Engineering Applications of Artificial Intelligence 138 (2024) 109337

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

A novel solution for routing a swarm of drones operated on a mobile host

Halil Savuran [a], Murat Karakaya [b,*]

[a] Independent Researcher, Izmir, Turkiye
[b] TED University, Software Engineering, Cankaya, Ankara, 06420, Turkiye

ARTICLE INFO

Keywords:
Endurance constraint
Mobile host station
Synchronized multiple vehicle routing problem
Swarm intelligence
Genetic algorithm
Evolutionary computation

ABSTRACT

The increasing use of drones across various sectors demands optimized deployment strategies under diverse constraints. This paper tackles the Multiple Capacitated Mobile Depot Vehicles Routing Problem (mCMoD-VRP), a challenging variant of the Vehicle Routing Problem (VRP) where multiple drones with limited flight range operate from a mobile depot. The goal is to maximize target coverage while considering flight endurance, depot mobility, and drone multiplicity. We introduce a novel evolutionary algorithm, Evolutionary Optimization for Synchronized Routing Problem (EOSRP), which constructs synchronized routes for the drone swarm, accounting for all constraints. EOSRP distinguishes itself with specialized genetic operators, specifically designed to efficiently handle the constraints of mCMoD-VRP, enhancing both exploration and exploitation of the search space. EOSRP also facilitates collaborative planning among drones, enabling them to share targets and optimize routes collectively, resulting in more efficient use of flight range capacity. Comprehensive simulations on benchmark problems demonstrate that EOSRP consistently outperforms a serialized version of our previous single-drone algorithm, Genetic Algorithm for Capacitated Mobile Depot (GA-CMoD), achieving an average of 8.7% higher target coverage and 7.28% more efficient use of flight range capacity. EOSRP's ability to generate synchronized solutions through collaborative planning leads to significantly improved mission efficiency.

1. Introduction

The rising accessibility of drones, combined with their low-risk, cost-effective, and operationally flexible nature, presents a wealth of opportunities for their deployment in diverse fields. Applications range from military intelligence and surveillance to commercial cargo delivery, agricultural data collection, and disaster relief. This expansion is driven by advancements in supporting technologies like artificial intelligence and advanced sensors, control, and communication capabilities. However, realizing the full potential of drone operations requires addressing the inherent challenges of optimizing their usage under various constraints.

A fundamental challenge in many drone applications is the need to visit a set of geographically dispersed locations for data collection or material delivery. This task can be addressed by a single drone launched from a fixed depot, but scenarios involving multiple drones and rapid deployment often necessitate more complex planning. Furthermore, geographical constraints, such as deliveries in remote areas or archipelagos beyond the range of fixed depots, introduce additional complexities. In such cases, the planning must also involve determining the temporary deployment location for a mobile depot, which could be a carrier ship or a truck.

This paper focuses on a particularly challenging scenario where the depot itself is in motion, requiring dynamic task planning for the drones. This scenario, known as the Multiple Capacitated Mobile Depot Vehicles Routing Problem (mCMoD-VRP), presents a significant research gap in the field of drone routing. The mCMoD-VRP problem involves optimizing the routes of multiple drones with limited flight range, operating from a mobile depot, to maximize the number of targets visited. This problem is a generalization of the classic Vehicle Routing Problem (VRP), incorporating additional constraints related to drone capacity, depot mobility, and the need for synchronized routing.

The mCMoD-VRP problem presents several key challenges:

- Determining Takeoff and Landing Points: Drones must be launched and retrieved at appropriate points along the mobile depot's route, considering both the drone's range and the depot's trajectory.
- Assigning Targets to Drones: Each drone must be assigned a subset of targets that can be visited within its flight range. Assign subsets of targets per each available drone that can be visited within the range constraint,

* Corresponding author.
E-mail address: murat.karakaya@tedu.edu.tr (M. Karakaya).

https://doi.org/10.1016/j.engappai.2024.109337
Received 26 February 2024; Received in revised form 21 July 2024; Accepted 12 September 2024
Available online 18 September 2024

- Synchronizing Drone Routes: The routes of multiple drones must be coordinated to ensure efficient collaboration and avoid unnecessary overlaps.
- Returning to the Mobile Depot: Drones must be able to return to the current location of the mobile depot within their flight range, despite the depot's continuous movement.

To address these challenges, this paper proposes a novel evolutionary algorithm, Evolutionary Optimization for Synchronized Routing Problem (EOSRP). EOSRP is a significant departure from existing approaches by introducing a unique combination of features:

- Synchronized Routing: EOSRP constructs synchronized routes for the drone swarm, considering the dynamic movement of the mobile depot and the limited flight range of each drone.
- Collaborative Planning: EOSRP incorporates a collaborative planning mechanism that allows drones to share targets and optimize their routes collectively, leading to more efficient use of flight range capacity.
- Hybrid Genetic and Memetic Algorithms: EOSRP combines genetic algorithms with local search heuristics to effectively explore and exploit the search space, ensuring robust and efficient solutions.

We demonstrate the effectiveness of EOSRP through comprehensive simulations on benchmark problems, comparing its performance to a serialized version of our previous single-drone algorithm, GA-CMoD. The results show that EOSRP consistently outperforms the baseline algorithm, achieving significantly improved mission efficiency.

We present the proposed algorithm with the following structure: Section 2 provides an overview of related works. Section 3 defines the research problem. Section 4 describes the proposed algorithm and the benchmark algorithm. Section 5 provides the simulation results and finally Section 6 concludes the study.

2. Literature review

The research on drone routing problems, particularly those involving mobile depots and multiple drones, has grown significantly in recent years. A comprehensive overview of various Vehicle Routing Problem (VRP) variants, methods, and applications can be found in the book by Toth and Vigo (Dell Amico et al., 2020). This work provides a valuable foundation for understanding the complexities of VRP and its numerous extensions.

2.1. Drone delivery from trucks: A focus on constraints, objectives, and algorithms

A significant portion of the literature focuses on two-echelon routing problems, where a ground vehicle (truck) acts as a mobile depot for drones. This "truck-and-drone" tandem model has gained popularity due to its potential for efficient last-mile delivery and its ability to overcome limitations of traditional delivery methods.

Several studies have investigated specific variants of the truck-and-drone tandem model, exploring various constraints, objectives, and algorithms.

- Constraints: Common constraints in these models include:
  - Drone Flight Range: Drones have limited flight range, restricting the distance they can travel from the truck.
  - Drone Payload Capacity: Drones have limited payload capacity, affecting the number and weight of packages they can carry.
  - Truck Route: The truck's route is often predetermined, influencing the launch and retrieval points for the drones.
  - Time Windows: Customers may have time windows for delivery, adding complexity to the scheduling of drone operations.
- Objectives: Typical objectives in truck-and-drone routing problems include:
  - Minimizing Total Travel Time: Reducing the overall time required to complete deliveries.
  - Minimizing Total Cost: Optimizing the balance between truck and drone operations to minimize costs.
  - Maximizing Target Coverage: Ensuring that as many delivery points as possible are reached within the constraints.
- Algorithms: A variety of algorithms have been proposed to solve truck-and-drone routing problems, including:
  - Genetic Algorithms: These algorithms use evolutionary principles to search for optimal solutions.
  - Heuristics: These algorithms provide approximate solutions that are often computationally efficient.
  - Dynamic Programming: This approach breaks down the problem into smaller subproblems that are solved iteratively.

2.2. Challenges of continuous depot movement

This paper focuses on a particularly challenging scenario where the depot itself is in motion, requiring dynamic task planning for the drones. This scenario, known as the Multiple Capacitated Mobile Depot Vehicles Routing Problem (mCMoD-VRP), presents a significant research gap in the field of drone routing. The mCMoD-VRP problem involves optimizing the routes of multiple drones with limited flight range, operating from a mobile depot, to maximize the number of targets visited. This problem is a generalization of the classic Vehicle Routing Problem (VRP), incorporating additional constraints related to drone capacity, depot mobility, and the need for synchronized routing. In essence, mCMoD-VRP extends the VRP by considering the dynamic movement of the depot and the unique characteristics of drone operations, such as limited flight range and the need for coordinated routing.

The continuous movement of the depot introduces several unique challenges such as:

- Dynamic Takeoff and Landing Points: Drones must be launched and retrieved at points that are constantly changing along the depot's route. This requires real-time adjustments to the drone's trajectory and flight plan.
- Synchronization with Depot Movement: The drone's routes must be synchronized with the depot's movement to ensure that the drone can rejoin the depot within its flight range. This adds complexity to the routing and scheduling process.
- Increased Computational Complexity: The dynamic nature of the problem requires more sophisticated algorithms to handle the continuous changes in the depot's location and the drone's flight paths.

2.3. Special properties of drone-only target visits

In contrast to scenarios where both the host station and drones can visit targets, this paper considers a situation where only drones visit targets. This scenario introduces specific properties:

- Increased Drone Responsibility: Drones become solely responsible for visiting all targets, requiring careful planning to ensure that each target is covered within the drone's flight range.
- Optimized Drone Utilization: The focus on drone-only target visits allows for more efficient utilization of drone resources, as the host station does not need to be involved in direct target visits.

- Simplified Host Station Operations: The host station's role is simplified, focusing primarily on providing ground services for the drones, such as reloading, maintenance, and regeneration.

2.4. Relevant studies

Numerous studies have investigated specific variants of the truck-and-drone tandem model. Murray and Chu (2015) introduce a shortest makespan problem where a truck and drone serve customers, with the truck launching and retrieving the drone at customer nodes. Ha et al. (2018) challenge the algorithm of Murray and Chu on the same problem model, aiming to minimize total operational costs. de Freitas and Penna (2020) consider the same problem with a different solution approach. Wang et al. (2020) investigate a similar variant where the drone can visit multiple customers. Baik and Valenzuela (2021) implement this problem type for optimal inspection of wind farms, minimizing total mission time. Yurek and Ozmutlu (2018) and Dell Amico et al. (2021) tackle the same problem with the objective of mission completion time.

Murray and Raj (2020) extend this problem type by introducing multiple serially operating drones with heterogeneous capacities. Moshref-Javadi et al. (2020) consider multiple capacitated drones that can be launched from the truck multiple times at customer locations, with the truck collecting them only at their dispatch points. Karak and Abdelghany (2019), consider a similar problem with different constraints, where drones can return to any predefined rendezvous locations. Dell Amico et al. (2020) consider a parallel routing model where a truck and a set of homogeneous drones depart from the depot to complete a delivery task. Luo et al. (2017) investigate a model where the ground vehicle travels on a road network and a drone it hosts visits off-road targets. Peng et al. (2019) study a similar model with a swarm of drones hosted by a truck for parcel delivery.

Sacramento et al. (2019) tackle a variant where capacitated trucks equipped with drones service customers with the cost minimization objective. Das et al. (2020) consider multiple trucks each equipped with a drone cooperating in a delivery mission, minimizing travel cost and maximizing time window matching. Poikonen et al. (2017) examine a model where the goal is to complete delivery with minimum time using multiple trucks each hosting multiple drones. Schermer et al. (2019b,a) study variants where drones can interact with the truck at customer nodes and discrete points along the truck's route. Poikonen and Golden (2019) consider a mothership and drone cooperation scenario where the drone can visit multiple targets. Kitjacharoenchai et al. (2019, 2020) consider scenarios with multiple trucks and multiple drones, where drones can use any truck as a host. Wang and Sheu (2019) consider a routing problem of drones cooperating with a network of trucks for parcel delivery.

