

Research Article

Inverting the Multiple-Assisting Tool Network Problem to Solve for Optimality

Robert Rich 

Industrial & Systems Engineering, Liberty University, Lynchburg, VA, USA

Correspondence should be addressed to Robert Rich; rkrich@liberty.edu

Received 20 October 2019; Revised 9 January 2020; Accepted 22 February 2020; Published 20 March 2020

Academic Editor: Yi-Kuei Lin

Copyright © 2020 Robert Rich. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Many network problems deal with the routing of a main tool comprised of several parallel assisting tools. These problems can be found with multi-tool-head routing of CNC machines, waterjets, plasma sprayers, and cutting machines. Other applications involve logistics, distribution, and material handling that require a main tool with assisting tools. Currently no studies exist that optimally route a main tool comprised of and fitted with multiple tools, nor do any studies evaluate the impact of adding additional capabilities to the tool set. Herein we define the network routing problem for a main tool comprised of multiple secondary tools. We introduce first principles to properly configure the main tool with the appropriate number of supporting tools such that that system is not overstatured. We invert the network geometry to extract the “best case” configuration for toolset configuration to include speed, range, and number of such that the system is lean. Our computational studies reveal that the theorems introduced herein greatly improve the overall system performance without oversaturating it with unused resources. In order to validate experiments, we define a mixed integer program and compare it to our metaheuristics developed for experimentation. Both the MIP and the metaheuristics herein optimally route a main tool with multiple assisting tools as well as the routing of a parcel delivery truck comprised of many drones.

1. Introduction

The use of multiple tools consisting of a main tool and several parallel resources in conjunction with the primary tool offers processing efficiencies in a range of applications. Semiconductor manufacturers are looking at ways to allow autonomous vehicles to interact such that overall travel time and traffic-congestion are reduced. Advances in multi-headed CNC and 3D printing-devices are expected to reduce print time significantly. In these designs, the main tool is confined to the primary axis of performance while assisting tool heads operate in parallel to the main tool. From a logistics standpoint, UPS, FedEx, and Amazon are evaluating the use of a single truck that is fitted with multiple parcel delivery drones to achieve greater efficiencies in delivery time.

The simplest example of this problem can be visualized as a mother with several children assisting with grocery shopping. The family has one grocery cart, and children are

dispatched in different directions around the store to retrieve items back to the cart while the parent routes through the store. It is obvious that there exists a set of optimal routes that the mother and children take that would optimize the total shopping time. However, it is less clear how the underlying relationships between the speed of the mother and children as well as their ranges of travel will also greatly impact any gained efficiencies in shopping. If a child is exceptionally slow, then the parent is better off to carry the child. If the child is exceptionally fast, but does not have the travel range, then the child's speed gives little advantage. Furthermore, that if the mother were to bring all her children into the store (in which case she has more than a dozen), she would be overwhelmed by idle children accompanying the overburdened shopping cart with much complaining due to their unemployed idle time.

So, it is easy to imagine that there exists a set of optimal assisting tool parameters (speed, range, and number of) that properly define the configuration of the overall system. For

simplicity, we adopt a parcel delivery truck comprised of and fitted with multiple delivery drones to describe the problem.

The main idea here is that there exists a geometric relationship between the size and delivery-density of the total operations area (Figure 1) and the density of truck-drone operations area. Specifically, the size and number of deliveries in the total operational area needs to be proportional to the size and number of deliveries in the truck-drone operations area for the most efficient operations. In words, a single truck-drone operation is described as a truck launching one or more drones, each drone traverses out to make deliveries, and then subsequently rendezvous with the truck at a downstream delivery location. In a single truck-drone operation, we know that there is a max coverage area (denoted by the elliptical circle) that n drones can reach (being constrained by range) and a total number of deliveries that can be performed. To achieve the maximum efficiencies, the truck-drone operation area density needs to be proportional to the total density of the larger delivery space. Since the larger space density is fixed, we adjust the drone parameters (speed, range, and number of) in order to match the truck-drone operations density to the larger delivery density. Thus, we advise to invert the network problem and solve the assisting tool parameters *a priori* such that they fit “the most” typical delivery scenarios for the delivery region in order to achieve maximum efficiencies.

In Section (2), we lay out the first principles geometric relationships between drone parameters and the truck-drone operations area. The notion here is that an operations is comprised of several truck-drone routing triangles. Each of these triangles fits within the area of an ellipse defined by truck speed and drone range. Thus, the elliptical area of the truck-drone operation and the number of deliveries within this space must be proportional to the larger delivery space delivery-density. To support the working theory, we conduct several computer experiments that reveal the production curve and “efficiencies gained” for systems described by the geometric relationships compared to systems defined by standard drone parameters of speed and range found in the literature. Section (3) reveals the results of these experiments.

An evolutionary algorithm (EA) was developed to perform empirical tests of our theory. The EA solves the truck-multidrone network routing problem by minimizing the total cumulative time based on either the max truck time or the max drone time for each set of launch-deliver-rendezvous operations cumulated in the total tour. Figure 1 shows \sim optimal routing solution for one of the problem scenarios. The algorithm is used to test the performance (total delivery time) of a *lean* configured truck-drone/s system to a *standard* configured system. The EA is validated by comparing it to the mixed integer program (MIP) shown in Section (5). The MIP is also validated by comparing it to known solutions and to brute force methods for smaller problem sets. As such, the validated metaheuristic solves problems optimally for nearly all of the smaller problem sets involving ten or less delivery stops and \sim optimal for problems as large as one hundred stops.

The sections comprised herein are as follow: Section (2) discusses working theory and theoretical insights. Section

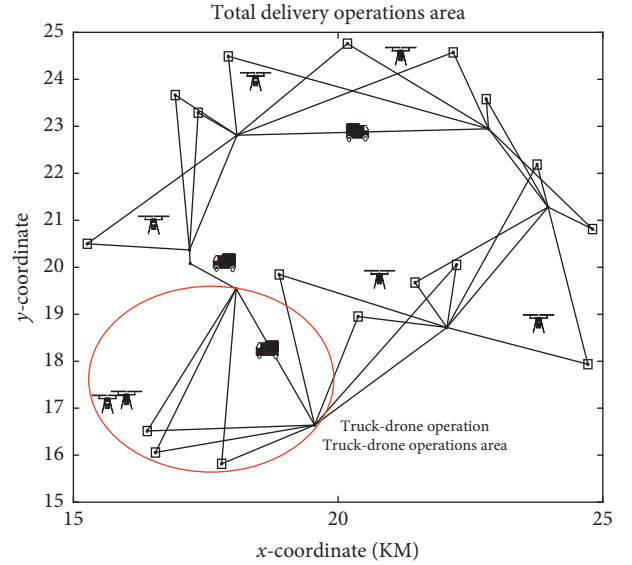


FIGURE 1: Total delivery operations area proportional to the truck-drone operations area.

(3) reveals the results of the hypothesis testing performed in relation to the fundamental theorems. Section (4) discusses the literature surrounding the network routing problems. Section (5) defines a tractable version of the mixed integer programming (MIP) used to validate metaheuristics that can be easily adopted to optimization software tools available to researchers. Section (6) formulates EA used to higher dimension truck-multidrone routing problems. Section (7) gives comparative studies: the first comparative study shows the performance of the evolutionary algorithms (EA-1 and EA-2) used to solve the optimal routing for empirical studies. The algorithms are compared to each other and to MIP-solved optimal solutions. The second study compares a *lean* configured system to a standard system configurations found in the literature. Section (8) concludes the research findings.

