL, J_2DL, J_3PL, J_3DL, J_4PL, J_4DL, 850]
D_4: [J_5PL, J_5DL, 600]

Step 2a: Review the potential driver list for each job selected.

From Table 1:

- Job J_1 could assigned to drivers D_1 or D_2
- Job J_3 could be assigned to drivers D_2 or D_3
- Job J_5 must be assigned to driver D_4

Step 2b: For each job selected, randomly reassigned the job to a different driver in its potential driver list

- Job J_1 was assigned to driver D_2. We reassign this job to driver D_1.
- Job J_3 was assigned to driver or D_3 We reassign this job to driver D_2
- Job J_5 must remain with driver D_4

Mutated Candidate Solution 1

D_1: [J_1PL, J_1DL, 300]
D_2:[J_3PL, J_3DL, 850]
D_3: [J_2PL, J_2DL, J_4PL, J_4DL, 400]
D_4: [J_5PL, J_5DL, 600]

Fig. 2 Sample mutation process

an effort to address potential conflicts, we aim to generate a Pareto optimal set consisting of solutions that are not dominated by either objective. For our problem, we define our first objective as minimizing the travel time and our second objective as minimizing the number of drivers.

We use the weighted approach to formulate our multi-objective problem into the following single scalar objective function:

$$\text{Min } z = w^* z_1(x) + (1-w)^* z_2(x), \quad \text{for } 0 \leq w \leq 1 \qquad (4.1.1)$$

where x represents a candidate solution and $z_i(x)$ represents the ith normalized objective function.

Our normalized first objective, $z_1(x)$ is

$$z_1(x) = f_1(x)/f_1{}^* \qquad (4.1.2)$$

In this equation $f_1(x)$ is the travel time required for all jobs to be completed using the driver allocation specified by candidate solution x. A separate GA model, which we refer to as our Inner GA (discussed below), is used to compute the minimum time for a driver to complete the assigned jobs. The Inner GA returns the time and optimal order that jobs should be picked up/delivered. Thus, for a single candidate solution, each driver assignment must be processed through our Inner GA, and the sum of the minimum times returned for all drivers will serve as $f_1(x)$ for the candidate solution. The function f_1^* is the minimal travel time to complete all jobs if we consider time as our only objective. We discuss f_1^* further below (Initial GA Model Parameters).

Conversely, our normalized second objective

$$z_2(x) = f_2(x)/f_2{}^* \qquad (4.1.3)$$

is computed as the total number of drivers that the candidate solution x allocated to complete all jobs, $f_2(x)$, divided by the minimal number of drivers, f_2^*. The denominator in this equation represents the minimal number of drivers to complete all jobs if we consider the total number of drivers as our only objective. We discuss f_2^* further below (Initial GA Model Parameters).

One of the challenges in using the weighted sum approach for addressing multi-objective problems is determining the appropriate weights to accurately reflect the priorities of the objectives. Since BSL was unable to definitively prioritize their goals, this issue of how to weight objectives surfaced in our research. Management asserted that determining the best trade-off between minimizing time and the total number of drivers is a judgment call that is made on a case-by-case basis. Accordingly, we choose to vary the weight in our multi-objective fitness function (Eq. 4.1.1) from 0 to 1, using increments of 0.1; the rationale is that varying the weight captures a spectrum of driver allocation solutions by gradually shifting the priority for each objective. We use this approach to determine the Pareto front solution set for our problem.

4.1.5 Initial GA model parameters

Before running our GA using our multi-objective fitness function (described in Eq. 4.1.1) we must determine f_1^* and f_2^*. f_1^* is determined by running our GA using total travel time as our fitness function and completely ignoring the number of drivers required to complete all jobs. The best time produced by our single objective GA, determines our f_1^* value for Eq. 4.1.1. We determine the value of f_2^* by running our GA using the single objective of minimizing the number of drivers needed to complete all jobs, without regard to the requisite travel distance. These parameters are then used in Eq. 4.1.1 to determine driver allocations for our multi-objective fitness function.

4.1.6 Termination condition

The Outer GA continues for a fixed number of generations or until no improvement in solution quality is seen. The final population is then analyzed to determine the solution with the minimum time and then the fewest number of total drivers.

4.2 Evolutionary algorithm for inner GA

The job assignments for a single driver are input into the Inner GA. The Inner GA is used to process all driver assignments and return the corresponding minimum time and ordering that jobs should be picked up/delivered. For example, let us consider the following assignment for Driver 2:

D_2 : [J_1PL, J_1DL, J_2PL, J_2DL]

Originally the driver is scheduled to pickup job J_1, deliver job J_1, pickup job J_2, then deliver job J_2. However, the minimum time solution may be to pickup job J_1, pickup job J_2, deliver job J_2, then deliver job J_1. The Inner GA would accept the original driver assignment above, and return the candidate solution below with its respective travel time:

D_2 : [J_1PL, J_2PL, J_2DL, J_1DL]

In computing the total travel time, we assume that drivers always begin at a known home location, before making their first pickup. The travel time ends after the last job is delivered. For a candidate solution to be considered valid, the pickup for each job must precede its delivery in the assignment ordering. Additionally, a substantial time penalty is added to any potential assignment that cannot (due to travel time conflicts) meet the required pickup and delivery times of all assigned jobs. This penalty significantly reduces the likelihood that these solutions are selected to continue on to later generations.

4.2.1 Candidate solution representation initial population generation

We retain the encoding scheme shown above for encoding candidate solutions for our Inner GA. The initial pool of candidate solutions for the first generation is created by taking the list of pickup and drop off locations for a driver and randomly shuffling their respective positions to form a new initial candidate solution. The generated solution is then checked to ensure that it is valid. Figure 3 shows a sample initial generation that could be generated from an original driver assignment. In this figure the notation D_kH denotes the home location of driver D_k.

4.2.2 Selection strategy

We employ the same selection strategy used for the Outer GA described above.

4.2.3 Genetic operator-mutation

Once a candidate solution is identified through our tournament selection process, our model utilizes mutation as the genetic operator. To be consistent with common GA notation, we refer to a candidate solution as a chromosome and we refer to a single item within a solution as a gene. A candidate chromosome is mutated as follows:

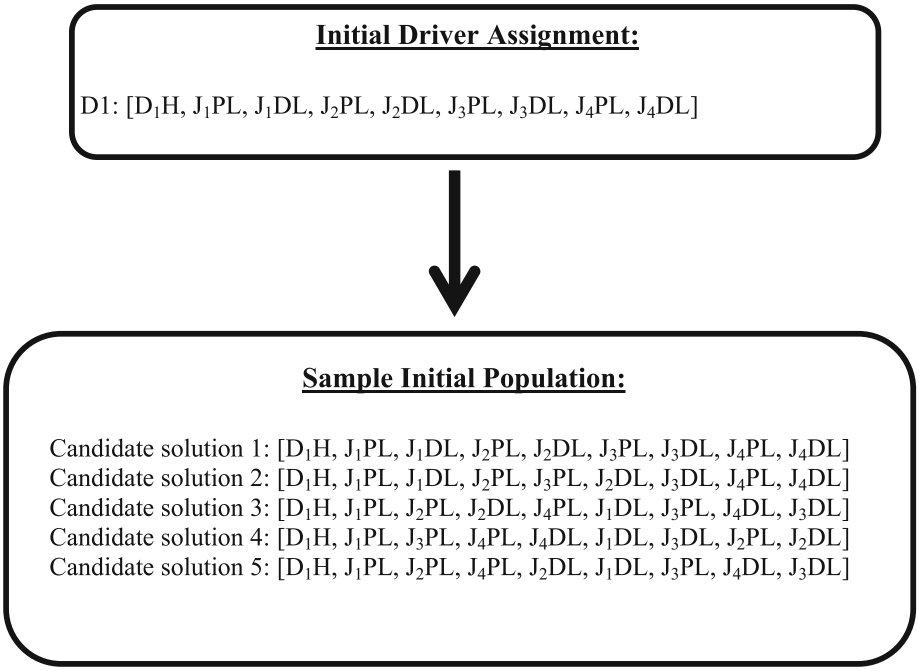
- Step1: Randomly select one drop off or pickup location within the chromosome (i.e., randomly select one gene).
- Step2: With a 0.5 probability, exchange the selected gene with its neighbor to the left or right.
- Step3: Check the validity of new solution by making sure that following conditions are met: (a) the first position must remain the driver's home location, and (b) drop off location must come after the respective pickup location. If the solution is not valid, repeat steps 1 and 2 above up to three times, in attempt to find a valid mutated solution. After 3 attempts, if a valid solution has not been found, the original candidate solution will be copied unaltered into the next generation.

Figure 4 shows a mutation process that yields an invalid candidate solution. For this instance, we must abandon the mutated solution and make up to two additional attempts to mutate the original candidate solution. If the future attempts yield a valid solution, the mutated candidate solution will continue to the next generation. If the future attempts fail to produce a valid mutated solution, the original candidate solution will move on to the next generation, unaltered. Figure 5 illustrates a mutation process that yields a valid candidate solution.