Nguyen et al. (2021) and Poikonen and Golden (2020) consider scenarios where trucks and drones, both dispatched from a central depot, cooperate in parcel delivery, optimizing transportation cost and minimizing mission completion time. Jeong et al. (2019) extend this model by including energy consumption based on parcel weight and no-fly zone constraints. Luo et al. (2021) examine a cooperative scenario of a truck and multiple drones where drones can visit multiple targets and the truck can launch and recollect multiple drones at a customer location. Chiang et al. (2019) consider multiple trucks and drones in a tandem delivery operation, minimizing total carbon emission.

Agatz, Bouman, and Schmidt in Agatz et al. (2018) and Bouman et al. (2018) compare heuristic-based truck and drone solutions with truck-only solutions and propose a dynamic programming approach for TSP with drone problems. Wang et al. (2019) study a hybrid model where trucks, truck-carried drones, and independent drones collaborate for delivery, optimizing time and distance. bin Othman et al. (2017) consider a single drone with one customer delivery capacity that can meet the truck at rendezvous points. Liu et al. (2019) investigate a scenario where the drone can visit multiple targets and meet the ground vehicle at a different location. Chang and Lee (2018) consider a model where customers are bundled in groups and a truck parks at cluster centers, launching multiple drones for delivery. Carlsson and Song (2018) consider a model where the truck moves along a route to enable the drone it hosts to visit one customer at a time. Marinelli et al. (2018) consider a scenario where the drone can be launched and retrieved by the truck en route. Chen et al. (2019) consider a scenario where a ground vehicle hosts a swarm of two air vehicles that visit one target at a time. Our previous work (Savuran and Karakaya, 2015b) introduced the Capacitated, Mobile Depot - Vehicle Routing Problem (CMoD-VRP), where a moving host station hosted a single drone operating on a set of dispersed targets. We proposed a novel genetic algorithm, GA-CMoD, which is adapted for drone multiplicity and used as a comparison algorithm in this study.

The problem structure presented in this paper differs from the studies discussed above in one or more of the following factors: drone multiplicity, n-tuple visiting capacity per flight of a drone, liberty of the drone and the host station meeting at random points along the route, and the problem objective being the visit of the maximum number of targets possible under capacity constraint. These factors and our contribution to the literature are discussed in detail in Section 3.

3. Problem definition

As discussed in Section 2, there are various classification approaches based on decision problems such as depot location, capacity planning, job scheduling, coupling and fleet sizing of trucks or drones. The problem structured in this paper can be classified by the individual functions of drone and host station assets. In most of the studies, host station and drones both are capable of visiting targets, while the host station can also act as the mobile platform for the drones, whereas drones are employed as extending apparatus to reach out to the targets for feasibility or practicality prospects. In our model, only the drones visit targets and the host station has no such responsibility whose function solely is to provide ground services for the drones such as reloading, maintenance and regeneration while on the move. Another point is, due to the consideration of scenarios where small drones are employed with parcel delivery, most literature assumes drones in three-tuple operations (visiting one customer at a flight). Our model considers drones that have the characteristics of endurance and regeneration time rather akin to that of advanced UAVs, which enables them to visit multiple targets at one flight, and in return limits its availability to only one flight during a mission. These distinct decision parameters define the contribution of our problem model to the literature.

In this paper, we consider an advanced variant of CMoD-VRP (Savuran and Karakaya, 2015b): Multiple, Capacitated, Mobile Depot - Vehicles Routing Problem (mCMoD-VRP), which is a VRP constrained by multiplicity, flight range, and depot mobility constraints. mCMoD-VRP has the following problem properties:

- Host station is mobile on a given route (for simplicity, in the use case scenarios we assume that the route is fixed on a constant heading and at a constant speed),
- Multiple drones are available (the replenishment need of the drones is assumed to allow them to be tasked only once during a mission),
- Drones have finite flight range (drones are assumed to operate at a constant speed and the time spent for their vertical moves and loiter phase over targets is assumed to be negligible),
- Maximum number of targets must be visited.

In essence, the synthesis of these constraints relates mCMoD-VRP to three different combinatorial optimization problems defined in the literature: vehicle routing problem conjoined with job shop scheduling and knapsack problems.

Vehicle Routing Problem (VRP) is a generalization of the well-known Traveling Salesperson Problem (TSP) which was first defined

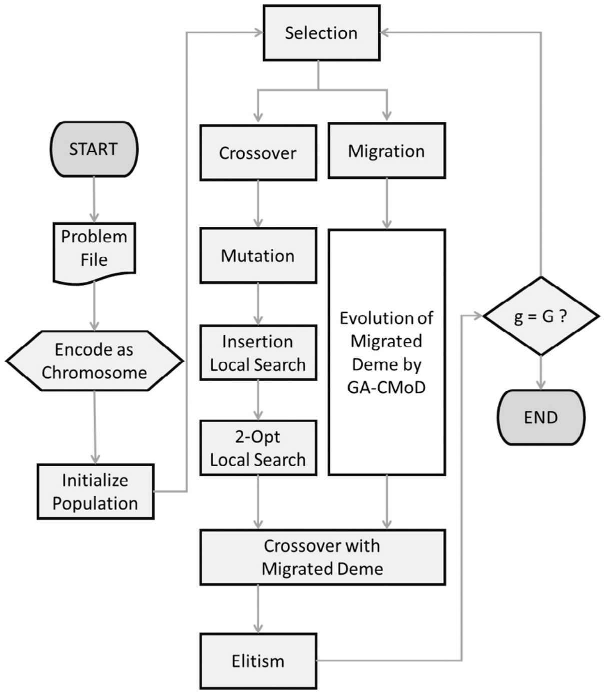

Fig. 1. Overview of EOSRP.

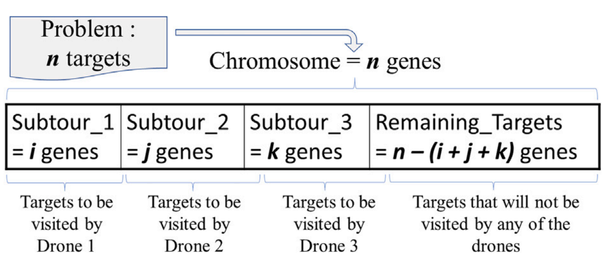

Fig. 2. Sample encoding of a three drone problem into a chromosome.

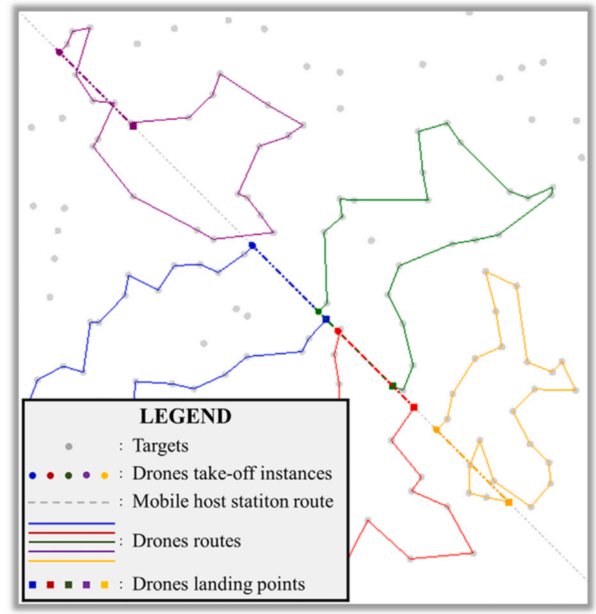

Fig. 3. A sample EOSRP solution.

formally by Karl Menger in 1930s (Menger, 1932) and aims to complete a Hamiltonian Circuit (Hamilton, 1856) for a set of vertices with the shortest route. The VRP was initially proposed by Dantzig and Ramser in 1959 (Dantzig and Ramser, 1959) addressing the routing optimization for a fleet of gasoline trucks, and attracted an enormous level of research interest ever since then owing to its facets in miscellaneous practical fields based on various constraints. Designing sequential visiting lists for drones of targets assigned to them in mCMoDVRP is therefore a VRP.

Job shop scheduling problem is known to be coined by Graham (1966) in 1966 and is another problem where the objective is to schedule n jobs on m processing elements in a way to minimize makespan. In the proposed solution, Evolutionary Optimization for Synchronized Routing Problem (EOSRP), processing elements are mapped to drones and jobs are mapped to the targets they are scheduled to visit, therefore determining the distribution of targets falls under this problem category.

Knapsack problem was initially studied by Dantzig (1931) in 1897 and models the decision making process in selection of a subset from a set of objects of variable weight and value, with the objective of maximizing the total value of selected objects while remaining within the total weight limit. Since drones have finite range and therefore must select the most optimal subset of weighted (in terms of travel cost) targets in EOSRP, knapsack problem is applied here. Even though virtually all targets have equal value, their relative position to other targets and the moving host station has direct effect on total value of the drone‘s tour.

4. Methodology

This section explains our methodology in handling the problem by describing the solution we propose and the competing algorithm we devised for the purpose of comparative evaluation.

4.1. The proposed approach: Evolutionary Optimization for Synchronized Routing Problem (EOSRP)

Evolutionary approaches are widely employed competent metaheuristic search methods that can effectively engage with NP-Hard problems (Fogel, 2000). In line with the evolving understanding of them as neatly noted by Sorensen et al. (2017), rather than strict algorithmic formulations we interpret metaheuristics as general frameworks where ideas, concepts and guidelines are offered for tailored combinations. Accordingly, in tackling Multiple Capacitated Mobile Depot Vehicles Routing Problem (mCMoD-VRP) that is described in Section 3, we propose an evolutionary approach based on a genetic algorithm (GA), combined with two local search heuristics (Nearest Neighbor and Insertion Local Search) and a collaborative hybridization strategy to effectively cover both exploration and exploitation dimensions.

The initial model of GA – which is often referred to as Canonical GA (Whitley, 1994; García-Martínez et al., 2018) – has been researched by Fraser, Bremermann and Holland in 1960s and 1970s (Fraser, 1957; Bremermann et al., 1962; Holland, 1975) mostly for the purpose of emulating the phenomenon of biological evolution. Not necessarily the purpose but the level of success achieved in designing an adaptive system and ideas put forth by this model inspired many future variants of GA that intended to cope with highly complex optimization problems.

In exploring a search space, GAs encode hyperplanes of a hyperspace into a string structure that is called a chromosome (*genotype*), where each variable in the string is represented as a gene within that chromosome. Then a collection of such strings form a population that undergo an iterative "selectrocombinative" process during which chromosomes are decomposed and reassembled by genetic operators. The objective function of the problem evaluates the fitness of chromosomes that determine their survivability at the end of each such cycle.

```
Algorithm Selection
    INPUT current population
    ratio_primary ← 0.75 (target coverage based percentage of population)
    ratio_secondary ← 0.25 (target coverage square based percentage of population)
    N ← population ceiling
    current population.orderBy(descending_total_target_coverage then
    ascending_total_tour_length)
    primary_selected_population ←current population.select from 0 to (N*ratio_primary)
    remaining population.orderBy(Descending.Square[total_target_coverage])
    secondary_selected_population ← remaining population.select from 0 to (N*ratio_secondary)
    selected population ← primary_selected_population + secondary_selected_population
    OUTPUT: selected population
```

Fig. 4. Algorithm of selection operator employed in EOSRP.

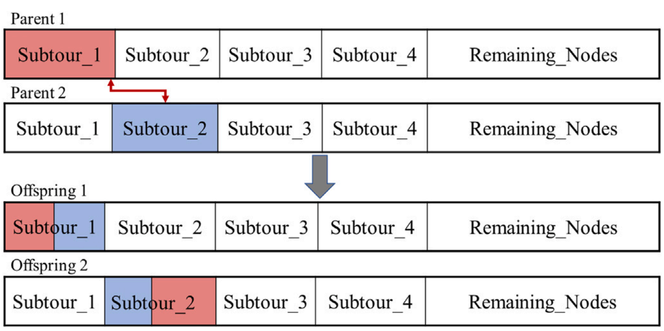

Fig. 5. Merge crossover.