2. Theoretical Insights

2.1. Truck-Drone Performance Boundaries. The simplest truck-drone example involves a truck and one drone (Figure 2). For this, there are three nodes comprised of a depot v_0 and two delivery locations i and j . If the truck’s speed is scaled to one (1) and the drone’s speed is a factor of the truck speed, then an optimal configuration exists when the drone’s range (κ) equals the drone’s speed factor (α). So, as the truck travels out one unit distance from the depot (Figure 2) to a delivery (i), the drone also travels one (κ) units of distance (where $\alpha = \kappa$) from the depot to delivery location j for best configuration. The important thing here is that under the “best” configuration both the truck and drone return at the same time such that neither is waiting on the other. So, the best percent time (II) improvement that the dual system can possibly achieve over a *truck-only* system (truck having to deliver the two parcels itself and return to the depot tsp) is defined by [1]. Thus, the best case (II) of a truck-one-drone over a truck-only system is

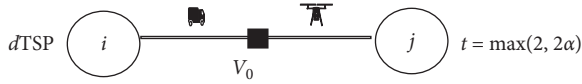


FIGURE 2: Best case for truck-drone (Agatz et al., 2015).

$$\Pi = 1 - \left(\frac{2}{2\alpha + 2} \right) = 1 - \left(\frac{1}{\alpha + 1} \right). \quad (1)$$

2.2. Truck-Multiple-Drone Performance Boundaries. The more complex case for a truck fitted with multiple drones necessarily involves more than three delivery locations, say (i, j, m, v_0) . For this argument, we assume that a well-configured system will involve the drones as much as possible throughout the delivery process. Therefore, if there are N delivery stops to be made during the day, then the “best case” number of deliveries allotted to each resources is (N) divided by the number of assisting drones (v) plus the main tool (truck): $(N/(v + 1))$. Furthermore, that delivery density plays an important role in determining the proper speed (α) and range (κ) factors in the analysis where total delivery density ρ is defined as the number (N) of required deliveries for the operation divided by the size of the geometric operating or delivery area (A) as in $\rho = N/A$. A densely populated delivery area requires more short-range drones while a sparse delivery density requires longer range and higher speed drones.

In order to construct a case for the *theoretical upper boundary* in potential performance improvement, we examine the geometry of a circle. We surmise that the best case (truck-multidrone) system for delivery time performance Π (over a truck-only) can be determined by launching drones along the perimeter of the circle and returning to the depot as shown in (Figure 3). Both drones and truck launch from the depot, traverse out, and return to the depot. Conversely, in an absolute worst case scenario, the truck would have to deliver to all delivery locations. This means that the worst-case truck route having no drones (i.e., hub-spoke route) will have to traverse from depot to edge of circle and then back to the depot.

Concretely, there exists a maximum and minimum yield (Π) on potential performance improvement for a truck and multiple drone/s system over the truck-only system. The worst case performance (lower boundary) is the truck having to make all the deliveries itself inside the circle (hub-spoke). The best case improvement requires the drones to traverse out to the radius of the circle (α) ; each makes a delivery, and then traverses back to the depot rendezvousing simultaneously with the truck.

Summarizing, if the truck traverses out one unit distance while the drones traverse κ units of distance where $(\kappa = \alpha)$, then the max time for any truck or drone is 2 units of time (Figure 3). Thus, a worst-case situation would exist if a truck had to visit all $v + 1$ deliveries on the circle (hub-spoke fashion). In such case, the absolute worst time the truck could encounter would be $2(1 + v\alpha)$ where (v) is the number

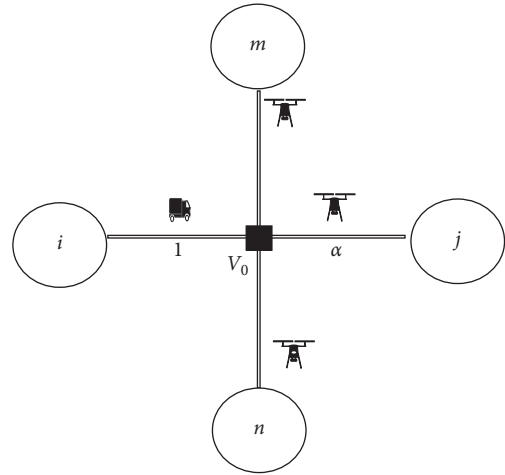


FIGURE 3: Practical best case truck-multiple drone delivery.

of deliveries around the circle (thus number of drones) at radius (α) and one delivery made by the truck (1). Conversely, the best case (Π) improvement for truck-multidrone (v) over a truck-only system (Π) is

$$\Pi = 1 - \frac{2}{2 + 2v\alpha} = 1 - \frac{1}{1 + v\alpha}. \quad (2)$$

2.3. Truck-Multidrone Operations Area. Expanding on the theoretical delivery problem means that trucks and each drone must traverse through the operating space and, therefore, do not return immediately to the depot. As such, drones travel using a triangular path to launch and deliver and then rendezvous with the truck downstream where the two legs of the drone’s triangular path cannot exceed the drone’s total range kappa (κ) . Since the truck and the drone do not have the same speed, we have to adopt a configuration using an ellipse (as opposed to a circle) (Figure 4) such that the foci f points on the ellipse denotes the truck’s delivery locations, while a triangle formed inside the ellipse (using the foci points and edge of ellipse) denotes any of the potential launch-delivery-rendezvous operation for any of the drones. Thus, the drones are confined by range to operate within the area of the ellipse where they are launched. They are launched from the first foci point and must then be retrieved at the second foci of the ellipse by the truck in a perfect scenario. Therefore, it is easy to imagine that there exists a proper sizing of the ellipse whereby the truck’s speed and distance (from foci to foci) must be aligned with the drone’s range, drone’s speed, and the number of drones. The number of drones is based on the density of deliver locations within the elliptical area.

The first interrelationship requires that in order for the drone to traverse out to its maximum distance and return back to the truck without any idle time, it is require that the truck-only travel kappa divided by alpha κ/α distance. Therefore, the distance between the two foci is a factor of both drone range and drone speed. For example, if the drone travels twice the speed of the truck $(\alpha = 2)$ and the drone can travel a total of 10 km $(\kappa = 10)$, then the truck should only

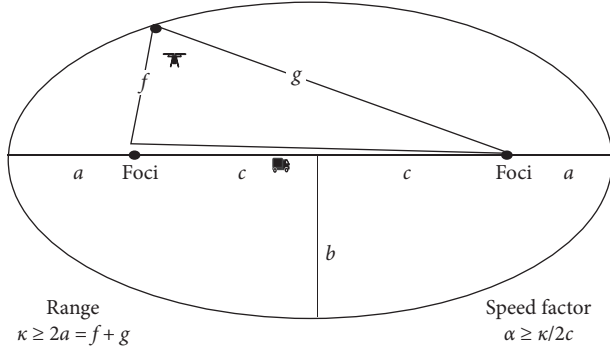


FIGURE 4: Example of truck-drone delivery inside an ellipse.

travel 5 km before having to retrieve the faster drone having travelled 10 km in the same time period if there is to be no idle time. As such, we limit our elliptical area such that from foci to foci is denoted as $(2c)$ must be equal to (κ/α) for a perfectly lean scenario.