4.2.4 Fitness evaluation

The fitness value for a candidate solution is calculated by adding the travel times to reach each successive location in the solution. For example, consider the following candidate solution:

Fig. 3 Sample initial population for five candidate solutions



Initial Driver Assignment:

D1: [D_1H, J_1PL, J_1DL, J_2PL, J_2DL, J_3PL, J_3DL, J_4PL, J_4DL]

Sample Initial Population:

Candidate solution 1: [D_1H, J_1PL, J_1DL, J_2PL, J_2DL, J_3PL, J_3DL, J_4PL, J_4DL]
Candidate solution 2: [D_1H, J_1PL, J_1DL, J_2PL, J_3PL, J_2DL, J_3DL, J_4PL, J_4DL]
Candidate solution 3: [D_1H, J_1PL, J_2PL, J_2DL, J_4PL, J_1DL, J_3PL, J_4DL, J_3DL]
Candidate solution 4: [D_1H, J_1PL, J_3PL, J_4PL, J_4DL, J_1DL, J_3DL, J_2PL, J_2DL]
Candidate solution 5: [D_1H, J_1PL, J_2PL, J_4PL, J_2DL, J_1DL, J_3PL, J_4DL, J_3DL]

Selected Candidate Solution for Mutation:

[D_1H, J_1PL, J_1DL, J_2PL, J_3PL, J_2DL, J_3DL, J_4PL, J_4DL]

Step 1: Select a gene to mutate.
Suppose the gene J_1DL was selected.

[D_1H, J_1PL, J_1DL, J_2PL, J_3PL, J_2DL, J_3DL, J_4PL, J_4DL]

Step 2: With a .5 probability, exchange the selected gene with its neighbor to the right or left.
Suppose this probability process selected the left neighbor, J_1PL. These two genes are exchanged.

Mutated Candidate Solution:
[D_1H, J_1DL, J_1PL , J_2PL, J_3PL, J_2DL, J_3DL, J_4PL, J_4DL]

Step 3: Check the validity of the mutated solution.
This solution is not valid since the ordering has job J1 being dropped off before it was picked up. Therefore we must abandon this solution.

Fig. 4 Sample mutation process yielding an invalid solution

Selected Candidate Solution for Mutation:

[D_1H, J_1PL, J_1DL, J_2PL, J_3PL, J_2DL, J_3DL, J_4PL, J_4DL]

Step 1: Select a gene to mutate.
Suppose the gene J_3PL was selected.

[D_1H, J_1PL, J_1DL, J_2PL, J_3PL, J_2DL, J_3DL, J_4PL, J_4DL]

Step 2: With a .5 probability, exchange the selected gene with its neighbor to the right or left.
Suppose this probability process selected the right neighbor, J_2DL. These two genes are exchanged.

Mutated Candidate Solution:
[D_1H, J_1PL, J_1DL , J_2PL, J_2DL, J_3PL, J_3DL, J_4PL, J_4DL]

Step 3: Check the validity of the mutated solution.
This solution is valid since the ordering has jobs being picked up before delivery. Thus this mutated candidate solution will move on to the next generation.

Fig. 5 Sample mutation process yielding a valid solution

[D_1H, J_1PL, J_1DL, J_2PL, J_3PL, J_2DL, J_3DL, J_4PL, J_4DL]

Its fitness value would be the travel time from driver D_1's home to job J_1's pickup location, plus the travel time from job J_1's pickup location to job J_1's drop off location, plus the travel time from job J_1's drop off location to job J_2's pickup location, and so on.

4.2.5 Termination condition

The Inner GA continues for a fixed number of generations or until no improvement in solution quality is seen. Once the termination condition is met, the Inner GA returns the minimum time solution (and ordering) for the driver's job assignments.

5 Experimental results and discussion

5.1 BSL parameters

5.1.1 Sample problem using BSL parameters

Initially, we used a small sample problem to test our GA and ensure that all six variants were properly incorporated into the model. We used BSLs Midwest metropolitan area service area for the pickup/delivery locations in the sample problem. To determine the time and distance required to complete jobs within the service area, 141 zones were established based on geographical locations. GPS software was then used to ascertain the distance and time between any two zones within the service area; finally, this information was used to calculate the overall time and distance required to complete all jobs.

There were fifty drivers, with unique driver identification numbers (ID#), available to service customers in this sample problem. Specifically, there were 18 car drivers (ID# D1–D18), 16 SUV drivers (ID# D19–D34), 13 Box Truck drivers (ID# D35–D47) and 3 tractor trailers drivers (ID# D48–D50). It is worth noting that SUVs have a unique degree of flexibility because they could be used for jobs, which require cars or SUVs.

The sample input data are displayed in Table 2. Specifically, the table gives the earliest pickup times, the latest delivery times, the pickup/delivery location zones, the vehicle type required, maximum vehicle type weight capacity, and the job weights for 11 different jobs. A perusal of the vehicle types for jobs 2, 4 and 6 raises two questions: (1) why is an SUV required for Job 2 and the job weight is only 180 lbs? and (2) why is a box truck required for jobs 4 and 6, when the weights are only 1000 and 468 lbs, respectively? These jobs highlight the need for transportation field experts to be involved in the process of assigning vehicle types. Although these three jobs weigh considerably less than the maximum vehicle type weight capacity, other factors such as the length, width, surface area, fragility, density, materials handling requirements, etc. must also be considered when choosing the appropriate vehicle type; thus, factors other than

Table 2 Sample input data

Job#	Pickup time	Delivery time	Pickup location	Delivery location	(Max vehicle type capacity) Vehicle type	Job weight
J1	6:30	16:00	134	76	C (1500)	280
J2	9:30	11:30	51	68	S (2000)	180
J3	8:00	11:00	130	88	T (43,500)	42,000
J4	8:30	11:30	127	98	B (10,500)	1000
J5	9:00	11:30	139	124	C (1500)	300
J6	8:30	11:30	136	65	B (10,500)	468
J7	7:00	9:00	43	42	C (1500)	500
J8	8:00	10:00	43	121	C (1500)	25
J9	9:00	11:00	103	55	C (1500)	200
J10	8:00	9:00	43	141	C (1500)	1300
J11	8:00	10:30	100	128	B (10,500)	10,000

simply the weight dictated that a larger vehicle was required to complete jobs 2, 4 and 6.

Table 3 displays the four Pareto front solutions and specific coordinates (total travel time, number of drivers) for the sample problem. In this table, the MTPW is the maximum total package weight at any time and the VC is the maximum amount of cargo that the vehicle can carry (or the maximum vehicle type capacity). Let us examine the $W = 0.3$, Run 2, driver 19 data. For this solution, driver 19 uses an SUV to complete jobs 1, 7 and 8. Specifically, driver 19 would pickup job 8, pickup job 7, drop off job 7, drop off job 8, pickup job 1 and drop off job 1. Jobs 7 and 8 would be in the vehicle at the same time (MTPW = 525 lbs) and job 1 (MTPW = 280 lbs) would be in driver 19's SUV alone. The maximum total package weight at any time would be less than the maximum amount of cargo that an SUV can carry (VC = 2000).

Figure 6 shows a graph of the 110 solutions found by our GA, under varying values of w. For each solution, the graph shows the travel time vs. the number of drivers required to complete all jobs. Our Pareto front solutions are depicted as square-shaped data points in the graph. The solutions in Table 3 and Fig. 6 encompass all six variants—pickup/delivery time windows, pickup/delivery locations and vehicle capacity constraints.

5.1.2 Results from GA vs. BSL job allocations

We conducted our experiments using actual BSL data from three different days. In Table 4, there is a complete list of all jobs that needed to be filled on three different days. This table also provides the earliest pickup times, the latest delivery times, the pickup/delivery locations, the type of vehicle needed for the job and the drivers that could potentially do the jobs. There were sixty-eight drivers, with unique driver identification numbers (ID#), available to service customers. Specifically, there were 18 car drivers (ID# D1–D18), 16 SUV drivers (ID# D19–D34), 13 Box Truck drivers (ID# D35–D47) and 21 tractor trailers drivers (ID# D48–D50, D56–D64, D66, D68, D70, D71, D73–D77).

BSL does not currently preserve information about the estimated weight or package dimensions of fulfilled jobs in their records. Thus, for these experiments we assume that BSL vehicle type assignments are made correctly and hence have the available capacity to fulfill all assigned jobs over the course of a day. However, it is worth noting that our model is designed to ensure that vehicle capacity constraints are not violated when package weights are provided (refer to the sample problem in the previous section).