Iterative application of this process through the pressure of cooperation and competition pushes better-fit combinations of genes in chromosomes up through generations. A common approach in augmenting GA is hybridization with local search methods, also called memetic approaches as coined by Moscato in 1989 (Mascato, 1989).

The method we developed for the mCMoD-VRP problem type is based on evolutionary genetic and memetic algorithms, that we name Evolutionary Optimization for Swarm Routing Problem (*EOSRP*). The general structure of the EOSRP algorithm is depicted in Fig. 1 and the details of the genetic, memetic, and repair operators employed are described in the following sections.

4.1.1. Encoding of a problem and initialization of a population

Genotype of a chromosome is constituted of element sets named *subtours* per the number of drones available for a mission and a stack of idle elements named *remaining targets*, as depicted for a sample three drones scenario in Fig. 2. Each subtour represents a target list to be visited by a drone within the range constraint, and remaining targets collection holds the targets that do not appear in any of the drones' visiting lists, due to not fitting within their range capacities as of given time. The sample encoding of a three-drone problem into a chromosome depicted in Fig. 2 demonstrates the EOSRP algorithm's ability to operate on drone fleets of varying sizes. Consequently, the examples throughout this paper will feature different fleet sizes, including three, four, and five drones.

A phenotype defined by such genotypes is sampled in Fig. 3, this time for a five-drone scenario. Such chromosomes are generated by randomly assigning targets to subtours while respecting the range constraint, up to the determined population parameter. This graphical notation in Fig. 3 will be used throughout the paper to illustrate all EOSRP outputs.

4.1.2. Selection

Selection in a GA is crucial not only in elevating the better-fit chromosomes but also in preserving the genetic diversity within the population. EOSRP implements fitness proportionate selection. The selection process is mainly based on the fitness of total target coverage of the swarm (by the summation of the target counts of all subtours first, and the summation of the tour lengths then). A smaller portion of the population (we set to %25 during our tests) is selected by fitness based on summation of square value of target counts of subtours. This approach is expected to give a second chance of survival for individually high performing subtours that reside in a chromosome whose overall fitness is depressed by the dominance of less optimal subtours and that may contain valuable genetic parts useful for future re-combinations. Refer to Fig. 4 the detailed algorithm.

4.1.3. Crossover

Crossover is an essential operator in GA that is expected to bring better traits of different parent chromosomes together to form offspring that may out-qualify both parents. The crossover operator in EOSRP randomly conducts a one-point merge operation between subtours of two chromosomes, as illustrated in Fig. 5. Crossover points from each

```
Algorithm Merge Crossover
    If (crossover probability)
        Parent_1, Parent_2 ← Select Random Chromosomes from Population()
        Crossover_Points_Pair ← find closest swap points (Parent_1, Parent_2)
        Offspring_1 ← Merge_Parts (Parent_1, Parent_2, Crossover_Points_Pair)
        Offspring_2 ← Merge_Parts (Parent_2, Parent_1, Crossover_Points_Pair)
        Offspring_1.Repair_operations()
        Offspring_2.Repair_operations()
        Population.Add(Offspring_1)
        Population.Add(Offspring_2)
    EndIf
```

Fig. 6. Algorithm of merge crossover operator employed in EOSRP.

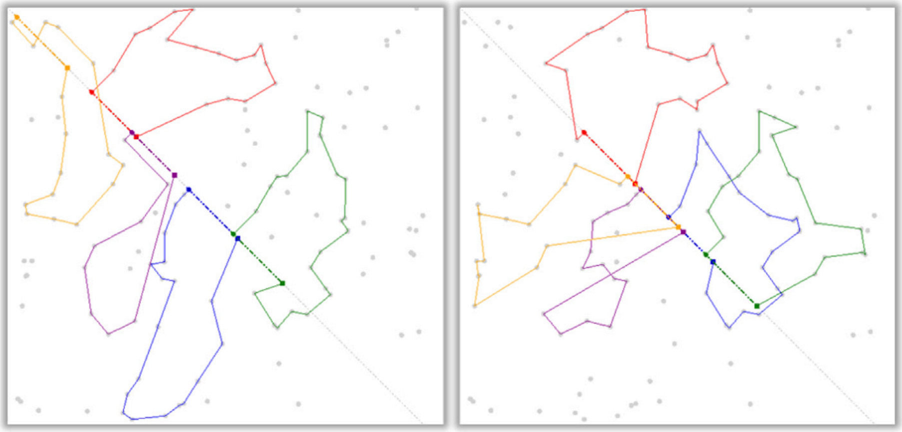

Fig. 7. Two chromosomes matched for crossover.

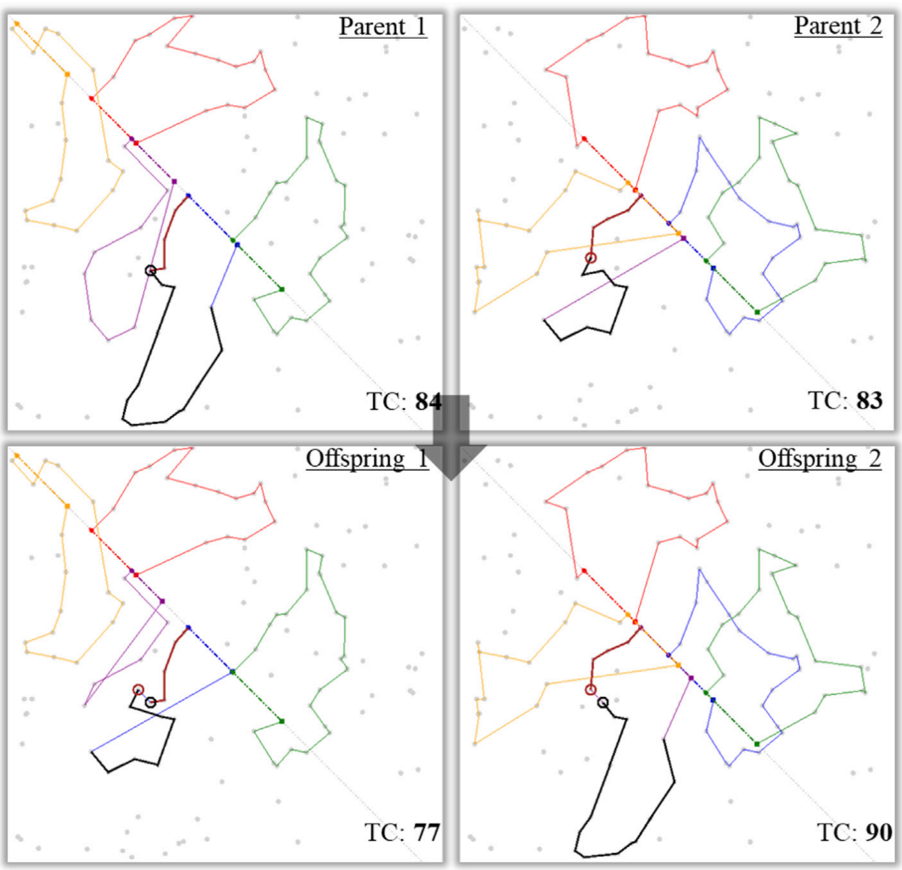

Fig. 8. Snapshot of a merge crossover in EOSRP.

parent chromosome are selected heuristically based on their geographical distance to one another, in a way to enable them crossover at where they come closest to each other. Depending on the crossover point this operator may either merge parts of different subtours or swap two subtours as a whole. Refer to Fig. 6 the detailed algorithm.

Fig. 7 represents a snapshot of two randomly matched chromosomes to undergo a crossover operation during a generation of a run of EOSRP on TSPLIB (Reinelt, 1991) ch130 problem for a five drones scenario.

As apparent in Fig. 8, in the result of this crossover one of the offsprings out-qualifies the both parents in total target coverage (TC).

4.1.4. Mutation

As mimicking the phenomena of genetic copying errors in biology, the purpose of the GA mutation is to induce minor deviations in chromosomes that through generations push random checks of the vicinity of the candidate solutions in the search space, which may otherwise get skipped as a consequence of lack of genetic diversity. As the nature of mCMoD-VRP suggests a rather greater number of potential combinations, in EOSRP we have implemented a rich set of mutation methods that include operators both we specifically designed for this particular problem type and operators that are popularly known

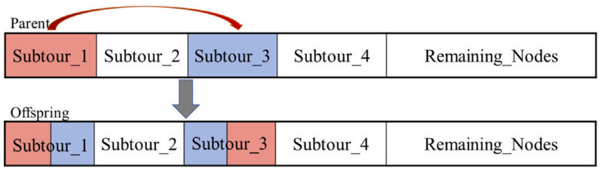

Fig. 9. Inbreed crossover mutation.

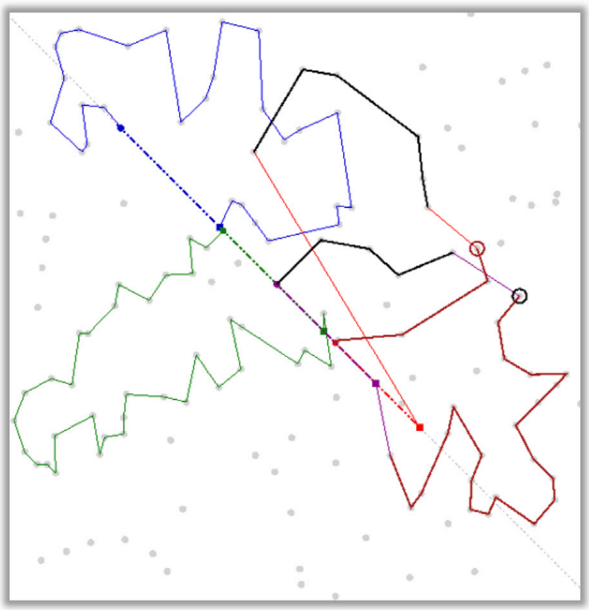

Fig. 10. Selection of inbreed crossover points and parts of subtours for exchange.

in the GA literature. In each mutation cycle of EOSRP, one of these operators are picked randomly by their predefined probabilities. Rather than increasing target coverage, mutation operators in EOSRP aim to yield transitional forms through itinerary rearrangements that can be exploited by future operations.

To effectively address the unique challenges of mCMoD-VRP, we introduce two novel mutation operators: inbreed crossover mutation and cost-aware swap mutation. In inbreed crossover mutation, from random sets of targets of two subtours of the same chromosome, a target couple that is geographically closest to each other are picked as crossover points where parts of subtours are exchanged at, as illustrated in the sample in Fig. 9.

Fig. 10 is a snapshot of inbreed crossover mutation operation captured during a run of EOSRP on TSPLIB ch130 problem for a four drones scenario and Fig. 11 is the end product of this operation, where the same number of targets are visited by the swarm with less tour length. Refer to Fig. 12 for the detailed algorithm.

The other specialized mutation operator, cost-aware swap mutation picks a random target from the receiving subtour of a chromosome and finds the least costly target in the sending subtour in relation to that target and transfers it next to the other. The least costly target refers to the target from sending subtour that causes least tour length increase in the receiving subtour when inserted. Fig. 13 is a sample of such operation taking place during a run of EOSRP on TSPLIB ch130 problem for a five drones scenario. Refer to Fig. 14 for the detailed algorithm.

Unlike traditional crossover operators that focus on exchanging fragments between different chromosomes, inbreed crossover mutation specifically targets the swap of the fragments of subtours of the same chromosome based on geographical convenience. This strategy helps retain the built-up fragments of the routes while seeking for further exploitation of them through internal recombinations.

The other mutation operators we implemented in GA-CMoD are displacement, inversion, insertion, and exchange mutations that are popular in the literature in application of GA on VRP-like problems. You can refer to Larranaga et al. (1999) for the implementation logic and details of these operators.