The second interrelationship requires that the total distance that the drone can travel is limited by the drone's range. Therefore, the drone should be limited to twice the axis of the major axis ($\kappa = 2a$). For relaxation of the problem, drone range must be greater than twice the major axis ($\kappa \geq 2a$), and truck travel distance $(2c)$ at speed one is consistent with drone speed factor ($\alpha \geq \kappa/2c$) where the truck speed s_T and drone speed s_D are defined by speed factor alpha $\alpha = s_D/s_T$.

Moreover, delivery density directly reflects the number of drones assigned to deliver within the elliptical area. We can imagine that if given the total delivery density (ρ) for the entire area is comprised of total deliveries (N) and a total area (A), then delivery density would be computed as $\rho = (N/A) \text{ km}^2$. Under a best-case delivery density scenario, the densities of two of the ellipses (ρ') would need to be proportional to the total delivery density for the problem space as in $\rho = \rho'$. Thus, we define the truck-drone operations area as that of two overlapping ellipses joined at the foci (minus intersection) as

$$A' = 2\pi ab - [\sqrt{\Phi} \pi(a-c)^2], \quad (3)$$

where (a) is the major axis, (b) is the minor axis ($a > b$), (c) is the foci distance from centre to foci, and (Φ) is the golden ratio (~ 1.618) shown in figure (Figure 5).

Furthermore, based on the interrelationship between truck speed, drone speed, and drone range, the total distance between two foci $(2c)$ is approximately equal to (κ/α) . Therefore, we would expect to see N' deliveries within the dual elliptical area A' . Thus, we compute the number of expected deliveries inside any of our elliptical drone operational areas as $N' = A'(\rho)$ for a "practical best-case" scenario.

In summary, we can surmise for practical best-case situation that twice the distance from the centre of the ellipse to the foci is a factor of both drone range (κ) and drone speed (α) as in $(2c = \kappa/\alpha)$ so that the truck and drone's operation reduces idle time and that drone range is greater than or equal to the two legs of the triangle ($\kappa \geq 2a$)

(Figure 5). In such case, the "practical best case" for the number of drones (ν) fitted to the truck is based on N' deliveries needed to be within the combined area of the two ellipses (A'). Also, since the truck can make three of the total (N') deliveries, then the number of drones required to be fitted to the truck under "best case" scenario is $\nu = (N' - 3)$.

Using the area of two ellipses A' and the appropriate "practical best case" density ρ' for a given system, the problem can be inverted by knowing *a priori* the delivery density ρ and then forcing ρ' to be equal to ρ by selecting proper parameters for range, speed, and number of drones. Thus inverting the problem, we solve for the drone range, drone speed, and number of drones that would produce a "practical best case" geometric design for the problem.

2.4. Theoretical Summary. In summary, the hypothesis for *dvtsp* is as follows.

2.4.1. Maximum Theoretical Upper Boundary. There exists a maximum *theoretical upper boundary* for best possible percent time improvement (Π) over a truck-only system that will never be exceeded without special construction of problem scenario based on the assisting drone's time improvement factor $(1 + \alpha\nu)$ where α denotes the speed factor and ν denotes the number of drones assigned.

$$\Pi < 1 - \frac{1}{1 + \alpha\nu}. \quad (4)$$

2.4.2. Lower Boundary. The *theoretical lower boundary percent improvement* (worst case) for a truck-multiple-drone system over a single truck only solution is $\Pi = 0\%$ based on the fact that a truck can perform all the deliveries by itself and default to a single truck route, thus no time improvement.

2.4.3. Improvement. Improvement Π moves toward the *practical upper boundary* as the delivery density of the given operating area (ρ) scenario moves toward the calculated theoretical "best case" delivery density (ρ'). As such, the problem can be inverted. As the absolute value between the problem delivery density (ρ) and the "practical best case" delivery density (ρ') are minimized, the best case (optimal) drone speed factor α , drone range κ , and number of drones ν based on the geometric layout of the ellipse emerge. Due to the stochastic nature of most problem scenarios, the optimal drone speed, drone range, and number of drones is a lower boundary optimal.

$$\min|\rho - \rho'|, \quad (5)$$

S.T.

$$\kappa \geq 2a, \quad (6)$$

$$\nu = N' - 3, \quad (7)$$

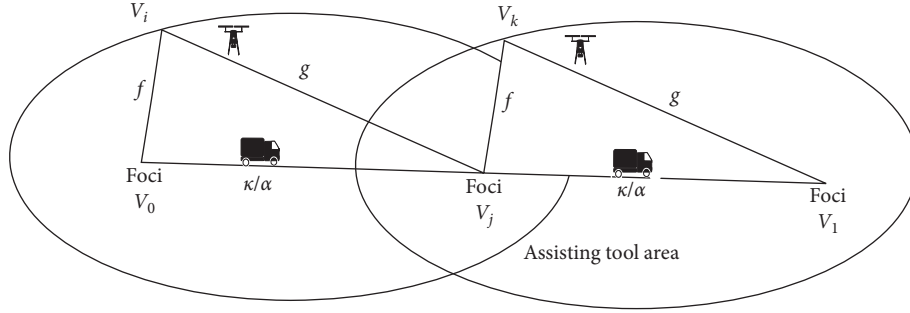


FIGURE 5: Best-case geometric layout for truck-multiple-drone scenario.

$$\alpha \geq \frac{\kappa}{2c}, \quad (8)$$

$$N' = A'(\rho), \quad (9)$$

$$\rho = \frac{N}{A}, \quad (10)$$

$$\rho' = \frac{N'}{A'}, \quad (11)$$

$$A' = 2\pi ab - [\sqrt{\Phi} \pi (a - c)^2], \quad (12)$$

$$c^2 + b^2 = a^2, \quad (13)$$

$$\alpha = \frac{s_D}{s_T}, \quad (14)$$

$$s_D > s_T, \quad (15)$$

$$s_{D_{LB}} \leq s_D \leq s_{D_{UB}}, \quad (16)$$

$$\kappa_{LB} \leq \kappa \leq \kappa_{UB}, \quad (17)$$

$$\nu_{LB} \leq \nu \leq \nu_{UB}, \quad (18)$$

$$\rho, \rho' > 0, \alpha, \kappa, \nu \geq 1. \quad (19)$$

For the proper configuration of the minimization problem, let (a) be the major axis, let (b) be the minor axis, and let (c) denote the distance from foci to the centre of the ellipse. Furthermore, let alpha (α) denote the speed factor of drone as related to the truck's speed of one (1). Let kappa (κ) denote the range of the drone whereby the range must be greater than twice the major axis of the ellipse; let nu (ν) be the number of drones assigned to assist the truck. Let the area of one ellipse be $A_{\text{ellipse}} = (\pi ab)$ and the combined area of two ellipses $(A_{\text{two}} = 2\pi ab)$. Since the ellipses are intersecting at a foci point (c) , then let the area of the two combined ellipses be reduced by the area of one of the intersected ellipses $[\sqrt{\Phi} \pi (a - c)^2]$ where (Φ) is the golden ratio (~ 1.618).