Preliminary experiments for both the inner and outer GAs, which systematically investigate alternative population sizes and maximum number of generations, were executed to determine efficacious GA parameter settings. Upon consideration of total computation time, overall job completion time, and marginal improvement of generated results, the population size and maximum number of generations were both set to 25. Prior to implementing our bi-criteria objective, the GA was run twice—once with the objective of minimizing the total travel time and once to minimize the number of drivers. The results from these two runs were used to determine f_1^* and f_2^* as described in Sect. 4 above. The minimum travel time, $f_1^{*,}$ was determined to be 1260 for our Day 1 data set, 1673 for Day 2, and 2039 for Day 3. The minimum number of drivers, f_2^*, was determined to be 8 for our Day 1 data set, 6 for Day 2, and 15 for Day 3. Our multi-objective fitness function was investigated for $w = [0, 0.1, 0.2, \ldots, 1]$. For each of the 3 data sets, the GA was run 10 times for each value of w (110 total runs per data set). All experimental results were performed using a Windows 7 Professional 64-bit Operating system with 6 GB of RAM and Inter(R) Xeon(R) CPU W3530 @ 2.8GHz.

Table 3 Pareto front Solutions for our Sample Data Problem

W=.9, Run 6	W=.3, Run 2	W=.3, Run 6	W=0, Run 4
D$_{27}$: [J$_5$PL, J$_5$DL] MTPW*: 300 lbs VC**: 2000 lbs	D$_{42}$: [J$_6$PL, J$_4$PL, J$_4$DL, J$_6$DL] MTPW*: 1468 lbs VC**: 10500 lbs	D$_{27}$: [J$_5$PL, J$_8$PL, J$_8$DL, J$_5$DL] MTPW*: 325 lbs VC**: 2000 lbs	D$_6$: [J$_5$PL, J$_5$DL] MTPW*: 300 lbs VC**: 1500 lbs
D$_{25}$: [J$_2$PL, J$_9$PL, J$_9$DL, J$_2$DL] MTPW*: 380 lbs VC**: 2000 lbs	D$_{33}$: [J$_2$PL, J$_9$PL, J$_9$DL, J$_2$DL] MTPW*: 380 lbs VC**: 2000 lbs	D$_{45}$: [J$_{11}$PL, J$_{11}$DL] MTPW*: 10000 lbs VC**: 10500 lbs	D$_{25}$: [J$_8$PL, J$_7$PL, J$_7$DL, J$_8$DL, J$_1$PL, J$_1$DL] MTPW*:525 lbs VC**: 2000 lbs
D$_{26}$: [J$_1$PL, J$_1$DL] MTPW*: 280 VC**: 2000 lbs	D$_{19}$: [J$_8$PL, J$_7$PL, J$_7$DL, J$_8$DL, J$_1$PL, J$_1$DL] MTPW*: 525 lbs VC**: 2000 lbs	D$_{25}$: [J$_{10}$PL, J$_7$PL, J$_7$DL, J$_{10}$DL, J$_1$PL, J$_1$DL] MTPW*: 1300 lbs VC**: 2000 lbs	D$_{46}$: [J$_6$PL, J$_4$PL, J$_4$DL, J$_6$DL, J$_{11}$PL, J$_{11}$DL] MTPW*:10000 lbs VC**: 10500 lbs
D$_{46}$: [J$_{11}$PL, J$_{11}$DL] MTPW*: 10000 lbs VC**: 10500 lbs	D$_{50}$: [J$_3$PL, J$_3$DL] MTPW*: 42000 lbs VC**: 43500 lbs	D$_{31}$: [J$_2$PL, J$_9$PL, J$_9$DL, J$_2$DL] MTPW*: 380 lbs VC**: 2000 lbs	D$_{50}$: [J$_3$PL, J$_3$DL] MTPW*: 42000 lbs VC**: 43500 lbs
D$_{31}$: [J$_8$PL, J$_7$PL, J$_7$DL, J$_8$DL] MTPW*: 525 lbs VC**: 2000 lbs	D$_{36}$: [J$_{11}$PL, J$_{11}$DL] MTPW*: 10000 lbs VC**: 10500 lbs	D$_{49}$: [J$_3$PL, J$_3$DL] MTPW*: 42000 lbs VC**: 43500 lbs	D$_{22}$: [J$_{10}$PL, J$_{10}$DL , J$_9$PL, J$_2$PL, J$_9$DL, J$_2$DL] MTPW*: 1300 lbs VC**: 2000 lbs
D$_{48}$: [J$_3$PL, J$_3$DL] MTPW*:42000 lbs VC**: 43500 lbs	D$_{16}$: [J$_{10}$PL, J$_{10}$DL] MTPW*: 1300 lbs VC**: 1500 lbs	D$_{39}$: [J$_6$PL, J$_4$PL, J$_4$DL, J$_6$DL] MTPW*: 1468 lbs VC**: 10500 lbs	
D$_{38}$: [J$_6$PL, J$_4$PL, J$_4$DL, J$_6$DL] MTPW*: 1468 VC**: 10500 lbs	D$_9$: [J$_5$PL, J$_5$DL] MTPW*: 300 lbs VC**: 1500 lbs		
D$_{18}$: [J$_{10}$PL, J$_{10}$DL] MTPW*: 1300 lbs VC**: 1500 lbs			
Min Time: 831 **# of Drivers: 8**	**Min Time: 859** **# of Drivers: 7**	**Min Time: 884** **# of Drivers: 6**	**Min Time: 1251** **# of Drivers: 5**

* MTPW is the maximum total package weight at any time

** VC is the maximum amount of cargo that vehicle can carry

Fig. 6 Graph of GA solutions and Pareto front for our sample data problem

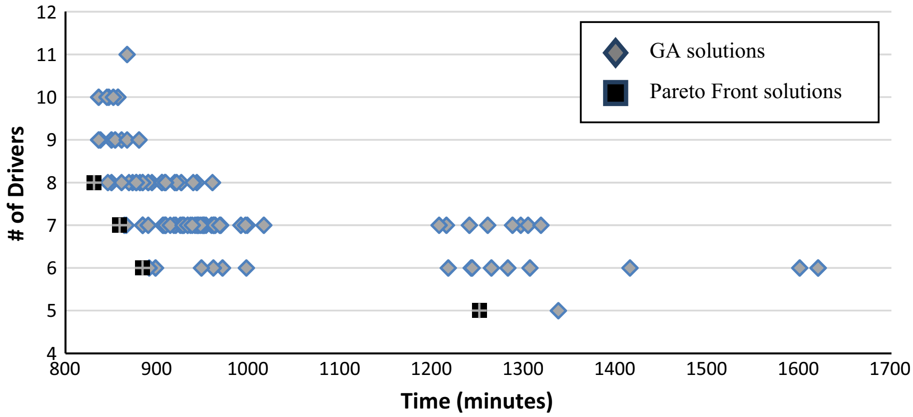


Table 4 Details for each Job on Days 1, 2 and 3

Job#	Pickup time	Delivery time	Pickup location	Delivery location	Vehicle type
Day 1					
Jl	6:30	16:00	134	76	C
J2	9:30	17:00	137	111	S
J3	9:30	11:30	119	78	C
J4	11:45	13:45	64	60	C
J5	9:30	11:30	51	68	S
J6	8:00	11:00	130	88	T
J7	6:30	7:30	82	43	C
J8	8:00	10:00	40	104	C
J9	8:30	11:30	127	98	B
J10	9:00	11:30	139	124	C
Jll	8:30	11:30	136	65	B
J12	7:00	9:00	43	42	C
J13	8:00	10:00	43	121	C
J14	9:00	11:00	103	55	C
J15	8:00	9:00	43	141	S
J16	8:00	10:30	100	128	B
Day 2					
Jl	11:45	13:45	94	60	C
J2	8:30	10:30	85	135	S
J3	9:00	12:00	46	99	B
J4	9:00	12:00	41	99	B
J5	10:00	13:00	127	98	B
J6	10:00	13:00	96	48	B
J7	9:30	11:30	75	132	C
J8	8:00	10:00	141	95	C
J9	7:00	16:00	69	62	C
J10	7:00	9:00	121	43	C
Jll	7:00	10:00	43	95	C
J12	5:00	7:00	43	121	C
J13	7:00	9:00	43	109	C
J14	9:00	8:30	100	73	B
J15	9:00	11:00	100	45	B
Day 3					
Jl	14:00	16:00	92	81	C
J2	8:30	10:30	122	90	C
J3	10:30	13:30	56	115	B
J4	9:00	11:00	71	72	S
J5	15:30	17:30	116	53	C
J6	9:00	11:00	110	118	C
J7	13:30	16:30	67	82	B
J8	9:00	12:00	107	130	B
J9	9:00	11:00	84	93	S
J10	11:00	13:30	61	114	S
Jll	7:00	10:00	89	63	B
J12	13:00	15:00	91	66	C
J13	10:30	12:30	97	50	S