4.1.5. Migration

Migration may not be among the most commonly employed operators in GA, but the excessive size of search space in mCMoDVRP led us to exploit this method in enriching the genetic diversity within the population to better escape premature convergence. Studies of Cantu-Paz can be referred for migration policies and their effects in GA in detail (Cantú-Paz, 1998; Cantu-Paz and Goldberg, 2000; Cantú-Paz, 2001). We adopted a multi-culturality oriented migration policy (Araujo and Merelo, 2010). In EOSRP, based on migration frequency (f) (set as 100 generations in EOSRP), a migrant deme (set as 25% of the population) is migrated back and forth between a segregated evolution environment (ie. island) and the main evolution environment (ie. mainland) as follows: A reduced search space is defined by sampling of the remaining targets part of the most fit individual at generation f (see Fig. 15), and migrant deme is initiated in this search space. Migrant deme evolves for the next f generations for a single drone solution in its own search space through GA-CMoD. Refer to Fig. 16 for the detailed algorithm.

Then every other fth generation, two operations take place: first, the chromosomes in island deme undergo a crossover operation with chromosomes of the mainland population, as sampled in Fig. 17 (also observe the effect of repair operations on resulting tours, explained in Section 4.1.7). This crossover operation deliberately picks subtours in swarms that have the least target coverage for replacement. Second, a new migrant deme is generated from the remaining targets of the most fit chromosome of the current generation, repeating the process described above. This migration cycle is continued throughout the run of EOSRP.

4.1.6. Insertion & 2-Opt local search memetics

As genetic algorithms emulate biological evolution, memetic algorithms mimic cultural evolution (Mascato, 1989), which is intrinsically based on top of genetic foundation. Likewise, the purpose of memetic operators in a hybrid GA is to seek opportunities for fine tuning of solutions within their immediate vicinity. EOSRP employs two memetic operators on the exploitation dimension.

Among various variants of Insertion Local Search heuristic (Hoos and Stützle, 2004), we implemented Nearest Insertion Local Search in EOSRP. This operator randomly picks a target in one of the subtours of a chromosome, and finds a target to insert among a random set of remaining targets that causes the least increase in the tour length of that subtour when inserted at that locus. Fig. 18 represents a sample of insertion local search operation that takes place during a run of EOSRP on TSPLIB ch150 problem for a four drones scenario. In this example, the algorithm has selected a new unvisited target (dashed red circle) from the subtour assigned to the drone with the blue route. It then searches through a random subset of the visited targets and select one of them (red circle) such that the new unvisited target can be inserted with the least increase in the drone's total travel distance.

2-Opt is a popular local search method first proposed by Croes in 1958 (Croes, 1958) that rearranges the order of edges in a graph by means of exhaustive sequence reversals to reduce tour length. It effectively removes intersections however introduces a heavy computational load. Therefore in EOSRP we apply 2-Opt only on the subtours of the most fit chromosome of each generation, and let this memetic learning spread around in the population naturally through genetic interactions.

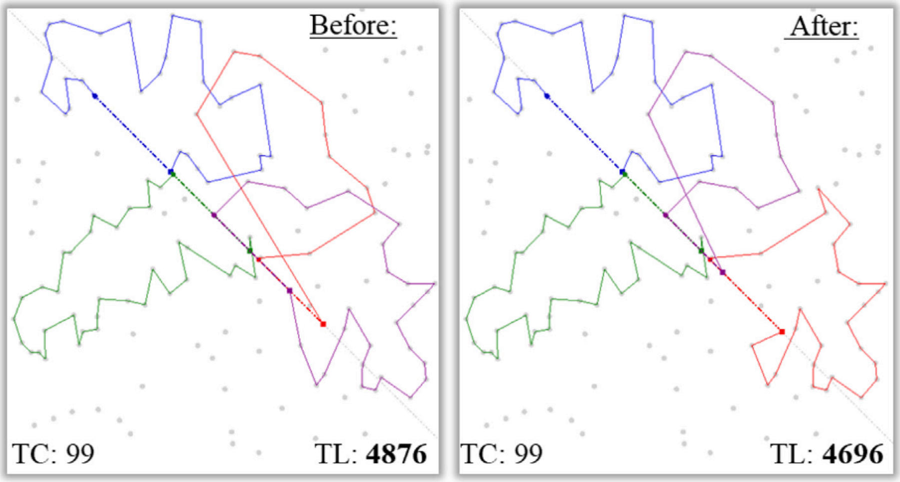

Fig. 11. Chromosome before and after undergoing inbreed crossover mutation.

Algorithm Inbreed Crossover Mutation

```
If (Inbreed Crossover mutation probability)
    Parent ← Select Random Chromosome from Population()
    SubTour_1, SubTour_2 ← Select Random Subtours from Parent()
    Crossover_Points_Pair ← find closest swap points (SubTour_1, SubTour_2)
    Offspring= Merge_Parts (SubTour_1, SubTour_2, Crossover_Points_Pair)
    Offspring.Repair_operations()
    Population.Add(Offspring)
EndIf
```

Fig. 12. Algorithm of in-breed crossover mutation operator employed in EOSRP.

4.1.7. Repair operators

A repair operator is *an algorithm that transfers the candidate solutions into the search space of the problem when they fall outside of it*, as described by Larranaga et al. (1999).

Since genetic algorithms work in a fashion of recombining building blocks of candidate solutions, while operating on a permutation problem they have to take measures to avoid repetition of building blocks in their delivered products (Whitley and Yoo, 1995). Besides this condition of non-repetition of targets within a swarm‘s subtours, in the case of mCMoD-VRP another condition is confining the tour lengths of drones within their range capacities.

Satisfaction of these conditions is ensured by means of repair operators which are typically appended after genetic operators. To ensure that the generated solutions are valid and adhere to the problem's constraints, we employ two specialized repair operators in EOSRP. The first one handles the cases where the same target appears twice within a swarm (as a result of being inherited from both parents after a crossover operation), by keeping the target in the itinerary of the drone where it adds less tour length compare to that of the other drone, and dropping the other instance. The second repair operator takes care of range violation. If the tour length of a drone exceeds its designated UAV range (as a result of a genetic operation), then among a random subset of its targets the operator drops the target(s) that adds the most tour length, iteratively until the constraint is satisfied.

4.1.8. Calculations of takeoff and landing points of drones

In EOSRP, the determination of takeoff and landing points of drones along the route of the mobile host station is handled as part of the optimization problem. Takeoff point is determined upon to take the shortest path to the first target of a subtour, and the landing point is predicted by calculating the meeting point of the drone and the carrier. Details of the geometrical calculations used in acquiring both of these points can be referred to at the Appendix A in our previous study (Savuran and Karakaya, 2015a).

4.2. A competing algorithm: Serialized GA-CMoD

In order to comparatively evaluate the performance of the proposed approach, we modified the GA-CMoD algorithm that we had proposed in an earlier work for single drone based mCMoD-VRP (Capacitated Mobile Depot Vehicle Routing Problem) to handle multiple drones case, as described below.

In the CMoD-VRP problem, there is only one drone available during a mission, and the objective of the GA-CMoD algorithm is to build a route to maximize the number of targets to be visited employing this single drone, as sampled in Fig. 19. Further details about the GA-CMoD algorithm and its performance can be found in Savuran and Karakaya (2015a).

For adapting the single-drone GA-CMoD algorithm for an mCMoD-VRP scenario where a swarm of drones are available for tasking in a mission, we modified its design simply by running the algorithm repeatedly on the search space per the number of allocated drones, by inputting the search space that is updated by deduction of the visited targets in every iteration. The modified version of GA-CMoD is given in Fig. 20. As this approach serializes single drone solutions to form up a swarm solution, we name this implementation of GA-CMoD as *Serial GA-CMoD*.

The serialization of single drone solutions generated by GA-CMoD through iterative reduction of search space and the end product delivered by this process is given in Fig. 21. This figure illustrates how Serial GA-CMoD works by sequentially finding the best route for each drone, considering the remaining targets after each iteration. The algorithm first finds the optimal route for the first drone (blue route), then for the second drone (red route), and finally for the third drone (green route). The final solution is then formed by combining these individual drone routes, as shown in the figure. Each drone's route is determined independently, without considering the potential for collaboration or synchronization with other drones.

5. Simulation results

In evaluating the competence of our proposed EOSRP algorithm, we observed its performance against the serialized GA-CMoD on multiple TSP benchmark problems for three, four, and five drones. This section explains the simulation environment, results of computational tests with providing quantitative metrics and some generated solution samples and summarizes the study with a discussion of our findings.

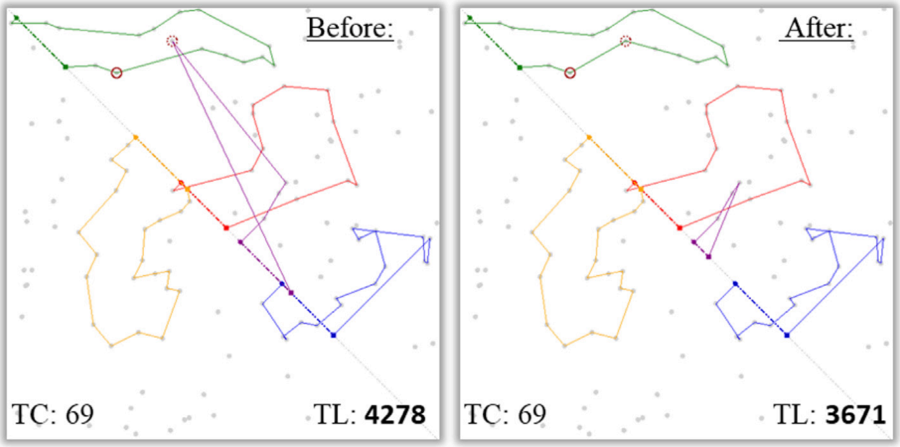

Fig. 13. Chromosome before and after undergoing cost-aware swap mutation.

```
Algorithm Cost-aware Swap Mutation
    If (Cost-aware Swap Mutation probability)
        Parent ← Select Random Chromosome from Population()
        SubTour_1, SubTour_2 ← Select Random Subtours from Parent()
        Swap_Points ← find cost-based swap points (SubTour_1, SubTour_2)
        SubTour_1 ← Subtour_1.Rearrange with insertion of SwapPoint_2 at SwapPoint_1.index()
        SubTour_2 ← SubTour_2.Rearrange with removal of SwapPoint_2()
    EndIf
```

Fig. 14. Algorithm of cost-aware swap mutation operator employed in EOSRP.

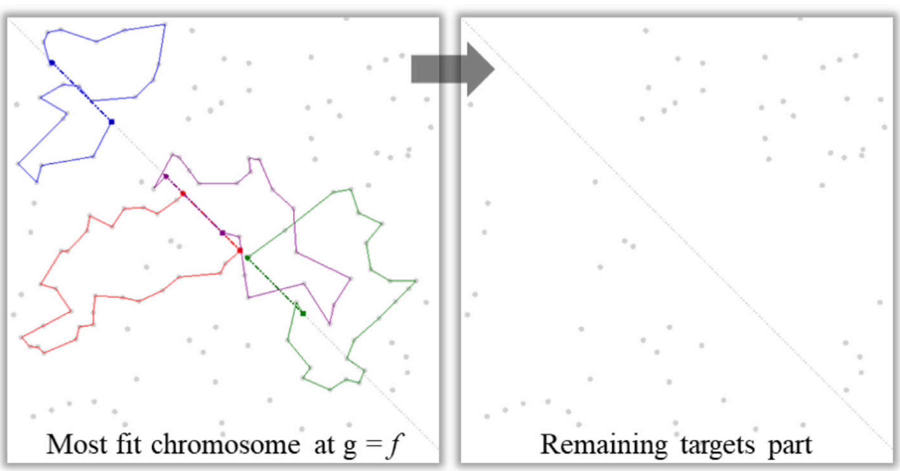

Fig. 15. Extraction of remaining targets part from the most fit individual for initiation of a migrant deme.