Concretely, equation (5) minimizes the absolute distance between the problem delivery density and the practical best case delivery density of the two ellipses. Equation (6) forces the range of the drone (κ) to be greater than that of the two legs of the triangle inscribed within an ellipse. Equation (7) solves for the optimal number of drones to be fitted to the truck by allowing the truck the opportunity to deliver to three of the delivery vertices found within the area of the two ellipses. Equation (8) guarantees that the interrelationship between truck speed, drone speed, and range are configured such that for each operation, (launch-deliver-rendezvous) the truck and drone arrive back at nearly the same time to reduce potential idle time for either truck or drone. The truck travels $(2c)$, while the drone travels a max of $(2a)$ and the interrelationship gives enough flexibility to solve the problem by establishing that $(2c \geq \kappa/\alpha)$. Equation (9) calculates the expected number of deliveries found within the area of two ellipses (A') . Equation (10) defines the delivery density as the number of deliveries per total area of operation. Equation (11) calculates the delivery density of the two ellipses. Equation (12) defines the area of the two ellipses minus their intersecting area. Equation (13) establishes the geometric relationship of the triangle inscribed within an ellipse where (a) is the hypotenuse. Equation (14) solves for drone speed factor alpha as the ratio of the drone's speed to the truck's speed. Equations (15)–(18) are boundary equations for the drone's speed, truck's speed, drone range, and the total number of drones. Equation (19) ensures that delivery density of ellipses is greater than zero while drone range, drone speed factor, and number of drones are greater than one.

3. Hypothesis Testing

To test our hypothesis, thousands of random experiments were conducted by holding range (κ) , speed factor (α) , and number of drones (ν) stationary while perturbing number of deliveries (N) over the delivery area (A) . In essence, the optimal delivery density (ρ') is precalculated based on drone parameters (α, κ, ν) . For each experiment, delivery density (ρ) changes based on changing number of deliveries (N) within the delivery space (A) . The experiments are designed to test the performance of a truck multidrone system against a truck-stand-alone (no drones) system. In the experiments,

we see that, when the delivery density (along the x -axis) reaches the optimal density line on the graph, then the overall performance of the system improves until it begins to saturate. In words, the denser the deliveries are in a delivery space, the better the system performs until saturation. Using the fundamentals mentioned above, we can now calculate the optimal density (ρ') at which the system saturates. The figure below (Figure 6) reveals low performance when delivery density is sparse. The system then begins to improve up to and including the optimal delivery density before the resources begin to saturate, and improvement curve asymptotically reaches its peak performance.

The figure shows that the fundamental relationships established in Section 3 accurately predict the system performance. Using the fundamentals as a guide, the experiments revealed that as the delivery density (ρ) approached the optimal density (ρ') the overall performance of the system improved. Concretely, that as density $\rho = N/A$ moved closer toward $\rho' = N'/A'$ the system improved until saturation. As such, we can surmise that the geometric layout of the ellipses comprising the delivery density ρ' is an accurate depiction of the practical best case geometric configuration for the system. Therefore, the minimization problem established in Section 3 is a “practical best case” method to derive speed factor alpha (α), drone range kappa (κ), and number of drones nu (ν) to be fitted to a truck based on a typical delivery situation.

4. Literature

A large body of literature exists about the traveling salesman problem (tsp) and the vehicle routing problem (vrp). Many approaches and variations to both can be found in surveys, reports, and papers [2–4]. As a general rule, the vrp problem extends the tsp problem by adding additional constraints. These constraints are comprised of time windows, priorities, range, loiter times, permissible-to-vehicle type route segments, load configurations, traffic patterns, etc. [5, 6]. Originally, Danzig and Ramser [7] investigated the vehicle routing problem. Later, Clarke and Wright [8] proposed an effective greedy heuristic which subsequently followed by several techniques or models involving exact and heuristic approaches to solve the extensions and variations of the vrp. An extensive survey can be found in [9] with exact methods to solve various routing problems.

While many variants to the tsp problem exist in the literature including multivehicle, customer pickup and delivery problem, multiple synchronization constraints [10], multiple depot vehicle scheduling problem [11], and many-to-many milk run routing problem [12], there exists only a handful of studies concerning the *truck-drone* problem. Moreover, the truck multidrone problem is a relatively new paradigm not found in the recent literature. Although the truck-drone problem has been addressed by research thus far, there is no work found on the truck- n -drone problem.

4.1. Truck-Drone Problem. The operational aspects and the formulation of the truck-drone problem were first shown by

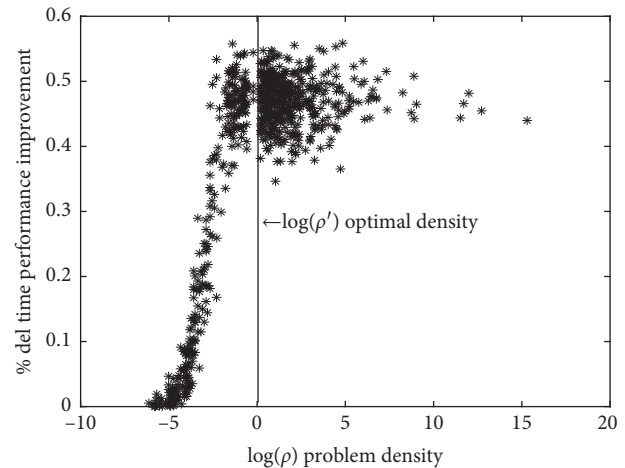


FIGURE 6: As problem delivery density nears optimal density for the system, performance improves.

Murray and Chu [13]. Shortly thereafter, Agatz et al. [1] considered a close, but slightly altered version of the problem. Murray and Chu identify two ways that drone can be useful in delivery: drones can be launched and recovered from the depot (parallel drone scheduling traveling salesman problem PDSTSP) or drones can assist the truck in a parallel operation (launch-deliver-recover) as in the “flying sidekick tsp” FSTSP. Both Murray and Agatz define the material aspects of the truck-drone problem and discuss the mixed integer formulation for the optimal min-time route. They consider the drone is constrained in range, capacity, and speed. Agatz only considers a slightly altered version of the FSTSP and does not address the PDSTSP. Agatz, like Murray, considers the truck-drone in tandem as a team whereby the truck launches the drone, traverses to a separate delivery location from the drone, and then rendezvous with the drone again. However, the main difference between the approaches is that Agatz et al. [1] proposed that the drone and truck traverse along the road network system, a constraint not enforced by [13]. They do this to facilitate construction of heuristic approaches with approximations that guarantee a bound on the maximum achievable gain of the delivery system over a “truck-only solution.” Murray and Chu [13] formally define the *flying sidekick traveling salesman problem (FSTSP)* as an NP-hard problem. Their study suggests a mixed integer programming (MIP) as well as a metaheuristic approach. They also consider a second similar hub-type problem that addresses the case where the customers are close enough to the depot to be serviced directly from the depot by the drone while the truck delivers to farther reach areas. This is denoted as the *parallel drone scheduling TSP (PDSTSP)*. More recently, Agatz et al. [14] presented an exact solution approach for the truck-drone problem denoted as the TSP-D based on dynamic programming. They conducted experimental studies using the different dynamic programming heuristics which indicated that the dynamic-programming-problem could solve larger problems better than mathematically programming approaches found in the literature.

Although existing research shows mathematical formulations and discusses various methods to approach the metaheuristics, the main *gaps* are that no studies reveal details concerning the expected percent improvement for a given configuration of the truck-drone to various delivery densities. Specifically, there are no studies that investigate the tradeoff between number of drones, speed factor, and drone range.

4.2. Truck Multiple Drones. Ferrandez et al. [15] introduce a method using *k-means* clustering to cluster deliveries based on distance and then to route the truck around to each of the clusters using a tsp approach. This cluster-first-route-second approach also introduces a genetic algorithm to optimally tsp route the truck through center mass quasinodes created from each of the clusters. In the described scenario, the truck may loiter at any center mass cluster node (hub) while the drone/s deliver to customers within the cluster region using a minimum spanning tree (MST) hub-spoke type of approach. Additionally, analysis was conducted on the tradeoffs between total time and total energy consumed by the truck-drone team as a factor of number of truck stops.

