On Optimising Topology of Agricultural Fields for Efficient Robotic Fleet Deployment

Zuyuan Zhu, Gautham P. Das, and Marc Hanheide

Lincoln Centre for Autonomous Systems (L-CAS), University of Lincoln,
Brayford Way, Lincoln, LN6 7TS, UK,
zzhu, gdas, mhanheide@lincoln.ac.uk

Abstract. Field-deployed robotic fleets can provide solutions that improve operational efficiency, control operational costs, and provide farmers with transparency over the day-to-day operations with scouting operations. The topology of agricultural farms such as polytunnels provides a basic environmental configuration that can be exploited to create a topological map to aid operational planning and robot navigation. However, these environments are optimised for operations by humans or for large farming vehicles and pose a major challenge for multiple moving robots to coordinate their navigation while performing tasks. The farm environment without any topological modifications for supporting robotic fleet deployments can cause traffic bottlenecks, eventually affecting the overall efficiency of the fleet. In this work, we propose a Genetic Algorithm-based Topological Optimisation (GATO) algorithm that discretises the search space of topological modifications into finite integer combinations. Each solution is encoded as an integer vector that contains the location information of the topology modification. The algorithm is evaluated in a discrete event simulation of the picking and in-field logistics process in a commercial strawberry farm and the results validate the effectiveness of our algorithm in identifying the topological modifications that improve the efficiency of the robotic fleet operations.

Keywords: robot traffic planning, multi-robot systems, agri-robotics, topological optimisation, discrete event simulation, genetic algorithm

1 Introduction

Autonomy is becoming increasingly important for robotic systems from factory floors to self-driving vehicles, integrating into our everyday lives [1]. Currently, robotic systems contribute to sustainable development by bridging the gaps between the global economy, social needs, and logistics. Any sustainable solution that reduces the final cost of the product, the environmental impacts of agriculture or the physical strain on human operators is welcome in a scenario of global economic growth [2].

Farming of high-value crops, e.g., soft-fruits such as strawberries, which heavily depend on manual labour, are ideal candidates for the future of robotic agriculture utilising fleets of electric, autonomous systems powered by renewable energy [3]. These robotic systems will perform tasks such as picking, transporting, and packing fruit while also collecting data to maximise yields, minimise waste, and minimise environmental impact.

The topology of a farm environment could be utilised to make a topological map that can underpin the navigation of the robots [19] and for planning the operations of the robots [21]. A topological map offers a discrete representation of the environment that enables robot navigation without necessarily maintaining the robot's position in a global frame of reference. It also provides an abstraction through topology information that can assist in advanced planning and inference tasks.

In this work, we are considering the deployment of a robotic fleet in a polytunnel environment to support human pickers with in-field logistics operations. In this, the tasks are collecting fruits from human pickers on-demand and delivering them to a drop-off point (or storage). This is a complex scenario due to the presence of human agents (human safety) as well as the dynamic appearance of new tasks (pickers requesting robotic runners). When robot fleets perform logistics tasks, the primary aim is to minimise the travelling time that robots spent on farms, executing the tasks (navigating to the pickers and back to the drop-off point). Reducing time means reducing travelling distance, which requires the path planner to find the combination of routes for multiple robots resulting in the shortest travelling time (also known as multi-robot path planning, MRPP). However, in the navigation of multiple robots sharing an environment, deadlocks are almost unavoidable when trying to find the shortest routes for multiple robots. A balance between minimising the deadlocks and travelling distance at the same time could be achieved by optimising the topological map that the robots use for navigation.

In this paper, we investigate the autonomous optimisation of a topological map using a Genetic Algorithm (GA), extending our previous work [4] by exploring further topology modification strategies. The GA uses two strategies: 1) adding cross lanes across the farm field; 2) allocating multiple drop-off points (storages) at different regions of the field. To the best of the author's knowledge, using multiple strategies for optimising the topology of agricultural fields for robotic deployments is proposed for the first time in this paper.

To economically evaluate our results, we use the example of multi-robot systems delivering in-field logistics tasks in strawberry farms and incorporate a genetic algorithm (GA) with Discrete Event Simulation (DES) for estimating the fitness. We empirically evaluate the proposed GA-based topological optimisation based on the improvements in the overall simulation time and the number of deadlocks.

2 Related Work

A common approach to increase the task execution efficiency of a multi-robot system (MRS) is to ensure the tasks are completed in the shortest time. Algorithms that enable an MRS to perform the desired tasks as quickly as possible have been widely studied, especially on task scheduling [5], [6], [7].

Gerkey and Matarić [8] believe that multi-robot task scheduling problems, in the absence of a theoretical foundation to explain or predict the behaviour of a multi-robot system, are essentially organizational in nature. The system aims to allocate limited resources efficiently to accomplish a task.