5.1. Simulation environment

We implemented and evaluated the proposed solution using the test bed we have designed and coded in C# language using MS Visual Studio IDE for experimental purposes specifically for this research.

Considering the diversity in size and topology as well as the computational capacity available for the tests, we selected 14 benchmark problems from TSPLIB as the target topology and each problem was tested for three, four, and five drones scenarios with range constraints per drone proportionate to the known optimal tour lengths (KOTL) of problems (Reinelt, 1991), as given in Table 1. Each such test scenario was run for 10 times.

Per our purpose of measuring the full solution convergence capability of tested algorithms (ie. not seeking faster convergence), we fixed the parameter sets that stabilize convergence at solution plateau for long enough as sampled in Fig. 22 and empirically tuned the parameters as given in Table 3 for EOSRP. The details of our parameter tuning methodology can be referred to at our preceding study (Savuran and Karakaya, 2015a).

For fairness, while scaling the generation (G) and population (N) parameters of EOSRP on serial GA-CMoD implementation, we have distributed the amounts equally per each drone (for example, berlin52 with 4 drones scenario was run with 600/4 = 150 generations and 800/4 = 200 populations for each drone in serial GA-CMoD).

```
Algorithm Migration
searchSpace ← All targets defined for the mission
population ← initialize population (searchSpace)
g←0, G←Number of generations to run, Freq ←Migration Frequency
While (g < G)
        If (g.Mod(Freq)=0)
            islandPopulation ← initialize population (mostFitIndividual.RemainingNodes)
        EndIf
        If (g > Freq)
            islandPopulation.Apply_genetic_ops()
        EndIf
        Population.Apply_genetic_ops()
        If ((g > Freq) AND ( g.Mod(Freq)=Freq-1))
            Do Migration Crossover between population and islandPopulation
        EndIf
        g←g+1
EndWhile
```

Fig. 16. Algorithm of migration operator employed in EOSRP.

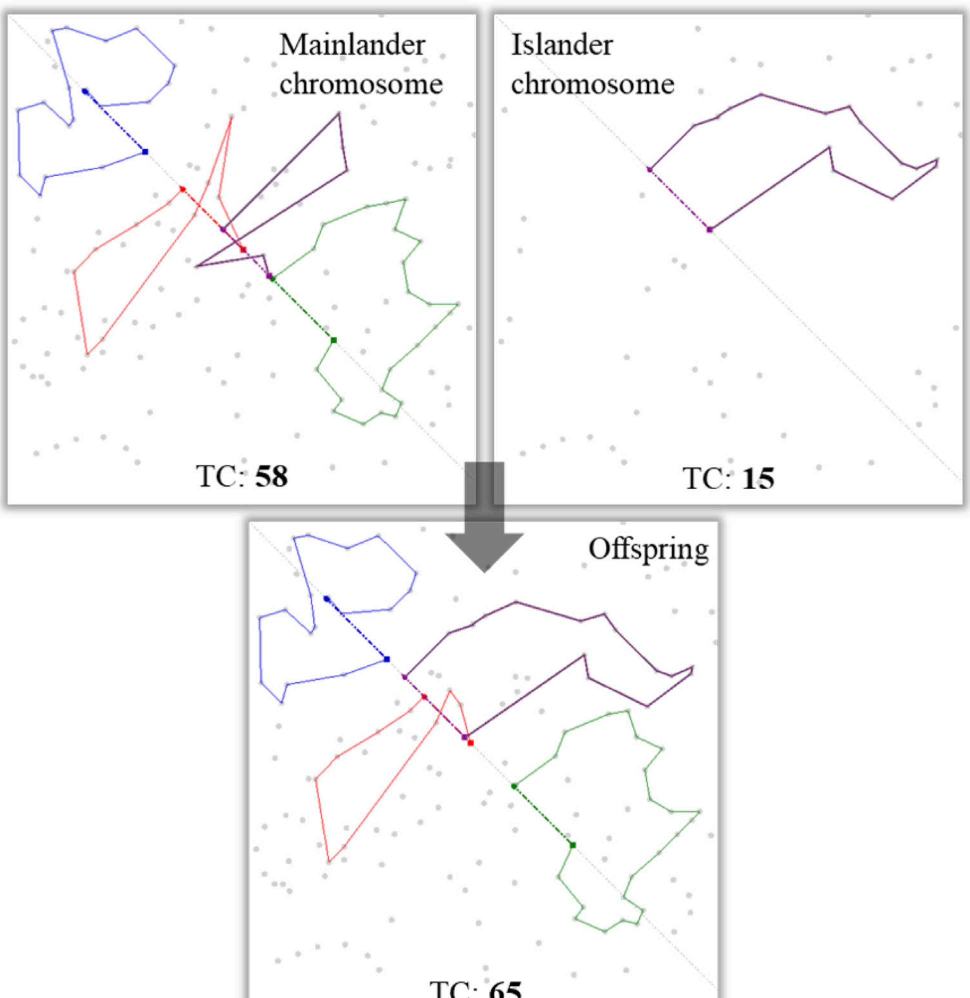

Fig. 17. Crossover with a migrated chromosome.

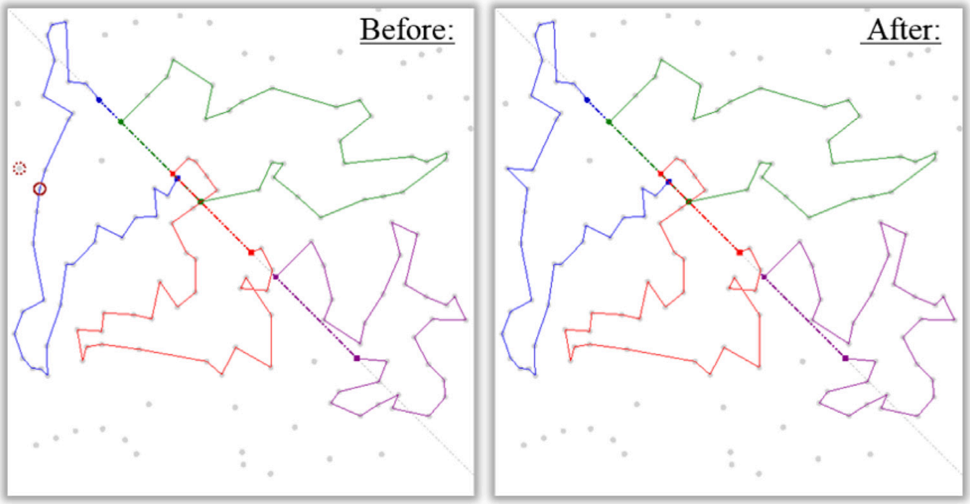

Fig. 18. Insertion local search operation.

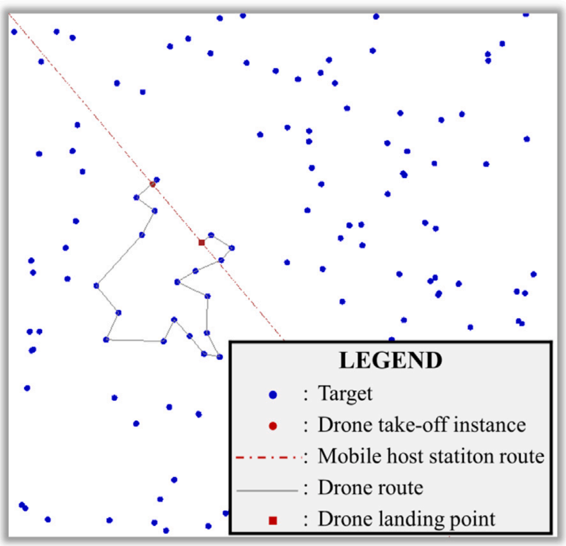

Fig. 19. A sample GA-CMoD solution.

Table 1
Tested benchmark problems and range constraints.

Problem Id.	KOTL	Range constraints (% of KOTL)		
		3 drones	4 drones	5 drones
berlin52	7542	25	20	15
eil76	538	25	20	15
pr76	108 159	25	20	15
kroB100	22 141	33	25	20
eil101	629	25	20	15
lin105	14 379	33	25	20
bier127	118 282	25	20	15
ch130	6110	25	20	15
pr144	58 537	33	25	20
ch150	6528	25	20	15
rat195	2323	25	20	15
kroB200	29 437	25	20	15
a280	2579	25	20	15
linHP318	41 345	25	20	15

For researchers interested in reproducing the tests, the environmental parameters used are given in Table 2.

Table 2
Environmental parameter values used in tests.

Parameter	Value
Host station start point (X,Y)	(0,0)
Host station heading (Degrees)	135
Host station speed (Unit)	40
Drone speed (Unit)	300

Table 3
Population (N), generation (G), crossover (P_c), mutation (P_m) and insertion local search (P_{ils}) probability parameters implemented for each benchmark problem in simulation tests.

Problem Id.	G	N	P_c	P_m	P_{ils}
berlin52	600	800	0.20	0.05	0.05
eil76					
pr76					
kroB100					
eil101					
lin105					
bier127		1200			
ch130	700	1600			
pr144					
ch150	1000				
rat195					
kroB200					
a280	1200	2000			
linHP318	1300				

5.2. Computational tests

In this section we discuss the performance of the EOSRP through two types computational tests. First we analyze the performance of individual genetic and memetic operators through ablation tests, then compare the overall performance of EOSRP against the competing serial GA-CMoD algorithm.

5.2.1. Individual contributions of operators

In order to test the individual contributions of the operators, we compared the performance of EOSRP against its variants from which the tested operator is removed from. We randomly selected the krob200 5 drones benchmark problem as the test case and conducted the tests with the same parameters given in Table 2. As given in Fig. 23 we observed that all tested operators contributed to the overall performance in various degrees. Especially the introduction of Inbreed Crossover

Algorithm Serialization of GA-CMoD on mCMoDVRP

```
dronesAvail ← number of drones allocated
searchSpace ← All targets defined for the mission
g←0, G←Number of generations to run,
singleSolution, swarmSolution ← nil
PROCEDURE GA-CMoD (dronesAvail, searchSpace)
        population ← initialize population (searchSpace)
        While (g < G)
            singleSolution     Population.Apply_genetic_ops()
            g←g+1
        EndWhile
swarmSolution.Add(singleSolution)
dronesAvail ← dronesAvail -1
searchSpace ← searchSpace – singleSolution.targets
If (dronesAvail>0)&(searchSpace > 0) then
        CALL (GA-CMoD(dronesAvail, searchSpace))
EndIf
END PROCEDURE
OUTPUT swarmSolution
```

Fig. 20. Algorithm of serialization of GA-CMoD.