Table 4 continued

Job#	Pickup time	Delivery time	Pickup location	Delivery location	Vehicle type
Day 3 continued					
J14	13:30	14:30	40	86	C
J15	14:30	15:30	86	40	C
J16	7:30	9:30	141	43	C
J17	9:00	11:00	74	58	C
J18	9:00	12:00	74	105	C
J19	9:00	11:30	74	140	C
J20	9:00	11:30	74	133	C
J21	9:00	11:00	74	87	C
J22	9:00	11:00	121	95	C
J23	9:00	12:00	74	83	C
J24	9:00	11:30	74	101	C
J25	9:00	11:00	74	59	C
J26	9:00	11:00	74	112	C
J27	9:00	11:00	74	77	C
J28	9:00	11:00	74	120	C
J29	9:00	11:00	74	70	C
J30	9:00	11:00	74	125	C
J31	9:00	11:00	74	131	C
J32	9:00	11:00	74	117	C
J33	9:00	11:00	74	54	C
J34	9:00	11:00	74	126	C
J35	9:00	11:00	74	57	C
J36	9:00	11:30	74	108	C
J37	9:00	12:00	74	47	C
J38	7:00	8:00	43	121	C
J39	10:00	11:00	138	123	C

Vehicle type	Available drivers that could do job
Cars (C)	D1–D34
SUVs (S)	D19–D34
Tractor trailer (T)	D48–D50, D56–D64, D66, D68, D70, D71, D73–D77
Box Truck (B)	D35–D47

The pickup, delivery, vehicle type needed and the available drivers for each job on Days 1, 2 and 3

For the Day 1 jobs, Fig. 7 shows a graph of the 110 solutions found by our GA, under varying values of w. For each solution, the graph shows the travel time vs. the number of drivers required to complete all jobs. Similarly, Figs. 8 and 9 show the solutions yielded from our GA for Days 2 and 3, respectively. Table 5 displays the specific coordinates (total travel time, number of drivers) for our 7 Pareto front solutions for Day 1, 6 Pareto front solutions for Day 2, and 8 Pareto front solutions for Day 3.

Tables 6, 7 and 8 show the specific job assignments for each Pareto solution, overall completion times and number of drivers for our GA for Days 1, 2 and 3. The final column of these tables displays the job assignments that BSL experts gave to each driver; while the set of jobs are listed for each driver, the order that the jobs were picked up and delivered was not provided. Thus, we ran the job assignments that BSL provided through our program that determines the optimal route (i.e. the order for pickups/deliveries) for a set of jobs. We used these optimal routes to determine the minimum time that it would take the drivers to complete all orders. It is important to note that these times are based on mapping software's estimates of driving times between two locations and may not account for delays due to traffic or weather or alternative route selections made by individual drivers.

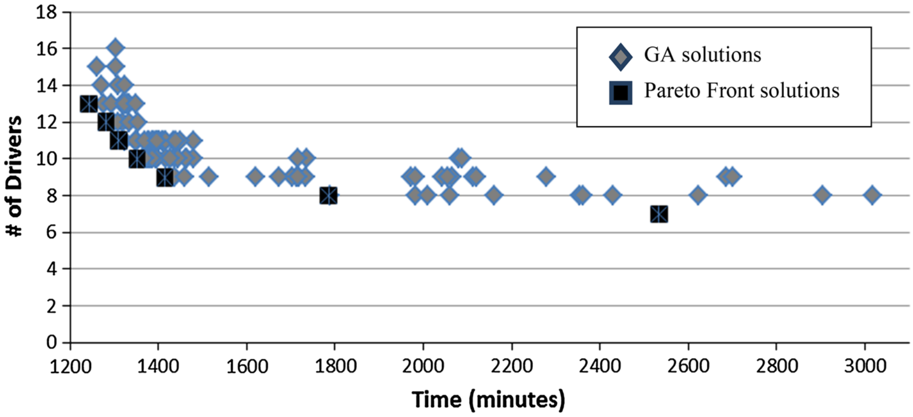


Fig. 7 Graph of GA solutions and Pareto front for day 1 jobs

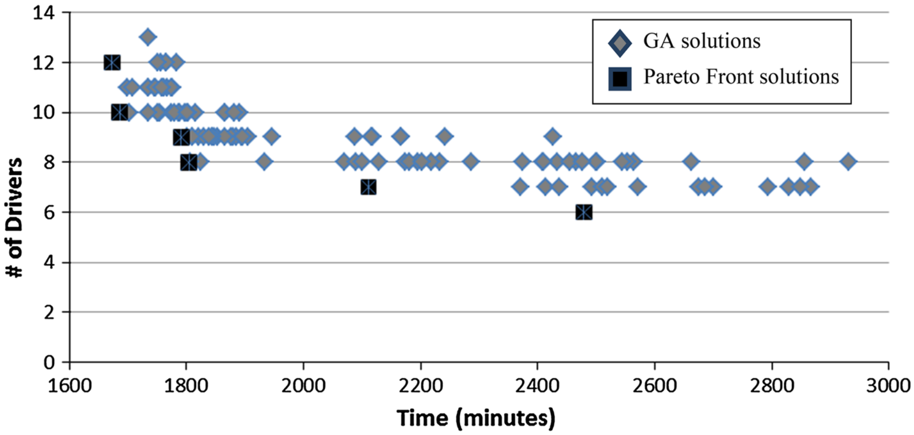


Fig. 8 Graph of GA solutions and Pareto front for day 2 jobs

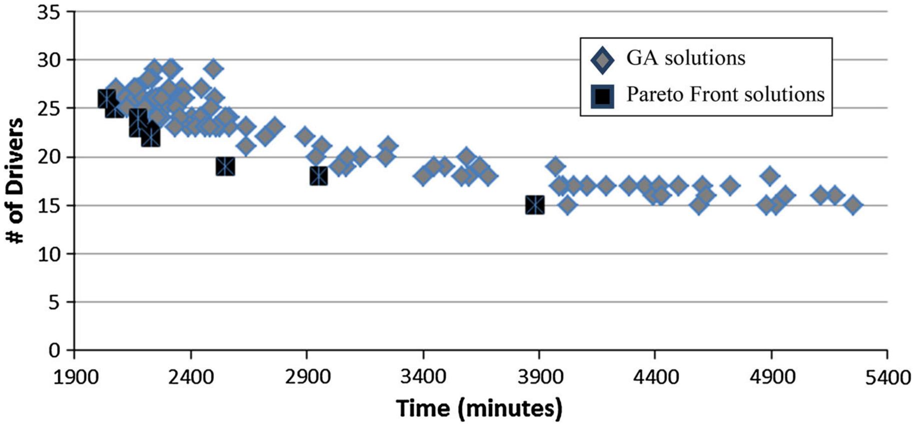


Fig. 9 Graph of GA solutions and Pareto front for day 3 jobs

If we examine our Day 1 results, there are 7 Pareto optimal solutions with total job completion times ranging from 1243 to 2533 min as compared with 1571 min from the BSL job assignments. This represents an improvement in completion time of up to 21 %. Additionally, while 14 drivers were required for the BSL solution, only 8–13 drivers were needed for the Pareto solutions from the GA. Thus, the GA was able to provide automated solutions that would reduce both travel time and the number of drivers. Specifically, one of our Pareto solutions yielded 1414 min and 9 drivers. This solution reduced the travel time by 10 % and showed a 36 % decrease in the number of drivers when compared to BSL's manual driver allocations.

For Day 2, the Pareto front consists of 6 distinct points. The total job completion times for the Pareto solutions ranged from 1673 to 2479 min for the GA while the BSL solution was 1858 min. This represents time improvements of up to 10 %. Here, 12 drivers were required for the BSL solution in

Table 5 Summary of Pareto front results for our three problems

W	Day 1: 16 Jobs ($f_1^* = 1260$, $f_2^* = 8$)	Day 2: 15 Jobs ($f_1^* = 1673$, $f_2^* = 6$)	Day 3: 39 Jobs ($f_1^* = 2039$, $f_2^* = 15$)
1	–	1673 min, 12 drivers	2073 min, 25 drivers
	–	–	2039 min, 26 drivers
0.9	1310 min, 11 drivers	–	–
	1243 min, 13 drivers	–	–
0.8	1281 min, 12 drivers	1686 min, 10 drivers	2177 min, 23 drivers
0.7	–	1804 min, 8 drivers	–
	–	1791 min, 9 drivers	–
0.6	–	–	2228 min, 22 drivers
	–	–	2174 min, 24 drivers
0.5	1414 min, 9 drivers	–	–
0.4	–	–	–
0.3	1352 min, 10 drivers	2111 min, 7 drivers	2952 min, 18 drivers
0.2	2533 min, 7 drivers	–	2548 min, 19 drivers
0.1	1785 min, 8 drivers	–	3883 min, 15 drivers
0	–	2479 min, 6 drivers	–

contrast to 6–12 drivers needed for the Pareto solutions from the GA. One of our Pareto solutions yielded 1686 min and 10 drivers; again, the GA was able to improve the travel time and number of drivers by 9 and 17 %, respectively, compared to BSL's solution.

For Day 3, the 8 solutions in the Pareto front had job completion times spanning 2039–3883 min while utilizing 15–26 drivers. BSL's route allocation used 15 drivers who completed the jobs in 3698 min. The GA was able to improve the total completion time by up to 45 %. However, in providing this dramatic improvement in time, it utilized 11 more drivers than the BSL solution. One of our Pareto solutions yielded 2948 min with 18 drivers. With this solution, the travel time was reduced by 20 % with a 20 % increase in the number of drivers.