Our study herein is different than Ferrandez et al. [15]. Here, we require that the truck launch multiple drones then proceed to the next delivery location prior to rendezvous. It is an improvement in time over a hub-spoke method. Furthermore, we use an MIP metaheuristic to solve for optimal network routing while minimizing total delivery time. Furthermore, we introduce the concepts of “practical best case” or lean configuration which has not been addressed by any author for any of the main tool/assisting tool problems.

Other gaps in the literature are that no research deals squarely with the truck-multidrone problem from a mathematical modeling or a metaheuristic perspective. Moreover, there exists no study that discusses of the tradeoffs or selection of system parameters α, κ, ν with changes in the operating area or density of the delivery area. There is obviously a delta time improvement that can be determined using two or more drones or proper selection of range and speed when these are used in conjunction with a truck addressed herein.

5. Mixed Integer Program (MIP)

The MIP for the truck-multiple (ν) drones d traveling salesman problem ($d\nu$ tsp) solves for the optimal network routing of a truck with multiple ν drones while minimizing total time. It was developed and used herein as the basis to analyze and evaluate the effectiveness of the “practical best case” configuration. Material elements of the $d\nu$ tsp are described as a network graph $G = (V, E)$ where V denotes the customers-delivery-stops and E the edges between stops. Each vertex $V = \{1, \dots, |V|\}$ where $n = |V|$. Edge E is described by two vertices $\{i, j\}$, and two edges or an operation is described by two vertices $\{i, j\}$ for a truck indicating launch and recovery. Furthermore, the binary variables

$(x_{ij}, y_{ik}y_{kj})$ denote a truck and the two edges used by a drone in an operation if an operation exists between (i, j) . The binary variable $x_{ij} = 1$ if truck traverses edge $E(i, j)$. If a drone is dormant, it is assigned no vertices for a null operation. As in the truck-drone problem, the first and last vertices of an operation must always be the same for the truck and all drones. The distance matrix is denoted as D where $d_{ij} \in D$ is the cost or distance of traversing edge $E(i, j)$. The subtour elimination variable U_j ensures that sequence of j follows i in the event that $E(i, j)$ is part of the solution; as such, it restricts subtour formations. The minimum of max (of max) objective function minimizes the max time of either the truck’s time or any of ν drone’s time for an operation. As such, z_{ij} is used to evaluate each vehicle’s max-time where i, j are the first and last nodes of the two visited nodes for truck (x_{ij}) as well as the first and last of three nodes traveled by any drone as in $(y_{ik}y_{kj})$. Since z_{ij} is greater than or equal to the max time of the truck or any of the ν drones, it forces z_{ij} to evaluate the max (of max) of all vehicle’s time.

$$\text{minimize } Z = \sum_{i \in V} \max_{j \in V} z_{ij}, \quad (20)$$

S.T.

$$\begin{cases} z_{ij} \geq d_{ij}x_{ij}, \\ z_{ij} \geq \frac{y_{ik}y_{kj}(y_{ik}d_{ik} + y_{kj}d_{kj})}{\alpha}, \\ \forall i, j, k, i, j, k \in V, \end{cases} \quad (21)$$

$$\sum_{i \in k} x_{ik} = \sum_{j \in k} x_{kj}, \quad \forall k \in V, \quad (22)$$

$$\sum_{i \in V} x_{ik} + \sum_{i \in V} y_{jk} \geq 1, \quad \forall k \in V, \quad (23)$$

$$\sum_{i \in V} x_{ki} + \sum_{j \in V} y_{kj} \geq 1, \quad \forall k \in V, \quad (24)$$

$$\sum_{i \in V} x_{ki} = 0, \quad \forall k \in V, \quad (25)$$

$$\sum_{i \in V} y_{ki} = 0, \quad \forall k \in V, \quad (26)$$

$$x_{ij} \geq y_{ik}y_{kj}, \quad \forall i, k, j \in V, \quad (27)$$

$$x_{ij} + y_{ij} \leq 1, \quad \forall i, j \in V, \quad (28)$$

$$\begin{aligned}
U_j &\geq U_k + x_{kj} - (N-2)(1 - x_{kj}) + (N-3)x_{jk}, \quad \forall k, j \in V, \\
j &> 1 \\
j &\neq k
\end{aligned} \tag{29}$$

$$\begin{aligned}
U_k &\leq N - 1 - (N-2)x_{1k}, \quad \forall k \in V, \\
k &> 1
\end{aligned} \tag{30}$$

$$\begin{aligned}
U_k &> 1 + (N-2)x_{k1}, \quad \forall k \in V, \\
k &> 1
\end{aligned} \tag{31}$$

$$3 \leq \left\lceil \frac{n}{\nu} \right\rceil \leq N \leq n, \tag{32}$$

$$y_{ik}y_{kj}(y_{ik}d_{ik} + y_{kj}d_{kj}) \leq \kappa, \quad \forall i, k, j \in V, \tag{33}$$

$$d_{ij} \geq 0, \kappa_{LB} \leq \kappa \leq \kappa_{UB}, \alpha_{LB} \leq \alpha \leq \alpha_{UB}, \quad \forall i, j \in V, \tag{34}$$

$$x_{ij} \in \{0, 1\}, \quad \forall i, j, i, j \in V, \tag{35}$$

$$y_{ij} \in \{0, 1\}, \quad \forall i, j, i, j \in V. \tag{36}$$

In the *dvts*p construction, equation (20) minimizes the sum all the max of truck or drone edges (i, j) to all operations described as a truck route $x_{ij} = 1$ and potentially a triangle drone operation described by $(y_{ik}y_{kj} = 1)$. Equation (21) forces the max (worst case) of either a truck edge or the drones' two-edges to be in the resultant minimization calculation for objective Z . The distance for a truck edge (i, j) is denoted as $d_{ij}x_{ij}$ whereas the drone's triangular operation distance is denoted by $y_{ik}y_{kj}(d_{ik}y_{ik} + d_{kj}y_{kj})/\alpha$ where α is the drone speed as a factor of truck speed s where $s = 1$. In essence, this equation serves as a mini-sum of max (of max) construction. Equation (22) constrains the truck portion of the route $x_{ij} = 1$ to make a circuit or at a minimum manifests a sub-tour. Equations (23) and (24) ensure each city has at least one truck or one drone visiting the city. Equation (23) establishes that each city must be visited (entered), while (24) constrains that each city visited must then be exited. Equations (25) and (26) prohibit any truck or drone from loitering as in x_{ii} or y_{ii} . Equation (27) requires that any drone operation triangle $y_{ik}y_{kj}$ has an associated truck launch and recovery x_{ij} . Equation (28) forces that any drone operation edge is not already a truck edge. Equation (29) forces a utility variable U_j and U_k to properly sequence truck route segments x_{jk} such that sequence number j follows k so that no subtour is formed in the truck sequence of the truck portion of the route. Equation (30) and (31) work in conjunction with equation (29) to set vertex (20) as first and vertex $N \mid (3 \leq n/\nu \leq N \leq n)$ as the last of the truck-route segments. Equation (32) sets the minimum number of truck cities visited to be greater than or equal to three $(3 \leq n/\nu \leq N \leq n)$ but less than total nodes available, n . Equation (33) limits any drone operation to be less than the

constrained range of the drone (κ) if those two edges (i, k) and (k, j) are used as a drone operation $y_{ik}y_{kj} = 1$. Equations (34)–(36) set minimum requirements for distance matrix, range, and speed factor as well as sets truck variable x_{ij} and drone variable y_{ij} as binary integers.