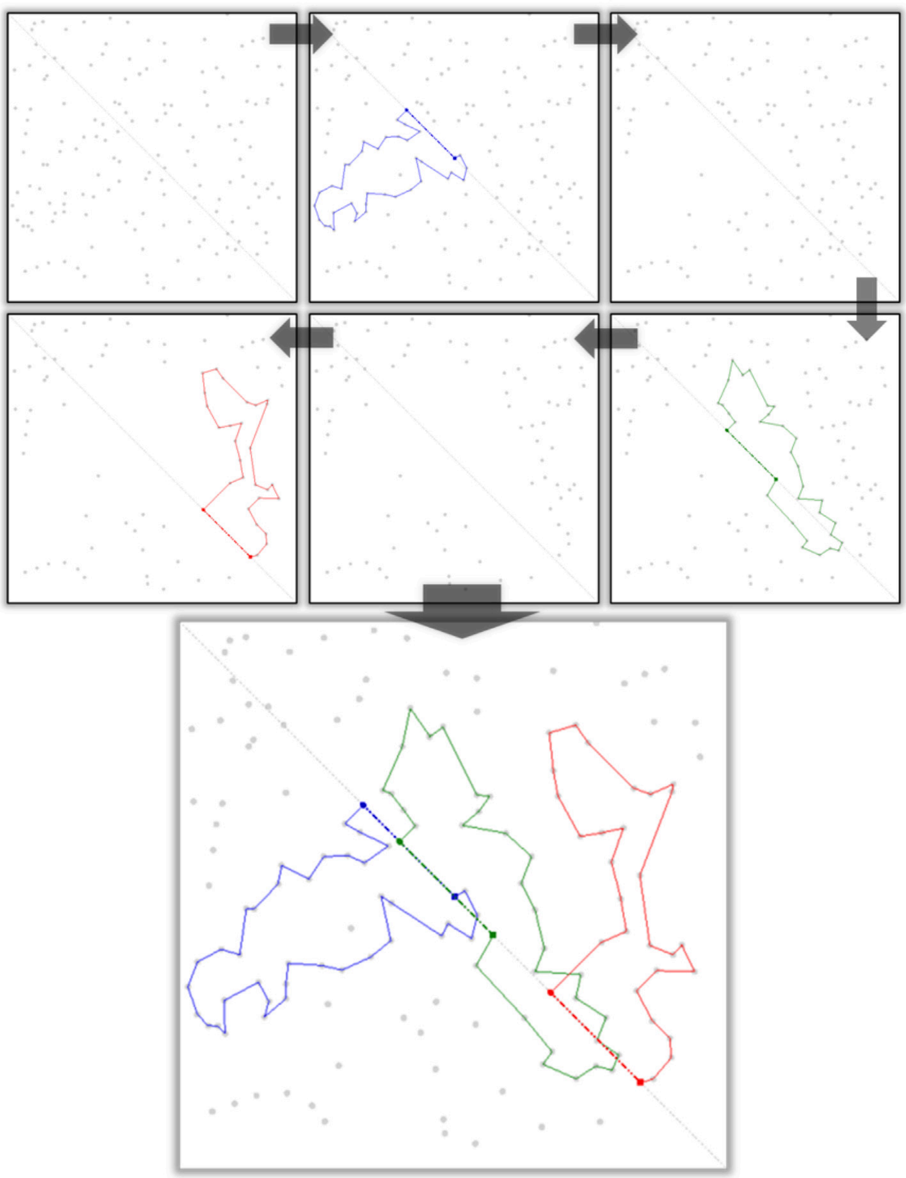

Fig. 21. Serial implementation of GA-CMoD in forming up a swarm solution.

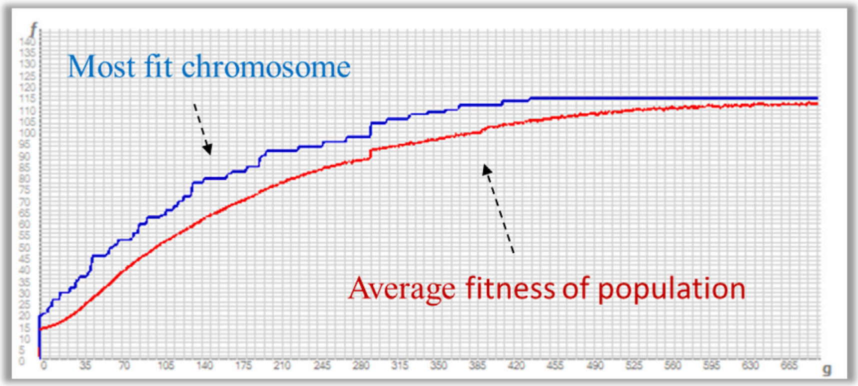

Fig. 22. Sample convergence of a solution for TSPLIB pr144. (f is Fitness measured as the total number of targets visited, and g is generation represented as the number of the algorithm iterations).

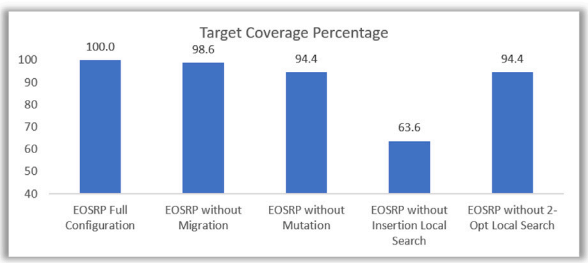

Fig. 23. Comparison target coverage of EOSRP variants with different operator configurations (Read: as the percentage of the target coverage performed by EOSRP in full configuration).

Mutation and Cost-Aware Swap Mutation significantly contributes to EOSRP's performance. These operators, by focusing on geographically close targets and cost-aware swaps, effectively address the unique challenges of mCMoD-VRP, leading to improved target coverage and flight range utilization.

5.2.2. Performance of EOSRP versus serial GA-CMoD

In comparing the performance of our proposed EOSRP algorithm against serial GA-CMoD on various TSP benchmark problems for three, four, and five drones scenarios, we employed the following performance metrics to compare the two algorithms:

- Target Coverage: This metric measures the total number of targets visited by the drone swarm during a mission. A higher target coverage indicates a more efficient solution, as it signifies that the drones can reach and service a greater number of locations within their flight range and time constraints.
- Flight Range Efficiency: This metric measures the ratio of the total distance flown by the drones to the total number of targets visited. A lower flight range efficiency indicates that the drones are using their flight range more effectively, minimizing unnecessary travel and maximizing the number of targets visited per unit of flight distance.
- Range Capacity Utilization: This metric measures the percentage of the allocated flight range that is used by the drones during a mission. A higher range capacity utilization indicates that the drones are making better use of their available flight range, minimizing wasted flight time and maximizing the overall efficiency of the drone deployment.

According to the test results, EOSRP outperformed serial GA-CMoD algorithm consistently without any exception, in all 42 test scenarios.

Fig. 24. Comparison of targets covered by EOSRP and Serial GA-CMoD.

On average, EOSRP covers 8.7% more targets than serial GA-CMoD. Fig. 24 represents the performance comparisons of both algorithms, where charts at the bottom demonstrate average values of 10 runs of each algorithm per each problem for three, four, and five drones scenarios individually, and the chart at the top is the comparison of the average of the averages of these three scenario types.

EOSRP achieves this through more efficient and effective use of flight range capacity allocated for drones. The solutions delivered by EOSRP are more capacity-efficient in terms of the ratio of capacity required per target:

$$Efficiency = \frac{Distance\ Flown}{Targets\ Visited}$$

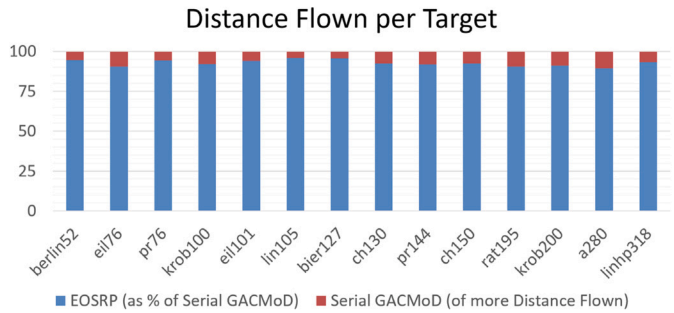

Fig. 25. Comparison of distances flown per target visited.

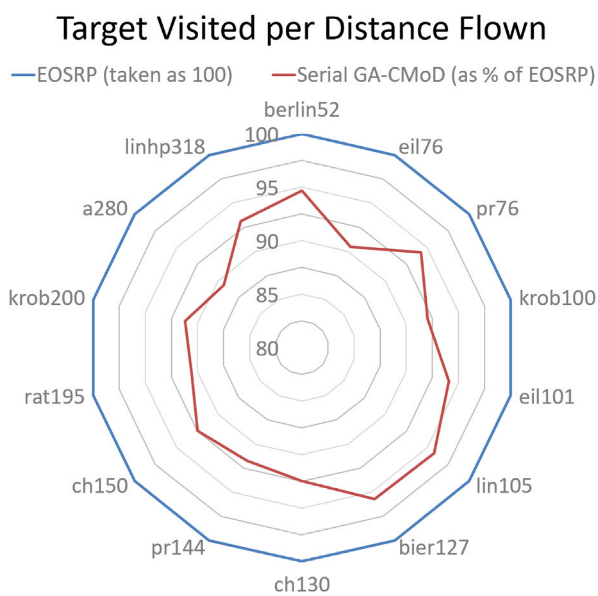

Fig. 26. Comparison of targets visited per distance flown.

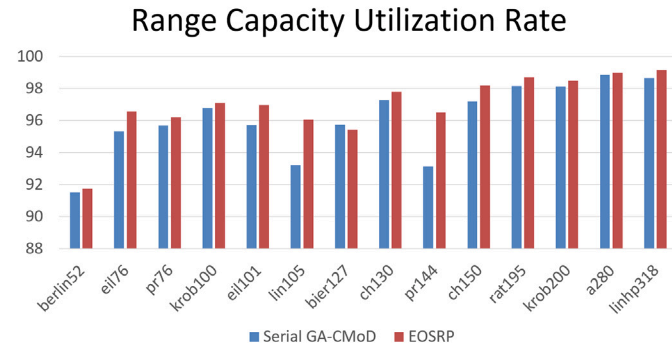

Fig. 27. Comparison of range capacity utilization rates.

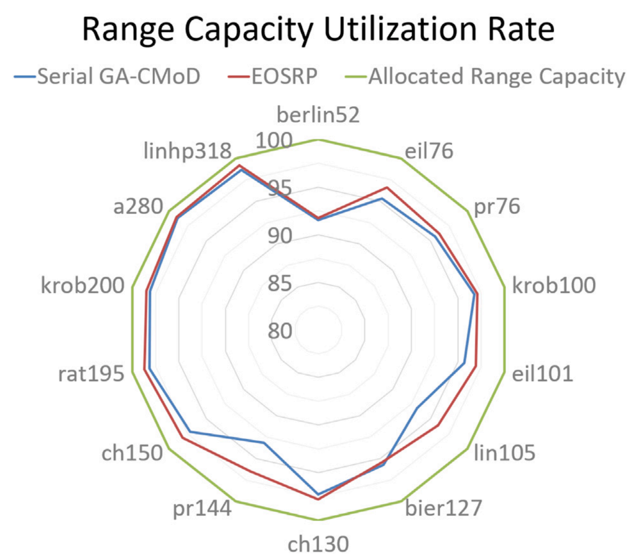

Fig. 28. Comparison of range capacity utilization rates.

On average, EOSRP consumes 7.28% less flight capacity than serial GA-CMoD per each target visited, as demonstrated in Figs. 25 and in 26 as averages of averages of 10 runs for each of the three scenario types.

EOSRP also is more effective as it better utilizes the flight range capacity:

$$Effectiveness = \frac{Distance\ Flown}{Flight\ Range\ Capacity} * 100$$

EOSRP is able to facilitate more effective use of the flight range as it utilizes 98.1% of the allocated ranges of the drones on average, against that of 96.9% of serial GA-CMoD. With the topology-specific exception for the bier127 problem, this is observable in Figs. 27 and 28, where utilized flight ranges of each problem file are included, as averages of averages of 10 runs for each of the three scenario types.