The results from the GA on Days 1 and 2 revealed that the overall job completion times were markedly better than the BSL solutions while the total number of drivers was equal or better. For the Day 3 results, the GA provides dramatically better completion times but utilizes more drivers in doing so. Overall, our model was able to successfully automate their driver assignment problem while providing options for marked improvement of their combined objectives.

6 Conclusion and future directions

In our collaborative efforts with BSL, we address their need to automate the route assignment process by developing a nested GA for a heterogeneous fleet of vehicles with multi-depot subcontractors, pickup/delivery time windows, varying pickup/delivery locations, job weights and vehicle capacity constraints. BSL can input job requests for a given day and the model will yield the Pareto optimal solutions for their problem. Our model specifies the driver assignments, the order that jobs should be picked up/delivered, and the total time and number of drivers needed to complete all jobs. This allows BSL the ability to fully explore the trade-offs between driving time and number of drivers. Once a particular solution representing the best trade-off has been selected, the model will specify the driver assignments that should be designated.

Three different data sets from BSL were used to compare the results of the GA and BSL field experts. Our GA showed that the total job completion times were improved up to 21, 10 and 45 % on Days 1, 2, and 3 respectively. In addition, 12 of 13 Pareto front solutions for Days 1 and 2 reveal a decrease in the total number of drivers needed to complete all jobs. On Day 3, while results from the GA solutions yielded remarkably lower total job completion times, the BSL assignment yielded the fewest number of drivers. The Day 3 results indicate the need for us to continue to work with BSL to further understand how they prioritize both objectives (minimizing time vs. number of drivers required) on a daily basis. Overall, our GA results were promising and provided a substantive alternative to manually allocating job assignments.

In the future, we plan to modify our algorithm so that we can explore how unforeseen changes in the environment or disruptions in transportation can impact route assignments. For instance, if a driver has delivered some jobs and has a breakdown, we will explore which drivers can take over the remaining jobs most efficiently. We also intend to solve this vehicle routing problem using other meta-

Table 6 Pareto front solutions from our GA and the BSL job Allocations for the Day 1 jobs

W=0.9, Run 1	**W=0.9, Run 9**	**W=0.8, Run 5**	**W=0.5, Run 2**
D_{30}: [J_3PL, J_3DL, J_5PL, J_5DL] D_{32}: [J_1PL, J_1DL] D_{35}: [J_{16}PL, J_{16}DL] D_{62}: [J_6PL, J_6DL] D_{22}: [J_{10}PL, J_{13}PL, J_{13}DL, J_{10}DL] D_3: [J_{14}PL, J_8PL, J_8DL, J_{14}DL] D_{25}: [J_{15}PL, J_{12}PL, J_{12}DL, J_{15}DL] D_{17}: [J_4PL, J_4DL] D_{14}: [J_7PL, J_7DL] D_{28}: [J_2PL, J_2DL] D_{47}: [J_{11}PL, J_9PL, J_9DL, J_{11}DL]	D_{21}: [J_4PL, J_4DL] D_{27}: [J_{10}PL, J_{10}DL] D_{31}: [J_{12}PL, J_7PL, J_7DL, J_{12}DL] D_{32}: [J_2PL, J_2DL] D_{35}: [J_{11}PL, J_9PL, J_9DL, J_{11}DL] D_{33}: [J_5PL, J_5DL] D_{62}: [J_6PL, J_6DL] D_4: [J_{15}PL, J_{15}DL] D_3: [J_3PL, J_3DL] D_{42}: [J_{16}PL, J_{16}DL] D_9: [J_{13}PL, J_{13}DL, J_1PL, J_1DL] D_{16}: [J_{14}PL, J_{14}DL] D_{13}: [J_8PL, J_8DL]	D_{30}: [J_{15}PL, J_{12}PL, J_{12}DL, J_{15}DL] D_{61}: [J_6PL, J_6DL] D_{44}: [J_{11}PL, J_9PL, J_9DL, J_{11}DL] D_{35}: [J_{16}PL, J_{16}DL] D_{32}: [J_1PL, J_1DL] D_{33}: [J_7PL, J_7DL] D_1: [J_3PL, J_3DL] D_{22}: [J_8PL, J_8DL, J_2PL, J_2DL] D_{23}: [J_5PL, J_{14}PL, J_{14}DL, J_5DL] D_{18}: [J_4PL, J_4DL] D_7: [J_{13}PL, J_{13}DL] D_{27}: [J_{10}PL, J_{10}DL]	D_{46}: [J_{11}PL, J_9PL, J_9DL, J_{11}DL] D_{50}: [J_6PL, J_6DL] D_{19}: [J_{15}PL, J_{12}PL, J_{12}DL, J_{15}DL, J_2PL, J_2DL] D_{34}: [J_3PL, J_{14}PL, J_5PL, J_3DL, J_{14}DL, J_5DL] D_{25}: [J_8PL, J_8DL, J_1PL, J_1DL] D_{43}: [J_{16}PL, J_{16}DL] D_{15}: [J_{10}PL, J_{13}PL, J_{13}DL, J_{10}DL] D_{14}: [J_4PL, J_4DL] D_{13}: [J_7PL, J_7DL]
Min Time: 1310	**Min Time: 1243**	**Min Time: 1281**	**Min Time: 1414**
# of Drivers: 11	**# of Drivers:13**	**# of Drivers: 12**	**# of Drivers: 9**
W=0.3, Run 8	**W=0.2, Run 9**	**W=0.1, Run 9**	**BSL Job Assignments**
D_{31}: [J_5PL, J_3PL, J_3DL, J_5DL] D_{20}: [J_{14}PL, J_{14}DL, J_{13}PL, J_{13}DL, J_{10}PL, J_{10}DL] D_{37}: [J_{11}PL, J_9PL, J_9DL, J_{11}DL] D_4: [J_4PL, J_4DL] D_{34}: [J_8PL, J_8DL, J_2PL, J_2DL] D_6: [J_{12}PL, J_{12}DL, J_1PL, J_1DL] D_{77}: [J_6PL, J_6DL] D_{25}: [J_7PL, J_7DL] D_{28}: [J_{15}PL, J_{15}DL] D_{47}: [J_{16}PL, J_{16}DL]	D_{30}: [J_5PL, J_{15}PL, J_{13}PL, J_4PL, J_{15}DL, J_5DL, J_4DL, J_{13}DL, J_2PL, J_2DL] D_{45}: [J_{11}PL, J_9PL, J_9DL, J_{11}DL] D_{62}: [J_6PL, J_6DL] D_{19}: [J_{10}PL, J_{10}DL] D_{34}: [J_8PL, J_8DL, J_{12}PL, J_{12}DL, J_{14}PL, J_{14}DL] D_{21}: [J_3PL, J_7PL, J_7DL, J_3DL, J_1PL, J_1DL] D_{38}: [J_{16}PL, J_{16}DL]	D_{30}: [J_7PL, J_7DL] D_{35}: [J_{11}PL, J_9PL, J_9DL, J_{11}DL] D_1: [J_3PL, J_4PL, J_3DL, J_4DL] D_4: [J_{13}PL, J_{12}PL, J_{12}DL, J_{13}DL, J_1PL, J_{14}PL, J_{14}DL, J_1DL] D_{57}: [J_6PL, J_6DL] D_3: [J_{15}PL, J_{15}DL, J_8PL, J_8DL] D_{41}: [J_{16}PL, J_{16}DL] D_{29}: [J_{10}PL, J_{10}DL, J_5PL, J_5DL, J_2PL, J_2DL]	D_2:[J_{13}] D_3 :[J_{10}] D_7: [J_1] D_{19}: [J_8] D_{20}: [J_7, J_{12}, J_{15}] D_{22}:[J_3] D_{31}: [J_5] D_{41}: [J_2] D_{44}: [J_9] D_{45}: [J_{11}] D_{57}: [J_4] D_{63}: [J_{16}] D_{70}: [J_6] D_{77}: [J_{14}]
Min Time: 1352	**Min Time: 2533**	**Min Time: 1785**	**Min Time: 1571**
# of Drivers: 10	**# of Drivers: 7**	**# of Drivers: 8**	**# of Drivers: 14**

Table 7 Pareto Front Solutions from our GA and the BSL Job Allocations for the Day 2 Jobs