6. Evolutionary Algorithm

The tournament-based evolutionary algorithm (EA) here adopts a *cluster-during-routing* approach to solve the truck-drone problem. More accurately, it assigns both *truck* and *drone* labels during the routing process. Currently, there are no other algorithms found in the literature. The algorithm denoted as EA1 creates a population matrix of randomly permuted routes whereby each node in a tour is evaluated as a potential drone-delivery node unless that node is out of range. Since a population of many randomly generated tours is evaluated simultaneously, any node not within drone range is autoassigned and labelled *truck*; otherwise, the algorithm labels it drone.

6.1. Evolutionary Algorithm (1) Steps. The EA randomly permutes a population P of m tours where each tour denoted as a genome sequence $(1, 2, \dots, n)$ for n delivery nodes in the tour. It determines the fitness for each population member (tours) based on total tour delivery time. All fitness times are saved for seed tournament. The total population is then divided into groups of five tours each to conduct a set of seed tournaments. For each of the groups, the best member within the seed group (of the five) is chosen as the single gene to mutate for the remaining four members of the seed group. Gene mutation (tour mutation) first copies the fittest member of the group of five within the seed tournament to replace the four less fit members. Each of the four less fit members (now identical to the fittest) is then slightly mutated to improve fitness. For each of the four, mutations are comprised of only one of (a) randomly selecting and swapping two nodes within the tour, (b) reverse ordering of the tour between two nodes, (c) sliding a tour segment down between nodes to left or right, and (d) replacing the last node in the tour with any other node. The algorithm iterates repeatedly until convergence, or a terminating condition is met based on a predetermined budget, tolerance, or a saturation found in improvements (Algorithm 1).

6.2. Evolutionary Algorithm (2). The second metaheuristic denoted as EA2 is nearly identical to EA1, but differs by performing a cost check before assigning a drone to the job. Concretely, EA2 performs a calculation to determine if it is more efficient to assign the truck or the next drone. If the truck is more efficient, the greedy algorithm assigns the truck and foregoes assigning a drone.

Both algorithms use an approach similar to simulated annealing. A seed tournament genetic algorithm's strength lies in the ability to overcome local optima by retaining multiple paths (or seeds) that simultaneously advance toward optimization. This ideology is especially critical for

network routing problems. Furthermore, because there are multiple members of the population within a seed tournament, the algorithm allows for various mutation methodologies to be performed on the members of the seed tournament. In this case, the *random swap* (or *pairwise swap*), *flip*, and *slide* mutations have proven to be robust, fast, and extremely accurate for problems involving permutations whereby order matters.

The performance of the evolutionary algorithm is based on the underlying theoretical principles: (a) by initializing a relatively large population (i.e., $5n$) of randomly permuted tours, multiple tracks are maintained toward optimal convergence (seed tournaments). These multiple, but different, paths slowly converge and, thus, increase the probability of an optimal convergence. (b) By saving the fittest gene in a seed and then slightly perturbing (mutating) the best gene (tour) found in the seed group ensure the solution never gets worse while promoting improvements at each iteration. (c) By autoassigning the drone to any “within range” node, the use of the drone is maximized throughout the routing process while simultaneously reducing the truck’s overall tour length. The risk of assigning the wrong node to a drone is mitigated by multiple seeds. (d) Multiple path (seed) random search is much faster than having to calculate the greediness or the exactness of each neighborhood within reach as in other algorithms. Therefore, the algorithm relies on computational speed and iterations without the burden of unnecessary calculations.

Both EA1 and EA2 use a *route-during-clustering* approach. This approach has several advantages over other algorithms. All algorithms found in the literature with the exception of [16] use route-first, cluster-second or a cluster-first, route-second approach. This results in a two-phase algorithmic process whereby the first phase solves for routing (as in travelling salesperson routing (tsp) or minimum spanning tree, (mst) routing); the second phase relying extensively on solutions found in first phase performs swaps between truck or drone delivery. The best known algorithms for *tsp* (or *mst*) routing are on the order of $O(n^2 2^n)$ where n is the number of nodes in the route. The second phase in a *route-first-cluster-second* algorithm relies heavily on the initial routing solutions being in the vicinity of a truck-drone solution. As such, the second phase is equally complex as the first phase with the disadvantage of being locked into a potentially suboptimal routing scheme especially when drone configurations change.

Conversely, the *route-during-clustering* algorithm performs the routing and the selection of the truck or drone for the next operational move, segment of the route. As such, the complexity of the algorithm is based on the number of iterations (n^2), the population size ($4n$), and the number of nodes (n) found in the tour. In such case, the algorithm is consistent with the order of $O(n^2 \times 4n \times n)$ or $O(4n^4)$. A computer with an Intel core (i5) and 1.70 GHz CPU, comprised of 50 nodes and 3 drones, requires approximately ~ 25 million iterations or ~ 35 seconds for convergence.

The overall complexity of the truck-multidrone problem denoted as (*dvtsp*) can be analysed from a *brute force* methodology. To evaluate every potential solution for a ten

node problem would require slightly less than $((10!/2) \times 2^{10})$ permutations or approximately $\sim 1.86e + 09$ routes and binary drone assignments. Every solution requires two genes, a gene describing the order of the tour ($10!/2$) and a gene to describe every possible truck or drone assignments 2^{10} . For larger problem sizes, say $n = 50$ would require $((50!/2) \times 2^{50}) = 6.8486e + 79$ iterations translating to approximately 2.8e61 seconds computer time or $\sim 7.1e53$ years on a 1.70 GHz CPU. For such method, the number of iterations to accomplish would exceed the time in the universe.

7. Comparison Studies

Two different computational studies are conducted here. The first study analyses the performance of the metaheuristics against the optimal solution based on MIP and brute force solution. The second study analyses the performance of the system model parameters (α, κ, ν) against arbitrarily (standard) assigned system variables.

In the first study, the performance of the two metaheuristics described in Section 6 is analysed. The two metaheuristics are compared against themselves as well as compared with the guaranteed optimal results obtained from the mixed integer program (MIP). The comparison study analyses four different performance factors: (1) the average delta delivery-time of the metaheuristic versus the optimal delivery time found using the MIP, (2) the max delta (delivery-time) between the metaheuristic versus the optimal time of MIP, and (3) the total number of metaheuristic experiments that resulted in optimal results out of ten studies, and the average computer solution time for the metaheuristic to compute the optimal solution.

The second study analyses the performance of properly (optimally) configured system parameters (α, κ, ν) as described in Section 2 (theoretical insights: system model) against standard parameters found in the literature ($\alpha = 2, \kappa = 10, \nu = 3$). In many cases, the literature arbitrarily chooses system parameters as drone speed factor set at two times the speed of the truck ($\alpha = 2$) and an arbitrary drone range set at ten kilometres ($\kappa = 10$) based on current technical capabilities. The system comparison study here analyses three performance metrics: (1) the total delivery-time obtained by a properly configured (lean) system versus the total delivery-time obtained by arbitrary system parameters (α, κ, ν), (2) the overall utilization of the drones throughout the delivery: a properly configured (lean) system versus the utilization of the drones from a system with arbitrary parameters, and (3) the under or oversaturation of the system determined by the average wait or idle time (by the truck or by drone) for all operations.