Serialized GA-CMoD has also demonstrated that it is fairly competitive in handling this problem type. However, contrary to EOSRP, it builds routes for individual drones discretely in isolation and lacks the awareness of capabilities of the other drones that are available for assignment during a mission, thus operating in an asynchronous planning mode. These distinctions are easily observable in the solutions both algorithms delivered during simulations, some of which are sampled from Figs. 29 to 33 where the best route plans out of 10 runs delivered by EOSRP and serial GA-CMoD on a set of problems are compared. One can clearly observe that how synchronous planning enables the swarm logic to make effective load sharing and resource utilization through smart trade-offs and hand-overs of targets across drones to create room for extending tours of the other drones onto prolific places around the landscape.

For instance, in Fig. 29, EOSRP is able to extend its target coverage to the west and south of the map that are not opted by serial GA-CMoD, whose unawareness of the swarm led it to cover 7.7% fewer targets by dispatching the first drone (green route) to the central cluster of 73 targets, second drone (red route) to the 65 eastern targets, which are more dense but at a longer distance, and third (blue route) drone to the better parts of the remaining targets at center and center-west, totaling 56. These tours have a high rate of overlapping with each other, which results in less efficient utilization of flight range capacities.

EOSRP generates a collaborative swarm plan thanks to its design by trading-off eastern targets at further distances with closer ones around the map, in conjunction with the hand-over of targets among drones that results in less overlapping of routes, leading to more efficient concentration. Consequently, EOSRP optes for dispatching the first drone (green route) to the center-east, second (red route) to the west and third (blue route) to collect the ignored targets by both of these at center-north and center-south, while elegantly linking these sectors by a line of *en passant* targets. This design results in a more balanced load sharing among drones by distributing 71, 68, and 71 targets respectively per each.

A similar distinction of design logic between the routes planned by the both algorithms is identifiable also in the other samples demonstrated in Figures from 30 to 33.

Furthermore, apart from the comparison of averages presented above, we observe that out of ten runs conducted for each test scenario,

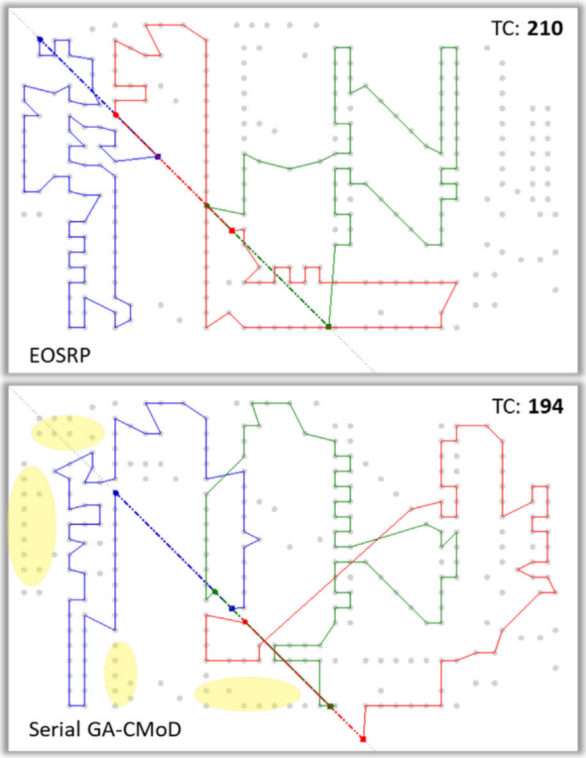

Fig. 29. Comparison of tours delivered by both algorithms for a280 three drones scenario.

93% of the times, even the worst results delivered by EOSRP have been better than the best results delivered by serial GA-CMoD. We consider this performance as an indication of solid superiority.

6. Conclusion

The Capacitated, Mobile Depot - Vehicle Routing Problem (CMoD-VRP) is a highly complex variant of VRP, with multiplicity, mobility, and capacity constraints. CMoD-VRP has rapidly growing practical reflections in various industrial fields. In this study, we proposed a novel Genetic Algorithm (GA) based solution for CMoD-VRP. Using GA approach, we proposed the Evolutionary Optimization for Synchronized Routing Protocol (EOSRP) which is a flexible and robust solution effectively explores and exploits the search space.

Since CMoD-VRP is a highly complex optimization problem with multiplicity, mobility, and capacity constraints, we develop the EOSRP solution by implementing novel approaches such as:

- Encoding of the problem into a chromosome structure in which the design of the building blocks structure donated them two-dimensional mobility both globally around the solution space and locally across the route designs of individual drones of a fleet, especially through crossover and migration operators. The effectiveness of the novel Merge Crossover, Inbreed Crossover Mutation and Cost-Aware Swap Mutation operators are particularly evident in the test results discussed in Section 5.
- Controlling the evolution based not only on swarm performance but also on the preservation of particular building blocks that possess good genetic traits.
- As solutions to this problem type being more prone to deception as a building block evolved for one pair of take-off and landing instances and in a certain swarm synchronization would not easily fit into another environment, escaping deception through the employment of a wide spectrum of operators in coherence.

By comprehensively comparing its performance against serialized GA-CMoD algorithm that was originally designed for single drone problem type and adapted for multiple drone solution through serial execution logic recursively, we have tested the effectiveness of EOSRP to eminently satisfactory results.

Serialized GA-CMoD also proved to be a powerful algorithm in handling this problem type. However, contrary to EOSRP, it lacks the awareness of the capabilities of the other drones that are available for employment during the mission. Therefore the distinction between the performance of these two algorithms is due to synchronous planning capability of one over the other. This observation indicates that our primary purpose of proposing an approach that enables effectively synchronized planning of multiple drones for a mission has been satisfactorily fulfilled.

Future directions to further extend this type of optimization should include:

- A similar environment this time with multiple mobile depots on the move on separate routes. In such a problem major scale delivery operations with a network of multiple drones and multiple mobile depots serving a large area with dispersed sectors can be modeled, where improving operational efficiency with these constraints are searched. Further complexity introduced by these constraints poses a serious challenge for evolutionary meta-heuristic approaches.

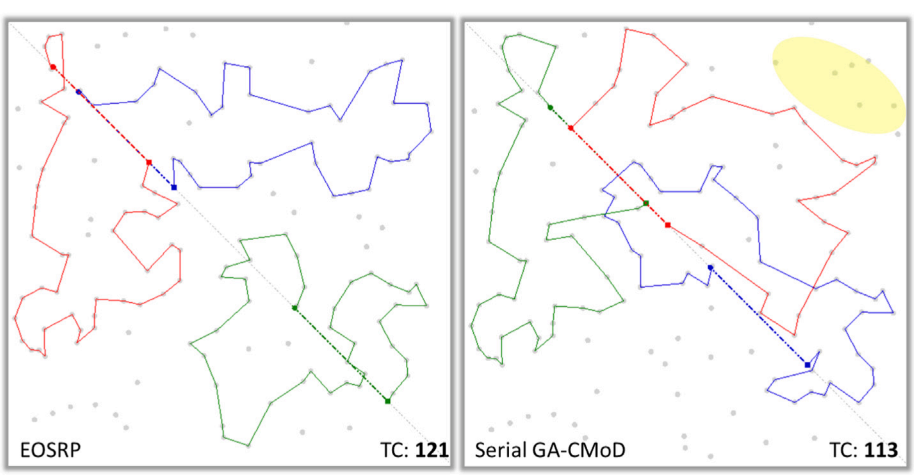

Fig. 30. Comparison of tours delivered by both algorithms for ch150 three drones scenario.

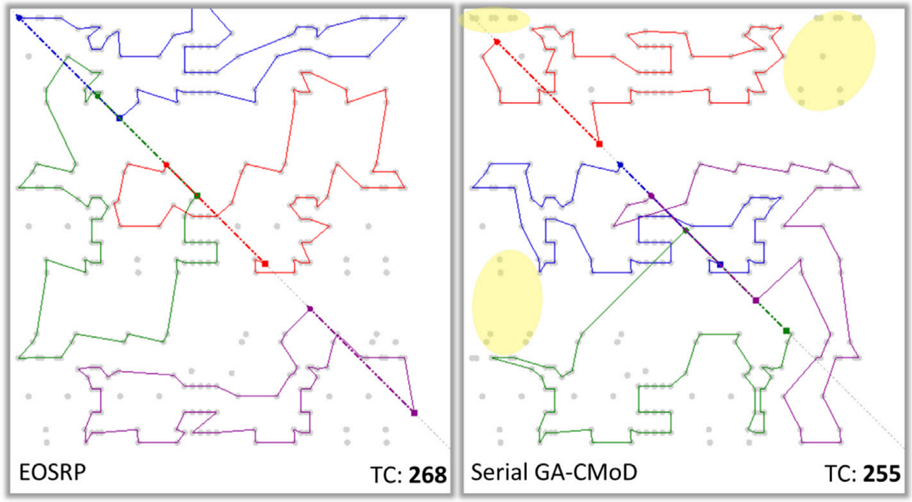

Fig. 31. Comparison of tours delivered by both algorithms for linhp318 four drones scenario.

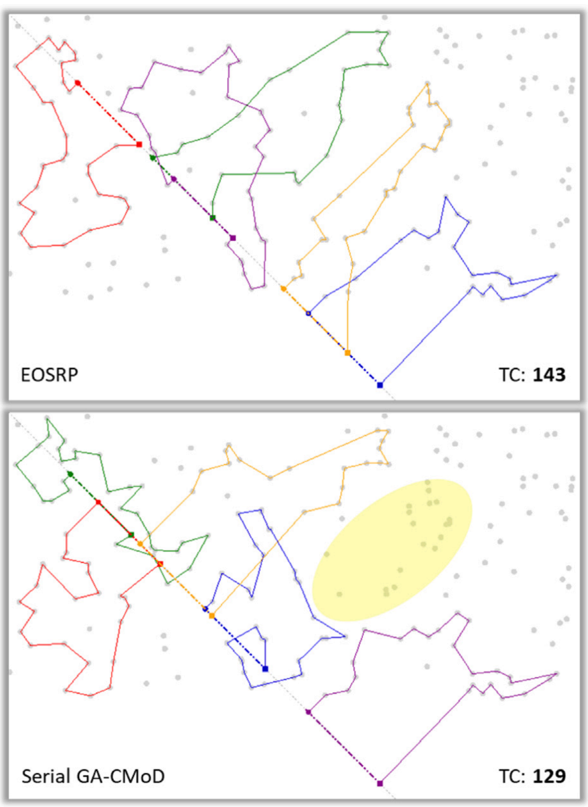

Fig. 32. Comparison of tours delivered by both algorithms for krob200 five drones scenario.

- Addition of fleet sizing into decision variables of the problem. With the purpose of visiting a given minimum percentage of targets, optimizing the size of the fleet to be allocated for the delivery mission can be aimed. This model addresses the practical problem of effective overall utilization of drones in inventory.
- Addition of route design of the mobile depot as another problem variable. In such a scenario mobility of the host station would be utilized in favor of the mission efficiency by altering the assumption that it has other operational purposes.

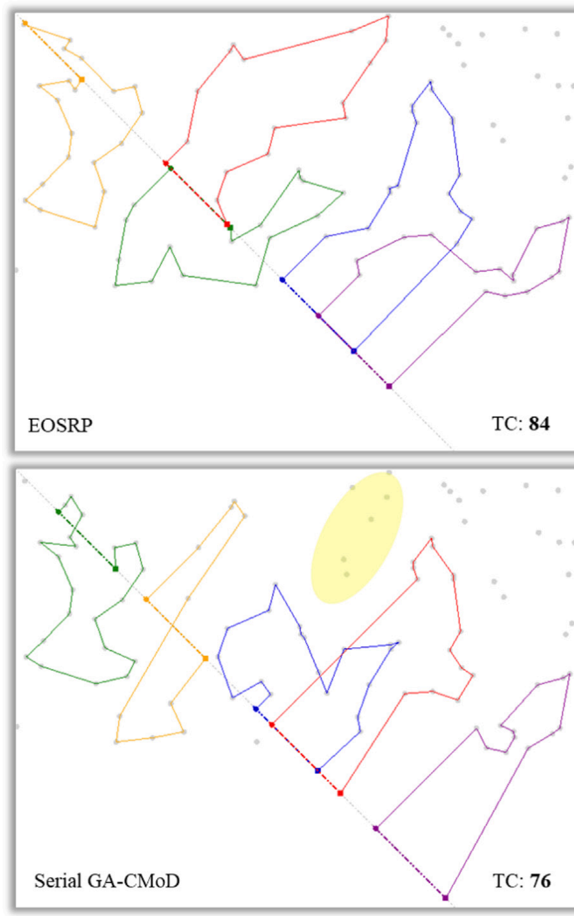

Fig. 33. Comparison of tours delivered by both algorithms for krob100 five drones scenario.