W=1, Run 8	**W=0.8, Run 2**	**W=0.7, Run 5**	**W=0.7, Run 8**
D_{45}: [J_3PL, J_4PL, J_3DL, J_4DL] D_{34}: [J_2PL, J_2DL] D_{39}: [J_5PL, J_5DL] D_{23}: [J_1PL, J_1DL] D_{38}: [J_{15}PL, J_{14}PL, J_{14}DL, J_{15}DL] D_5: [J_{13}PL, J_{13}DL] D_{25}: [J_9PL, J_{11}PL, J_9DL, J_{11}DL] D_7: [J_{12}PL, J_{12}DL] D_{17}: [J_8PL, J_8DL] D_{42}: [J_6PL, J_6DL] D_{29}: [J_7PL, J_7DL] D_{12}: [J_{10}PL, J_{10}DL]	D_{27}: [J_2PL, J_2DL] D_{45}: [J_6PL, J_6DL] D_{25}: [J_{12}PL, J_{12}DL] D_{42}: [J_{15}PL, J_{14}PL, J_{14}DL, J_{15}DL] D_{46}: [J_5PL, J_5DL] D_{26}: [J_7PL, J_8PL, J_8DL, J_7DL] D_{12}: [J_{10}PL, J_{13}PL, J_{10}DL, J_{13}DL] D_{15}: [J_1PL, J_1DL] D_{18}: [J_9PL, J_{11}PL, J_9DL, J_{11}DL] D_{47}: [J_3PL, J_4PL, J_3DL, J_4DL]	D_4: [J_8PL, J_8DL] D_{43}: [J_6PL, J_6DL, J_3PL, J_5PL, J_3DL, J_5DL] D_{33}: [J_{12}PL, J_{12}DL] D_{30}: [J_2PL, J_2DL] D_{31}: [J_7PL, J_9PL, J_7DL, J_9DL] D_{12}: [J_{10}PL, J_{10}DL, J_{11}PL, J_{13}PL, J_{13}DL, J_{11}DL] D_{37}: [J_4PL, J_4DL, J_{14}PL, J_{15}PL, J_{14}DL, J_{15}DL] D_{22}: [J_1PL, J_1DL]	D_{27}: [J_2PL, J_2DL] D_{40}: [J_6PL, J_6DL] D_3: [J_1PL, J_1DL] D_{28}: [J_{12}PL, J_{12}DL] D_{15}: [J_{10}PL, J_9PL, J_{10}DL, J_{11}PL, J_9DL, J_{11}DL] D_{36}: [J_3PL, J_4PL, J_3DL, J_4DL] D_{17}: [J_7PL, J_7DL] D_{38}: [J_5PL, J_5DL, J_{15}PL, J_{14}PL, J_{14}DL, J_{15}DL] D_7: [J_{13}PL, J_{13}DL, J_8PL, J_8DL]
Min Time: 1673	**Min Time: 1686**	**Min Time: 1804**	**Min Time: 1791**
# of Drivers: 12	**# of Drivers: 10**	**# of Drivers: 8**	**# of Drivers: 9**

W=0.3, Run 7	**W=0, Run 6**	**BSL Job Assignments**
D_{30}: [J_7PL, J_7DL] D_2: [J_1PL, J_1DL] D_{34}: [J_{11}PL, J_{12}PL, J_{12}DL, J_{10}PL, J_{10}DL, J_8PL, J_8DL, J_{11}DL, J_2PL, J_2DL] D_{37}: [J_4PL, J_4DL, J_5PL, J_5DL] D_{36}: [J_3PL, J_3DL] D_{38}: [J_6PL, J_6DL, J_{15}PL, J_{14}PL, J_{14}DL, J_{15}DL] D_7: [J_{13}PL, J_9PL, J_{13}DL, J_9DL]	D_{45}: [J_5PL, J_5DL, J_{14}PL, J_{15}PL, J_{14}DL, J_{15}DL] D_{44}: [J_6PL, J_6DL, J_3PL, J_3DL] D_{39}: [J_4PL, J_4DL] D_{26}: [J_2PL, J_2DL, J_8PL, J_8DL, J_9PL, J_9DL, J_1PL, J_1DL] D_{18}: [J_{11}PL, J_{11}DL, J_7PL, J_7DL] D_{27}: [J_{10}PL, J_{12}PL, J_{13}PL, J_{10}DL, J_{12}DL, J_{13}DL]	D_2:[J_{13}] D_4:[J_1] D_8: [J_7] D_9: [J_{10}] D_{19}: [J_8, J_{11}] D_{20}:[J_{12}] D_{22}: [J_2] D_{23}: [J_9] D_{35}: [J_5, J_{14}, J_{15}] D_{36}: [J_6] D_{40}: [J_3] D_{41}: [J_4]
Min Time: 2111	**Min Time: 2479**	**Min Time: 1858**
# of Drivers: 7	**# of Drivers: 6**	**# of Drivers: 12**

Table 8 Pareto front solutions from our GA and the BSL job allocations for the day 3 jobs

W=1, Run 1	**W=1, Run 6**	**W=0.8, Run 5**	**W=0.6, Run 5**
D_{44}: [J_{11}PL, J_{11}DL]	D_{46}: [J_{11}PL, J_{11}DL]	D_{40}: [J_{8}PL, J_{8}DL]	D_{20}: [J_{6}PL, J_{32}PL, J_{2}PL, J_{2}DL, J_{32}DL, J_{6}DL, J_{10}PL, J_{10}DL]
D_{27}: [J_{13}PL, J_{13}DL]	D_{10}: [J_{1}PL, J_{1}DL]	D_{41}: [J_{3}PL, J_{3}DL]	D_{19}: [J_{28}PL, J_{28}DL]
D_{41}: [J_{7}PL, J_{7}DL]	D_{27}: [J_{13}PL, J_{13}DL]	D_{25}: [J_{5}PL, J_{5}DL]	D_{24}: [J_{9}PL, J_{22}PL, J_{25}PL, J_{25}DL, J_{9}DL, J_{22}DL]
D_{25}: [J_{23}PL, J_{37}PL, J_{23}DL, J_{37}DL]	D_{25}: [J_{10}PL, J_{10}DL]	D_{26}: [J_{14}PL, J_{14}DL]	D_{23}: [J_{15}PL, J_{15}DL]
D_{26}: [J_{38}PL, J_{38}DL]	D_{43}: [J_{7}PL, J_{7}DL]	D_{19}: [J_{21}PL, J_{9}PL, J_{21}DL, J_{9}DL]	D_{27}: [J_{13}PL, J_{13}DL]
D_{19}: [J_{34}PL, J_{22}PL, J_{34}DL, J_{9}PL, J_{9}DL, J_{22}DL]	D_{19}: [J_{2}PL, J_{2}DL]	D_{20}: [J_{25}PL, J_{28}PL, J_{25}DL, J_{28}DL, J_{10}PL, J_{10}DL]	D_{30}: [J_{1}PL, J_{14}PL, J_{14}DL, J_{1}DL]
D_{20}: [J_{10}PL, J_{10}DL]	D_{20}: [J_{19}PL, J_{23}PL, J_{20}PL, J_{20}DL, J_{19}DL, J_{23}DL]	D_{35}: [J_{7}PL, J_{7}DL]	D_{31}: [J_{4}PL, J_{4}DL, J_{23}PL, J_{23}DL]
D_{35}: [J_{8}PL, J_{8}DL]	D_{36}: [J_{3}PL, J_{3}DL]	D_{23}: [J_{12}PL, J_{12}DL]	D_{1}: [J_{37}PL, J_{36}PL, J_{29}PL, J_{34}PL, J_{34}DL, J_{29}DL, J_{36}DL, J_{37}DL]
D_{21}: [J_{16}PL, J_{16}DL]	D_{22}: [J_{5}PL, J_{5}DL]	D_{37}: [J_{11}PL, J_{11}DL]	D_{33}: [J_{17}PL, J_{17}DL]
D_{39}: [J_{3}PL, J_{3}DL]	D_{39}: [J_{8}PL, J_{8}DL]	D_{22}: [J_{37}PL, J_{32}PL, J_{33}PL, J_{32}DL, J_{33}DL, J_{37}DL]	D_{37}: [J_{8}PL, J_{8}DL]
D_{28}: [J_{4}PL, J_{4}DL, J_{32}PL, J_{32}DL]	D_{29}: [J_{24}PL, J_{24}DL]	D_{29}: [J_{27}PL, J_{26}PL, J_{27}DL, J_{26}DL]	D_{3}: [J_{12}PL, J_{12}DL]
D_{5}: [J_{30}PL, J_{36}PL, J_{30}DL, J_{36}DL]	D_{28}: [J_{9}PL, J_{25}PL, J_{4}PL, J_{4}DL, J_{39}PL, J_{39}DL, J_{25}DL, J_{9}DL]	D_{28}: [J_{4}PL, J_{4}DL, J_{30}PL, J_{30}DL]	D_{36}: [J_{3}PL, J_{3}DL]
D_{3}: [J_{18}PL, J_{18}DL]	D_{6}: [J_{34}PL, J_{34}DL]	D_{6}: [J_{29}PL, J_{29}DL]	D_{6}: [J_{16}PL, J_{5}PL, J_{5}DL, J_{16}DL]
D_{32}: [J_{14}PL, J_{14}DL]	D_{5}: [J_{12}PL, J_{12}DL]	D_{5}: [J_{15}PL, J_{15}DL]	D_{5}: [J_{33}PL, J_{35}PL, J_{21}PL, J_{30}PL, J_{30}DL, J_{33}DL, J_{35}DL, J_{21}DL]
D_{33}: [J_{17}PL, J_{20}PL, J_{20}DL, J_{17}DL]	D_{34}: [J_{15}PL, J_{15}DL]	D_{34}: [J_{39}PL, J_{39}DL, J_{36}PL, J_{36}DL]	D_{8}: [J_{39}PL, J_{39}DL]
D_{30}: [J_{27}PL, J_{31}PL, J_{29}PL, J_{2}PL, J_{27}DL, J_{31}DL, J_{29}DL, J_{2}DL]	D_{3}: [J_{22}PL, J_{35}PL, J_{33}PL, J_{33}DL, J_{35}DL, J_{22}DL]	D_{3}: [J_{24}PL, J_{17}PL, J_{23}PL, J_{18}PL, J_{24}DL, J_{23}DL, J_{18}DL, J_{17}DL]	D_{40}: [J_{11}PL, J_{11}DL]
D_{31}: [J_{25}PL, J_{35}PL, J_{25}DL, J_{35}DL]	D_{32}: [J_{36}PL, J_{29}PL, J_{21}PL, J_{29}DL, J_{21}DL, J_{36}DL]	D_{32}: [J_{22}PL, J_{35}PL, J_{35}DL, J_{22}DL]	D_{17}: [J_{38}PL, J_{38}DL]
D_{12}: [J_{6}PL, J_{6}DL]	D_{1}: [J_{38}PL, J_{38}DL]	D_{30}: [J_{13}PL, J_{13}DL]	D_{43}: [J_{7}PL, J_{7}DL]
D_{13}: [J_{24}PL, J_{24}DL]	D_{31}: [J_{37}PL, J_{37}DL]	D_{13}: [J_{20}PL, J_{20}DL]	D_{9}: [J_{20}PL, J_{20}DL]
D_{14}: [J_{33}PL, J_{21}PL, J_{21}DL, J_{33}DL]	D_{12}: [J_{14}PL, J_{14}DL]	D_{16}: [J_{19}PL, J_{19}DL]	D_{14}: [J_{27}PL, J_{18}PL, J_{27}DL, J_{18}DL]
D_{15}: [J_{28}PL, J_{19}PL, J_{28}DL, J_{19}DL]	D_{13}: [J_{18}PL, J_{18}DL]	D_{9}: [J_{38}PL, J_{38}DL, J_{16}PL, J_{16}DL, J_{2}PL, J_{2}DL]	D_{13}: [J_{31}PL, J_{26}PL, J_{26}DL, J_{31}DL]
D_{16}: [J_{5}PL, J_{5}DL]	D_{14}: [J_{6}PL, J_{6}DL]	D_{8}: [J_{31}PL, J_{34}PL, J_{6}PL, J_{6}DL, J_{34}DL, J_{31}DL]	D_{12}: [J_{24}PL, J_{19}PL, J_{24}DL, J_{19}DL]
D_{17}: [J_{12}PL, J_{12}DL]	D_{15}: [J_{30}PL, J_{26}PL, J_{26}DL, J_{30}DL]	D_{18}: [J_{1}PL, J_{1}DL]	
D_{18}: [J_{15}PL, J_{1}PL, J_{15}DL, J_{1}DL]	D_{16}: [J_{17}PL, J_{31}PL, J_{32}PL, J_{27}PL, J_{27}DL, J_{31}DL, J_{32}DL, J_{17}DL]		
D_{7}: [J_{39}PL, J_{39}DL, J_{26}PL, J_{26}DL]	D_{9}: [J_{16}PL, J_{16}DL]		
	D_{7}: [J_{28}PL, J_{28}DL]		
Min Time: 2073	**Min Time: 2039**	**Min Time: 2177**	**Min Time: 2228**
# of Drivers: 25	**# of Drivers: 26**	**# of Drivers: 23**	**# of Drivers: 22**