7.1. Computational Study 1: Metaheuristic Performance to Optimal Solution. For the sake of simplicity, a plane of (x, y) coordinates is uniformly randomly generated from the Cartesian coordinate system with Euclidean distance between the nodes. Concretely, all nodes for problem comparisons herein were sampled from the uniform distribution from $\{0, 1, 2, \dots, 30\}$. For each metaheuristic

```

DATA : Random Population matrix tours where  $Route_i \in Population$ 
RESULT :  $Route_{opt} \sim optimal\ tour \in Population\ of\ n\ delivery\ stops$ 
FOR iterations in budget LOOP
  FOR each (i) population member LOOP
     $Route_i \leftarrow$  a tour within the populations of tours
    launch  $\leftarrow$   $Route_i$  (1) first node;
    rendezvous  $\leftarrow$   $Route_i$  (2) second node;
    candidate  $\leftarrow$   $Route_i$  (3) third node;
    drone count  $\leftarrow$  0;
    route time  $\leftarrow$  0;
  WHILE candidate  $\leq$  total delivery nodes DO
    truck op time  $\leftarrow$  get truck time (launch, rendezvous);
    drone op time  $\leftarrow$  get drone time (launch, candidate, rendezvous);
    drone op dist  $\leftarrow$  get drone distance (launch, candidate, rendezvous);
    IF candidate  $\geq$  total delivery nodes THEN
      IF drone count  $\geq$  number drones avail. OR drone op dist.  $>$  drone range THEN
        Max op time = max (truck op time, drone op time, multiple drone op time);
        route time  $\leftarrow$  route time + max op time + truck op time;
        route time  $\leftarrow$  route time + get truck time (candidate depot);
      ELSE (drone makes last delivery)
        Max op time = max (truck op time, drone op time, multiple drone op time);
        Route time  $\leftarrow$  route time + max op time + get truck time (rendezvous, depot);
      END IF
      BREAK while loop;
    END IF
    IF drone count  $\geq$  number drone avail OR drone dist.  $>$  drone range THEN
      Max op time  $\leftarrow$  max (truck op time, multiple drone op time);
      Route time  $\leftarrow$  route time + max op time;
      Launch  $\leftarrow$  rendezvous;
      rendezvous  $\leftarrow$  candidate;
      candidate  $\leftarrow$  min ( $Route_i$ (candidate + 1), total number stops);
      drone count  $\leftarrow$  0;
      multi drone op time  $\leftarrow$  0;
    ELSE (assign delivery to drone)
      max op time  $\leftarrow$  max (truck op time, drone op time, multiple drone op time);
      route time  $\leftarrow$  route time + max op time
      candidate  $\leftarrow$   $Route_i$ (candidate + 1);
      drone count  $\leftarrow$  drone count + 1;
      multi drone op time(drone count)  $\leftarrow$  drone op time;
    END IF
    List of all Route Times (p)  $\leftarrow$  route time;
  END WHILE
END FOR (each population member)
  Population  $\leftarrow$  randomly shuffle among the tours in population, keep routes intact
FOR each of five tours in population LOOP through, keep tournament winners, mutate losers
  Best time, Id, Best Route  $\leftarrow$  Get Fittest Member tour of tournament of 5;
  Overwrite each of the four less fit tours with the fittest member;
  Mutate first of the four less fit tours by random swap;
  Mutate second of the four less fit tours by random segment slide;
  Mutate third of the four less fit tours by random segment flip;
  Mutate fourth of the four less fit tours by swapping last node in tour with any other
  Do nothing for fifth tour; keep the fittest tour intact;
END FOR mutations and tournament winners
  Population  $\leftarrow$  update old Population with new Population mutations and winners
END FOR total budget exhausted
RETURN overall best tour in population (time, route) and graph route;

```

ALGORITHM 1: Evolutionary Algorithm (EA1).

experiment, 10 runs were conducted in order to average the results. Results obtained were based on delta from optimal, whereby delta is defined as the difference between delivery-

time (and final route) obtained from the metaheuristic versus the delivery-time (and final route) obtained by an optimal guarantee process (i.e., MIP).

TABLE 1: Comparison of EA1 and EA2 to optimal solutions.

Sire	kg	From optimal				Comp. performance time		
		EA-1 Max	#opt	Avg.	EA-2 Max	#opt	EA-1 Elapse time (s)	EA-2 Elapse time (s)
10 nodes	0.00	0.00	10/10	0.00	0.00	111,10	23	25
15 node.	0.03	0.03	9/10	0.04	0.04	9/10	32	41

Note: algorithm performances are compared to known optimal solutions (brute force, 4EP). Results: averaged over ten MICHMelat. Standard drone parameters: Delivery nodes sampled from random uniform distribution, drone speed factor $\alpha = 2$, drone range $\kappa = 10$, number drones assigned truck $\nu = 4$.

TABLE 2: Lean system performance results. System performance Comparing geometric-based optimal drone parameters to standard drone parameters

Exp.	Size operating area (km ²)	Standard parameters ($\alpha = 2, \kappa = 10, \nu = 3$)			Optimal system parameters ($\alpha, \kappa, \text{ and } \nu$)		
		Avg. utiliL	Avg. wait	Avg. time	Avg. utiliL	Avg. wait	Avg. time
1	1	1.00	0.063	2.48	0.98	0.05	1.08
2	25	1.00	0.286	11.82	0.99	0.24	7.91
3	100	0.95	0.93	25.03	0.86	1.76	21.13
4	300	0.75	2.03	51.52	0.64	2.10	47.10
5	700	0.22	2.73	102.94	0.29	3.30	93.24
6	1000	0.24	2.22	139.47	0.29	3.16	103.31

For each of the six experiments, ten runs were conducted for each delivery area (km²) and results were averaged. Each experiment was comprised of 30 delivery nodes. Arbitrary system parameters ($\alpha = 2, \kappa = 10, \nu = 3$) were compared to lean system parameters determined by nonlinear-optimization equations found in Section 2.

$$\Delta = \frac{\text{objective value heuristic} - \text{optimal objective value}}{\text{optimal objective value}}. \quad (37)$$

The metaheuristic experiments were run on a Windows 10 operating system, Intel® Core™ i5-8350U CPU @ 1.70 GHz and 4 GB of memory. The main purpose of the experiments was to determine the accuracy of the metaheuristics denoted by delta (Δ or deviation from optimal) and total computer-solution time (*elapse time*). Since optimization can only be guaranteed (compared to MIP) by smaller job sizes, the job size for experiments ranged from 10 to 15 jobs (nodes). Table (Table 1) below shows results for 10 randomly generated instances with 10 and 15 nodes of each instance type. For each experiment (10 runs), the delta from optimal is averaged (avg.), the max deviation from optimal is determined (max), the total number of experiments found to be optimal (#opt out of 10), and the total computer-solution time is given (*elapse time*) for comparison.

Results indicate that both EA-1 and EA-2 are capable of solving the truck-multidrone problem near *optimal*. The EA-1 slightly outperformed EA-2 due to the higher utilization of drones. It is estimated that depending on the configuration of the system parameters to the operational spaced, EA-2 is efficient when there are more drones than necessary fitted to the truck or the system is highly saturated with resources.