CRediT authorship contribution statement

Halil Savuran: Writing – review & editing, Writing – original draft, Visualization, Software, Methodology, Formal analysis, Conceptualization. **Murat Karakaya:** Writing – review & editing, Writing – original draft, Supervision, Methodology, Formal analysis, Conceptualization.

Declaration of competing interest

We, as the authors of this work declare that we have no financial or personal interest or belief that could affect our objectivity.

Acknowledgment

The code and testbed utilized in this study are publicly available in the GitHub repository at https://github.com/HalilSvrn/Code-Space-for-EOSRP.

References

Agatz, N., Bouman, P., Schmidt, M., 2018. Optimization approaches for the traveling salesman problem with drone. Transp. Sci. 52 (4), 965–981.
Araujo, L., Merelo, J.J., 2010. Diversity through multiculturality: Assessing migrant choice policies in an island model. IEEE Trans. Evol. Comput. 15 (4), 456–469.
Baik, H., Valenzuela, J., 2021. An optimization drone routing model for inspecting wind farms. Soft Comput. 25 (3), 2483–2498.
Bouman, P., Agatz, N., Schmidt, M., 2018. Dynamic programming approaches for the traveling salesman problem with drone. Networks 72 (4), 528–542.
Bremermann, H.J., et al., 1962. Optimization through evolution and recombination. Self-Organ. Syst. 93, 106.
Cantú-Paz, E., 1998. A survey of parallel genetic algorithms. Calc. Paralleles Reseaux Syst. Repartis 10 (2), 141–171.
Cantú-Paz, E., 2001. Migration policies, selection pressure, and parallel evolutionary algorithms. J. Heuristics 7 (4), 311–334.
Cantu-Paz, E., Goldberg, D.E., 2000. Efficient parallel genetic algorithms: theory and practice. Comput. Methods Appl. Mech. Engrg. 186 (2–4), 221–238.
Carlsson, J.G., Song, S., 2018. Coordinated logistics with a truck and a drone. Manage. Sci. 64 (9), 4052–4069.
Chang, Y.S., Lee, H.J., 2018. Optimal delivery routing with wider drone-delivery areas along a shorter truck-route. Expert Syst. Appl. 104, 307–317.
Chen, Y., Ren, S., Chen, Z., Chen, M., Wu, H., 2019. Path planning for vehicle-borne system consisting of multi air–ground robots. Robotica 1–19. http://dx.doi.org/10.1017/s0263574719000808.
Chiang, W.-C., Li, Y., Shang, J., Urban, T.L., 2019. Impact of drone delivery on sustainability and cost: Realizing the UAV potential through vehicle routing optimization. Appl. Energy 242, 1164–1175.
Croes, G., 1958. A method for solving traveling-salesman problems. Oper. Res. 6 (6), 791–812.
Dantzig, T., 1931. Number: The language of science. Bull. Amer. Math. Soc. 37 (9).
Dantzig, G.B., Ramser, J.H., 1959. The truck dispatching problem. Manage. Sci. 6 (1), 80–91. http://dx.doi.org/10.1287/mnsc.6.1.80.
Das, D.N., Sewani, R., Wang, J., Tiwari, M.K., 2020. Synchronized truck and drone routing in package delivery logistics. IEEE Trans. Intell. Transp. Syst..
de Freitas, J.C., Penna, P.H.V., 2020. A variable neighborhood search for flying sidekick traveling salesman problem. Int. Trans. Oper. Res. 27 (1), 267–290.
Dell Amico, M., Montemanni, R., Novellani, S., 2020. Matheuristic algorithms for the parallel drone scheduling traveling salesman problem. Ann. Oper. Res. 289 (2), 211–226.
Dell Amico, M., Montemanni, R., Novellani, S., 2021. Drone-assisted deliveries: New formulations for the flying sidekick traveling salesman problem. Optim. Lett. 15 (5), 1617–1648.
Fogel, D.B., 2000. What is evolutionary computation? IEEE Spectr. 37 (2), 26–32.
Fraser, A.S., 1957. Simulation of genetic systems by automatic digital computers I. Introduction. Aust. J. Biol. Sci. 10 (4), 484–491.
García-Martínez, C., Rodriguez, F.J., Lozano, M., 2018. Genetic algorithms.
Graham, R.L., 1966. Bounds for certain multiprocessing anomalies. Bell Syst. Tech. J. 45 (9), 1563–1581.
Ha, Q.M., Deville, Y., Pham, Q.D., Hà, M.H., 2018. On the min-cost traveling salesman problem with drone. Transp. Res. C 86, 597–621.
Hamilton, W.R., 1856. Memorandum respecting a new system of roots of unity. Phil. Mag. 12 (446), 1856.
Holland, H.J., 1975. Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. U Michigan Press.
Hoos, H.H., Stützle, T., 2004. Stochastic Local Search: Foundations & Applications. Elsevier.
Jeong, H.Y., Song, B.D., Lee, S., 2019. Truck-drone hybrid delivery routing: Payload-energy dependency and No-Fly zones. Int. J. Prod. Econ. 214, 220–233.
Karak, A., Abdelghany, K., 2019. The hybrid vehicle-drone routing problem for pick-up and delivery services. Transp. Res. C 102, 427–449.
Kitjacharoenchai, P., Min, B.-C., Lee, S., 2020. Two echelon vehicle routing problem with drones in last mile delivery. Int. J. Prod. Econ. 225, 107598.
Kitjacharoenchai, P., Ventresca, M., Moshref-Javadi, M., Lee, S., Tanchoco, J.M., Brunese, P.A., 2019. Multiple traveling salesman problem with drones: Mathematical model and heuristic approach. Comput. Ind. Eng. 129, 14–30.
Larranaga, P., Kuijpers, C.M.H., Murga, R.H., Inza, I., Dizdarevic, S., 1999. Genetic algorithms for the travelling salesman problem: A review of representations and operators. Artif. Intell. Rev. 13 (2), 129–170.
Liu, Y., Luo, Z., Liu, Z., Shi, J., Cheng, G., 2019. Cooperative routing problem for ground vehicle and unmanned aerial vehicle: The application on intelligence, surveillance, and reconnaissance missions. IEEE Access 7, 63504–63518. http://dx.doi.org/10.1109/access.2019.2914352.
Luo, Z., Liu, Z., Shi, J., 2017. A two-echelon cooperated routing problem for a ground vehicle and its carried unmanned aerial vehicle. Sensors 17 (5), 1144.
Luo, Z., Poon, M., Zhang, Z., Liu, Z., Lim, A., 2021. The multi-visit traveling salesman problem with multi-drones. Transp. Res. C 128, 103172.
Marinelli, M., Caggiani, L., Ottomanelli, M., Dell'Orco, M., 2018. En route truck–drone parcel delivery for optimal vehicle routing strategies. IET Intell. Transp. Syst. 12 (4), 253–261.
Mascato, P., 1989. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. p. 67.
Menger, K., 1932. Das botenproblem. Ergeb. Eines Math. Kolloquiums 2 (4), 11–12.
Moshref-Javadi, M., Lee, S., Winkenbach, M., 2020. Design and evaluation of a multi-trip delivery model with truck and drones. Transp. Res. E 136, 101887.
Murray, C.C., Chu, A.G., 2015. The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. Transp. Res. C 54, 86–109. http://dx.doi.org/10.1016/j.trc.2015.03.005.
Murray, C.C., Raj, R., 2020. The multiple flying sidekicks traveling salesman problem: Parcel delivery with multiple drones. Transp. Res. C 110, 368–398.
Nguyen, M.A., Dang, G.T.-H., Hà, M.H., Pham, M.-T., 2021. The min-cost parallel drone scheduling vehicle routing problem. European J. Oper. Res..
bin Othman, M.S., Shurbevski, A., Karuno, Y., Nagamochi, H., 2017. Routing of carrier-vehicle systems with dedicated last-stretch delivery vehicle and fixed carrier route. J. Inf. Process. 25, 655–666.
Peng, K., Du, J., Lu, F., Sun, Q., Dong, Y., Zhou, P., Hu, M., 2019. A hybrid genetic algorithm on routing and scheduling for vehicle-assisted multi-drone parcel delivery. IEEE Access 7, 49191–49200. http://dx.doi.org/10.1109/access.2019.2910134.
Poikonen, S., Golden, B., 2019. The mothership and drone routing problem. INFORMS J. Comput. http://dx.doi.org/10.1287/ijoc.2018.0879.
Poikonen, S., Golden, B., 2020. Multi-visit drone routing problem. Comput. Oper. Res. 113, 104802.
Poikonen, S., Wang, X., Golden, B., 2017. The vehicle routing problem with drones: Extended models and connections. Networks 70 (1), 34–43.
Reinelt, G., 1991. TSPLIB-A traveling salesman problem library. ORSA J. Comput. 3 (4), 376–384.
Sacramento, D., Pisinger, D., Ropke, S., 2019. An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones. Transp. Res. C 102, 289–315.
Savuran, H., Karakaya, M., 2015a. Efficient route planning for an unmanned air vehicle deployed on a moving carrier. Soft Comput. 20 (7), 2905–2920. http://dx.doi.org/10.1007/s00500-015-1970-4.
Savuran, H., Karakaya, M., 2015b. Route optimization method for unmanned air vehicle launched from a carrier. Lect. Notes Softw. Eng. 3 (4), 279–284. http://dx.doi.org/10.7763/LNSE.2015.V3.204.
Schermer, D., Moeini, M., Wendt, O., 2019a. A hybrid VNS/Tabu search algorithm for solving the vehicle routing problem with drones and en route operations. Comput. Oper. Res. 109, 134–158.
Schermer, D., Moeini, M., Wendt, O., 2019b. A matheuristic for the vehicle routing problem with drones and its variants. Transp. Res. C 106, 166–204.
Sorensen, K., Sevaux, M., Glover, F., 2017. A history of metaheuristics. arXiv preprint arXiv:1704.00853.
Wang, D., Hu, P., Du, J., Zhou, P., Deng, T., Hu, M., 2019. Routing and scheduling for hybrid truck-drone collaborative parcel delivery with independent and truck-carried drones. IEEE Internet Things J. 6 (6), 10483–10495.
Wang, Z., Sheu, J.-B., 2019. Vehicle routing problem with drones. Transp. Res. B 122, 350–364.
Wang, K., Yuan, B., Zhao, M., Lu, Y., 2020. Cooperative route planning for the drone and truck in delivery services: A bi-objective optimisation approach. J. Oper. Res. Soc. 71 (10), 1657–1674.
Whitley, D., 1994. A genetic algorithm tutorial. Stat. Comput. 4 (2), 65–85.
Whitley, D., Yoo, N.-W., 1995. Modeling simple genetic algorithms for permutation problems. In: Foundations of Genetic Algorithms. Vol. 3, Elsevier, pp. 163–184.
Yurek, E.E., Ozmutlu, H.C., 2018. A decomposition-based iterative optimization algorithm for traveling salesman problem with drone. Transp. Res. C 91, 249–262.