Table 8 continued

W=0.6, Run 10	**W=0.3, Run 4**	**W=0.2, Run 6**	**W=0, Run 6**
D_{20}: [J_{10}PL, J_{10}DL]	D_{11}: [J_{18}PL, J_{26}PL, J_{17}PL, J_{26}DL, J_{17}DL, J_{18}DL]	D_{22}: [J_{25}PL, J_{37}PL, J_{25}DL, J_{31}PL, J_{39}PL, J_{39}DL, J_{31}DL, J_{37}DL, J_{5}PL, J_{5}DL]	D_{11}: [J_{26}PL, J_{34}PL, J_{1}PL, J_{34}DL, J_{26}DL, J_{1}DL]
D_{19}: [J_{20}PL, J_{20}DL]	D_{10}: [J_{38}PL, J_{38}DL]	D_{24}: [J_{20}PL, J_{20}DL]	D_{21}: [J_{9}PL, J_{9}DL, J_{20}PL, J_{20}DL]
D_{21}: [J_{37}PL, J_{21}PL, J_{26}PL, J_{26}DL, J_{21}DL, J_{37}DL]	D_{19}: [J_{16}PL, J_{16}DL]	D_{23}: [J_{35}PL, J_{9}PL, J_{35}DL, J_{9}DL]	D_{24}: [J_{12}PL, J_{12}DL]
D_{23}: [J_{13}PL, J_{13}DL]	D_{24}: [J_{20}PL, J_{19}PL, J_{28}PL, J_{28}DL, J_{20}DL, J_{19}DL]	D_{26}: [J_{29}PL, J_{18}PL, J_{29}DL, J_{19}PL, J_{19}DL, J_{18}DL, J_{10}PL, J_{10}DL, J_{14}PL, J_{14}DL]	D_{23}: [J_{33}PL, J_{4}PL, J_{16}PL, J_{33}DL, J_{16}DL, J_{4}DL]
D_{26}: [J_{30}PL, J_{33}PL, J_{30}DL, J_{33}DL]	D_{26}: [J_{9}PL, J_{24}PL, J_{25}PL, J_{23}PL, J_{25}DL, J_{9}DL, J_{24}DL, J_{23}DL]	D_{25}: [J_{1}PL, J_{1}DL]	D_{27}: [J_{31}PL, J_{31}DL]
D_{25}: [J_{15}PL, J_{1}PL, J_{15}DL, J_{1}DL]	D_{25}: [J_{15}PL, J_{1}PL, J_{15}DL, J_{1}DL]	D_{29}: [J_{4}PL, J_{4}DL]	D_{30}: [J_{27}PL, J_{17}PL, J_{23}PL, J_{27}DL, J_{17}DL, J_{23}DL, J_{10}PL, J_{10}DL]
D_{27}: [J_{22}PL, J_{4}PL, J_{4}DL, J_{22}DL]	D_{27}: [J_{22}PL, J_{13}PL, J_{13}DL, J_{22}DL, J_{14}PL, J_{14}DL]	D_{30}: [J_{12}PL, J_{12}DL]	D_{31}: [J_{6}PL, J_{18}PL, J_{6}DL, J_{18}DL, J_{13}PL, J_{14}PL, J_{14}DL, J_{13}DL]
D_{30}: [J_{12}PL, J_{12}DL]	D_{30}: [J_{4}PL, J_{4}DL, J_{21}PL, J_{33}PL, J_{5}PL, J_{33}DL, J_{21}DL, J_{5}DL]	D_{31}: [J_{13}PL, J_{13}DL]	D_{2}: [J_{36}PL, J_{37}PL, J_{22}PL, J_{22}DL, J_{37}DL, J_{36}DL]
D_{32}: [J_{9}PL, J_{29}PL, J_{29}DL, J_{9}DL]	D_{31}: [J_{30}PL, J_{30}DL, J_{36}PL, J_{36}DL, J_{10}PL, J_{10}DL]	D_{33}: [J_{17}PL, J_{30}PL, J_{26}PL, J_{26}DL, J_{30}DL, J_{17}DL]	D_{1}: [J_{25}PL, J_{28}PL, J_{32}PL, J_{28}DL, J_{15}PL, J_{15}DL, J_{25}DL, J_{32}DL]
D_{34}: [J_{27}PL, J_{35}PL, J_{2}PL, J_{27}DL, J_{35}DL, J_{2}DL]	D_{32}: [J_{35}PL, J_{34}PL, J_{35}DL, J_{34}DL]	D_{37}: [J_{3}PL, J_{3}DL, J_{8}PL, J_{8}DL]	D_{33}: [J_{29}PL, J_{35}PL, J_{38}PL, J_{38}DL, J_{2}PL, J_{29}DL, J_{2}DL, J_{35}DL]
D_{4}: [J_{24}PL, J_{36}PL, J_{24}DL, J_{36}DL]	D_{35}: [J_{7}PL, J_{7}DL]	D_{34}: [J_{23}PL, J_{23}DL]	D_{36}: [J_{7}PL, J_{7}DL]
D_{3}: [J_{32}PL, J_{32}DL]	D_{3}: [J_{6}PL, J_{31}PL, J_{29}PL, J_{37}PL, J_{29}DL, J_{6}DL, J_{31}DL, J_{37}DL]	D_{36}: [J_{7}PL, J_{7}DL]	D_{38}: [J_{11}PL, J_{11}DL]
D_{5}: [J_{19}PL, J_{6}PL, J_{6}DL, J_{19}DL]	D_{36}: [J_{8}PL, J_{8}DL]	D_{7}: [J_{33}PL, J_{21}PL, J_{34}PL, J_{34}DL, J_{21}DL, J_{33}DL]	D_{41}: [J_{3}PL, J_{3}DL, J_{8}PL, J_{8}DL]
D_{7}: [J_{28}PL, J_{28}DL]	D_{5}: [J_{39}PL, J_{39}DL]	D_{18}: [J_{27}PL, J_{27}DL]	D_{17}: [J_{30}PL, J_{19}PL, J_{24}PL, J_{30}DL, J_{24}DL, J_{19}DL]
D_{9}: [J_{31}PL, J_{31}DL]	D_{43}: [J_{11}PL, J_{3}PL, J_{11}DL, J_{3}DL]	D_{43}: [J_{11}PL, J_{11}DL]	D_{12}: [J_{21}PL, J_{5}PL, J_{5}DL, J_{39}PL, J_{39}DL, J_{21}DL]
D_{36}: [J_{8}PL, J_{8}DL]	D_{17}: [J_{12}PL, J_{12}DL]	D_{9}: [J_{15}PL, J_{15}DL]	
D_{39}: [J_{11}PL, J_{11}DL]	D_{16}: [J_{2}PL, J_{2}DL]	D_{15}: [J_{36}PL, J_{6}PL, J_{22}PL, J_{6}DL, J_{22}DL, J_{36}DL]	
D_{18}: [J_{17}PL, J_{23}PL, J_{34}PL, J_{17}DL, J_{34}DL, J_{39}PL, J_{39}DL, J_{23}DL]	D_{9}: [J_{27}PL, J_{32}PL, J_{27}DL, J_{32}DL]	D_{13}: [J_{38}PL, J_{38}DL, J_{16}PL, J_{16}DL, J_{2}PL, J_{2}DL]	
D_{40}: [J_{7}PL, J_{7}DL]		D_{12}: [J_{28}PL, J_{32}PL, J_{24}PL, J_{24}DL, J_{28}DL, J_{32}DL]	
D_{16}: [J_{18}PL, J_{25}PL, J_{25}DL, J_{18}DL]			
D_{15}: [J_{5}PL, J_{5}DL]			
D_{14}: [J_{38}PL, J_{38}DL, J_{16}PL, J_{16}DL]			
D_{13}: [J_{14}PL, J_{14}DL]			
D_{47}: [J_{3}PL, J_{3}DL]			
Min Time: 2174	**Min Time: 2952**	**Min Time:2548**	**Min Time: 3883**
# of Drivers: 24	**# of Drivers: 18**	**# of Drivers: 19**	**# of Drivers: 15**