7.2. Computational Study 2: Lean System Parameters Compared to Standard Parameters. The second study compares the performance of a lean or properly configured system parameters (α, κ, ν) against an arbitrarily set of system parameters ($\alpha = 2, \kappa = 10, \nu = 3$) often found in the literature. For the sake of simplicity, an experiment was drawn from a uniform random distribution of the (x, y) plane

comprised of Euclidean distance between each of the coordinate delivery nodes. Since the experiments are designed to compare the performance of a lean system to an arbitrary system, the performance parameters selected for comparison are delivery-time t_i , utilization $util.$, and truck or drone $wait$ time. Each experiment was comprised of ten (10) runs in order to obtain an averaged set of results. The delivery area was perturbed between 1 km² to 1000 km² to evaluate various delivery densities for 30 delivery sites. Table 2 shows the results of six experiments each with ten runs comparing the arbitrary system parameters to the lean system parameters derived from the nonlinear optimization in Section 3. The evolutionary algorithm EA-1 (Section 7) was used to optimally route and cluster the truck and n-drones throughout the delivery area to arrive at the optimal delivery times for each run for each experiment. EA-1 was also used to obtain average utilization (of drones) and the average wait time for either the truck or the drone for each operation within each run. The average of the ten runs is reported in Table 2.

Experiment results show that by adopting the lean system parameters, the overall delivery time is consistently reduced. Results show delivery time reductions for each experiment (56%, 33%, 15%, 8%, 9% and 26%). It is noted that for at least two experiments, the arbitrary configuration was close to the calculated lean configuration; thus, no major improvements were expected for experiments 4 and 5. An expected tradeoff is utilization. In poorly configured systems, the drones are either highly utilized or rarely utilized; thus, they range from bottleneck to saturated resources. The lean configuration slightly relieves the bottleneck system and unsaturates low utilization configurations. Average wait time for each operation slightly increased due to the stochastic nature of within the experiment design. In this case, the drones often found themselves advance of the truck in

operations and therefore at an increased opportunity to accumulate wait time.

8. Conclusions

The current literature lacks any usable information in terms of the highest yield regions of the design space (speed, range, and number drones) for a main tool with multiple assisting tools; specifically, herein, we analysed a truck-multiple-drone configuration. Few, if any, research articles address a tractable, usable MIP for the truck-multidrone problem for quick transfer and testing within standard optimization tools (IBM® ILOG CPLEX, Lindo® Lingo). Therefore, any business wishing to evaluate the use of a single truck fitted with multiple drones has a way neither to guide the selection of drone parameters nor to evaluate the efficiencies of those selections without developing scientific experiments. The “practical best case” geometric approach shown herein inverts the problem space and solves for practical best case drone speed, drone range, and number of drones given a general delivery scenario. Concretely, it serves as a basis to address practical design decisions regarding the proper configurations to achieve the maximum desired yields for a system. It serves as a way to screen out low performing designs while giving the decision maker a Pareto front of potential configuration solutions that achieve a lean alignment. Moreover, both the MIP and metaheuristics developed herein for empirical studies accurately solve for the optimal truck-multidrone routing, thus giving the business ample tools for evaluating generalized scenarios.

The simplified version of the MIP is not found in any other study; it is useful and an easily implemented metaheuristic necessary to solve for the optimal route and optimal time for the truck-multiple-drone for smaller problem sets. For larger problem sets, the single chromosome evolutionary algorithm (EA-1) is best in class metaheuristic to test various test-case scenarios. Both EA-1 and EA-2 were modelled as functions within the MATLAB® development environment language, and the files were made available at Mathworks® file exchange (dvtsp_ga_basic) for evaluation and general purpose use/testing. As far as we could surmise, our EA is the only algorithm available for such problems found in the literature or in an open resource environment.

In conclusion, this research answers the questions of expected efficiencies in time that would be expected given a truck-multidrone configuration as well as finding “what is the proper configuration for a truck-multidrone situation.” It gives business a foundation to evaluate a variety of configurations against a typical daily last-mile parcel-delivery scenario. The work also opens several additional questions for future research. These questions tend to be toward the total energy savings and the energy saturation for different truck-multiple-drone generalized situations the business may encounter. Future research studies a fleet of trucks each fitted with the optimal number and type of drones.

Data Availability

Genetic algorithms and metaheuristics are available online in MATLAB after publication.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

(1) R. R. conceived the presented idea, developed the system theory, performed the computations, developed the theoretical formalism, performed the analytic calculations, performed the numerical simulations, verified the analytical methods of the system theory, mixed integer program and metaheuristics, wrote all sections of the paper including all the tables and graphs, conducted the fundamental experiments, conducted the performance comparison for evolutionary algorithms 1 and 2 against optimal solution, and conducted the lean comparison tests to the arbitrarily configured system.

References

- [1] N. Agatz, P. Bouman, and M. Schmidt, “Optimization approaches for the traveling salesman problem with drone,” *Transportation Science*, vol. 52, no. 4, pp. 965–981, 2018.
- [2] B. Golden, S. Raghavan, and E. Wasil, “The vehicle routing problem: latest advances and new challenges,” in *Operations Research/Computer Sciences Interfaces*, vol. 43, Springer, Berlin, Germany, 2008.
- [3] P. Toth and D. Vigo, *The Vehicle Routing Problem SIAM Monographs on Discrete Mathematics and Applications*, SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, PA, USA, 2002.
- [4] S. N. Kumar and R. Panneerselvam, “A survey on the vehicle routing problem and its variants,” *Intelligent Information Management*, vol. 4, no. 3, pp. 66–74, 2012.
- [5] P. Toth and D. Vigo, *Vehicle Routing: Problems, Methods, and Applications*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2nd edition, 2014.
- [6] C. E. Miller, A. W. Tucker, and R. A. Zemlin, “Integer programming formulation of traveling salesman problems,” *Journal of the ACM (JACM)*, vol. 7, no. 4, pp. 326–329, 1960.
- [7] G. Dantzi and J. Ramser, “The truck dispatching problem,” *Management Science*, vol. 6, no. 1, pp. 80–91, 1959.
- [8] G. Clarke and J. W. Wright, “Scheduling of vehicles from a central depot to a number of delivery points,” *Operations Research*, vol. 12, no. 4, pp. 568–581, 1964.
- [9] M. Desrochers, J. K. Lenstra, and M. W. P. Savelsbergh, “A classification scheme for vehicle routing and scheduling problems,” *European Journal of Operational Research*, vol. 46, no. 3, pp. 322–332, 1990.
- [10] M. Drexel, “Synchronization in vehicle routing-A survey of VRPs with multiple synchronization constraints,” *Transportation Science*, vol. 46, no. 3, pp. 297–316, 2012.
- [11] A.-S. Pepin, G. Desaulniers, A. Hertz, and D. Huisman, “A comparison of five heuristics for the multiple depot vehicle scheduling problem,” *Journal of Scheduling*, vol. 12, no. 1, pp. 17–30, 2009.
- [12] Y. Lin, Z. Bian, S. Sun, and T. Xu, “A two stage simulated annealing algorithm for the many-to-many milk run routing problem with pipeline inventory cost,” *Mathematical Problems in Engineering*, vol. 2015, Article ID 428925, 22 pages, 2015.

- [13] C. C. Murray and A. G. Chu, "The flying sidekick traveling salesman problem: optimization of drone-assisted parcel delivery," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 86–109, 2015.
- [14] P. Bouman, N. Agatz, and M. Schmidt, "Dynamic programming approaches for the traveling salesman problem with drone," September 2017, <https://ssrn.com/abstract=3035323>.
- [15] S. Ferrandez, T. Harbison, T. Weber, R. Sturges, and R. Rich, "Optimization of a truck-drone in tandem delivery network using K -means and genetic algorithm," *Journal of Industrial Engineering and Management*, vol. 9, no. 2, pp. 379–383, 2016.
- [16] R. Rich, "Inverting the truck-drone network problem to find best case configuration," *Advances in Operations Research*, vol. 2020, Article ID 4053983, 10 pages, 2020.