Table 8 continued

BSL Job Assignments
D_1: [J_{21}, J_{29}]
D_9: [J_{17}]
D_{20}:[J_2, J_{22}, $J_{25,}$ J_{38}]
D_{21}: [J_5]
D_{22}: [J_4, J_{16}, J_{18}, J_{23}, J_{24}, J_{28}]
D_{23}:[J_1]
D_{24}: [J_{12}, J_{39}]
D_{25}: [J_{13}]
D_{26}: [J_{19} , J_{20}]
D_{27}: [J_{14}, J_{15}]
D_{31}:[$J_{6,}$ $J_{34,}$ J_{35}]
D_{33}: [J_9, J_{10}]
D_{34}: [J_{26}, J_{27}, J_{30}, J_{31}, J_{32}, J_{33}, J_{36}, J_{37}]
D_{37}: [J_3, J_{11}]
D_{40}: [J_7, J_8]
Min Time: 3698
of Drivers: 15

heuristic algorithms, such as a Tabu Search and Simulated Annealing. We plan to compare these metaheuristics and refine our methods of minimizing both objectives simultaneously.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

Abounacer R, Rekik M, Renaud J (2012) An exact solution approach for multi-objective location transportation problem for disaster response. CIRRELT 26:1–32

Aneja YP, Nair PK (1979) Bicriteria transportation problem. Manag Sci 25(1):73–78

Baldacci R, Mingozzi A (2009) A unified exact method for solving different classes of vehicle routing problems. Math Program 120(2):347–380

Baldacci R, Bartolini E, Mingozzi A (2011) An exact algorithm for the pickup and delivery problem with time windows. Oper Res 59(2):414–426

Banos R, Ortega J, Gil C, Fernandez A, De Toro F (2013) A simulated annealing-based parallel multi-objective approach to vehicle routing problems with time windows. Expert Syst Appl 40(5):1696–1707

Brandao J (2011) A tabu search algorithm for the heterogeneous fixed fleet vehicle routing problem. Comput Oper Res 38(1):140–151

Chen W, Song J, Shi L, Pi L, Sun P (2012) Data mining-based dispatching system for solving the local pickup and delivery problem. Ann Oper Res 203(1):351–370

Choi E, Tcha D (2007) A column generation approach to the heterogeneous fleet vehicle routing problem. Comput Oper Res 34(7):2080–2095

Coello CA, van Veldhuizen DA, Lamont GB (2002) Evolutionary algorithms for solving multi-objective problems. Kluwer Academic Publishers, Dordrecht

Cordeau JF, Maichberger M (2011) A parallel iterated tabu search heuristic for vehicle routing problems. Tech rep, CIRRELT

Dantzig GB, Ramser JH (1959) The truck dispatching problem. Manag Sci 6(1):80–91

Dumas Y, Desrosiers J, Soumis F (1991) The pickup and delivery problem with time windows. Eur J Oper Res 54(1):7–22

Evans JP, Steuer RP (1973) A revised simplex method for multiple objective programs. Math Program 5:54–72

Gal T (1975) Rim multiparametric linear programming. Manag Sci 21:567–575

Holland JH (1975) Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor

Konak A, Coit D, Smith A (2006) Multi-objective optimization using genetic algorithms: a tutorial. Reliab Eng Syst Saf 91:992–1007

Lau H, Chan T, Tsui W, Pang W (2010) Application of genetic algorithms to solve the multidepot vehicle routing problem. IEEE Trans Autom Sci Eng 7(2):383–392

Lightner-Laws C, Agrawal V, Lightner C, Wagner N (2016) An evolutionary algorithm approach for the constrained multi-depot vehicle routing problem. Int J Intell Comput Cybern 9(1):2–22

Likaj R, Shala A, Bruqi M (2013) Application of graph theory to find optimal paths for the transportation problem. Int J Curr Eng Technol 3(3):1099–1103

Michalewicz Z (1995) A Survey of constraint handling techniques in evolutionary computation methods. In: Proceedings of the 4th annual conference on evolutionary programming. MIT Press, Cambridge, pp 135–155. https://cs.adelaide.edu.au/~zbyszek/Papers/p17.pdf

Miettinen KM (1999) Nonlinear multiobjective optimization. Kluwer Academic Publishers, Dordrecht

Muller J (2010) Approximative solutions to the bicriterion vehicle routing problem with time windows. Eur J Oper Res 202(1):223–231

Ombuki-Berman B, Hanshar T (2009) Using genetic algorithms for multi-depot vehicle routing. In: Pereira AMS, Tavares J (eds) Bio-inspired algorithms for the vehicle routing problem, pp 77–99

Pisinger D, Ropke S (2006) An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transp Sci 40(4):455–472

Pisinger D, Ropke S (2007) A general heuristic for vehicle routing problems. Comput Oper Res 34:2403–2435

Potvin JY, Rousseau JM (1995) An exchange heuristic for routing problems with time windows. J Oper Res Soc 46:1433–1466

Prakash S, Saluja RK, Singh P (2014) Pareto optimal solutions to the cost-time trade-off bulk transportation problem through a newly evolved efficacious novel algorithm. J Data Inf Process 2(2):13–25

Ropke S, Cordeau JF, Laporte G (2007) Models and branch-and-cut algorithms for pickup and delivery problem with time windows. Networks 49(4):258–272

Ropke S, Cordeau JF (2009) Branch and cut and price for the pickup and delivery problem with time windows. Transp Sci 43(3):267–286

Solomon M (1987) Algorithms for vehicle routing and scheduling problem with time window constraints. Oper Res 35(2):254–265

Tan TC, Chew YH, Lee LH (2006) A hybrid multi-objective evolutionary algorithm for solving vehicle routing problem with time windows. Comput Optim Appl 34:115–151

Vidal T, Crainic T, Gendreau M, Lahrichi N, Rei W (2011a) A hybrid genetic algorithm for multi-depot and periodic vehicle routing problems. Oper Res 60(3):611–624

Vidal T, Crainic T, Gendreau M, Prins C (2011b) A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time windows. Tech Rep 61:CIRRELT

Yu PL, Zeleny M (1974) The techniques of linear multiobjective programming. Revue Francoise d'Automatique, Informatique et Recherche Operationnelle 3:51–71

Zeleny M (1974) Linear multiobjective programming. Springer, New York

Zou S, Li J, Li X (2013) A hybrid particle swarm optimization algorithm for multi-objective pickup and delivery problem with time windows. J Comput 8(10):2585–2589