More recently, Digani et al. [7] proposed a quadratic optimisation method for coordinating a fleet of vehicles in automated warehouses aimed at lowering the amount of time that mobile robots spend manoeuvring through intricate traffic arrangements.

Nishi et al. [6] tackle the simultaneous scheduling and conflict-free routing problems for automated guided vehicles through a bi-level decomposition algorithm. The algorithm decomposes the problems into two parts: a master problem at the upper level concerning task assignment and scheduling, and a conflict-free routing sub-problem at the lower level. Similarly, [5] put forth a hybrid technique that involves a decomposition method, in which the scheduling task is modelled through constraint programming and the routing task (which must be conflict-free) is modelled using mixed integer programming. The sub-problem then generates logic cuts that are utilized by the master problem to eliminate scheduling solutions with conflicting routing plans.

In the context of a warehouse logistics application, Farinelli et al. [9] examined the problem of scheduling tasks for a group of robots. The robots' objective is to transport items from loading bays to unloading bays in order to complete packages for delivery to customers. To maximize the number of packages completed in a given time frame, the robots must work together and avoid interfering with one another while moving within the environment. This problem was formalised as a distributed constrained optimisation problem, and a solution was presented using the binary max-sum algorithm.

As we can see, the task scheduling problem has been approached using several techniques, including Mixed Integer Programming (MIP) approaches [10], reactive methods [11], and biologically inspired approaches like ant colony algorithm [12]. One of the most commonly utilised methods for task assignment is the market-based framework [13], where robots trade valuations (bids) to perform tasks and each task is allocated to the robot with the highest valuation for that task.

On the other hand, optimisation of a topological map is another proffered way to improve the MRS efficiency by reducing the deadlocks and shortening the total travelling distance. Despite the fact that several researchers [14], [15], [16] have dealt with the task scheduling using GAs, they have not taken into consideration optimising the topology of the navigation map to address the MRPP challenges.

Topological optimisation has received extensive attention in the literature as the performance of tasks such as localisation and navigation is closely tied to the topological map [17]. To enhance navigation efficiency, the robotic system must first have an effective localisation method, which can be accomplished through the use of image clusters based on appearance similarities [18] or laser scanners [19].

The primary benefit of GAs over gradient-optimisation methods is that the GAs do not require the derivative of the objective function, making them capable of discovering the near-global optimum of non-continuous and procedural functions. Furthermore, GAs are simple to understand and require a minimal mathematical background.

In this paper, a method is introduced to determine the optimum topology of the navigation map that an MRS used for executing in-field logistics tasks across the farm and it can be applied to any topological maps. This method is based on a GA and a DES framework is introduced to simulate the picking and transporting tasks in a polytunnel environment.

3 Methodology

3.1 Genetic Algorithms

In this work, we considered the optimisation of the topological map in two ways: adding cross lanes to the polytunnel rows and allocating drop-off points to the head/rear nodes area of each polytunnel. We aim to decrease or eliminate the deadlocks for multi-robot systems when delivering logistics tasks and improve overall transporting efficiency by reducing the robots' travelling time on the farm. Cross-lanes can reduce the overall distance a robot may have to travel inside a polytunnel row and provide parallel routes that may reduce deadlocks. The implementation of GA elements in adapting a GA algorithm for the topological optimisation problem is described in the following sections.

Chromosome coding and encoding The selection of the solution representation (Chromosome) impacts the transformation and evaluation methods. For adding cross lanes and allocating drop-off points, we use different representations to encode the chromosome. The idea is to use vector numbers to represent the location of cross lanes and drop-off points. Therefore, we first mark each interested area with unique numbers, then we use these numbers as data sets to generate random integer vectors, i.e., chromosomes.

- *Adding cross lanes.* We use an integer vector which contains a permutation of task element (activity) numbers, i.e., a vector of pointers to the position of cross lanes to be added. As shown in Fig. 1, the chromosome contains 3 integers (1,3, and 4) indicating the 1st, 3rd, and 4th columns of the map are connected with cross lanes respectively. When nodes are not aligned, the cross lane crosses the nearest neighbouring node of the next row. The GA always selects the longest row and creates cross lanes in opposite directions. For example, in Fig. 1, the map has three rows, assume the middle row has more or the same number of nodes as others. The GA selects a node (1st, 3rd, or 4th) from the middle row and extends it to both edges of the map. This ensures all nodes have a chance to be part of the cross-lane.
- *Allocating drop-off points.* As shown in Fig. 2, the chromosome contains 2 integers (1 and 4), meaning that the head and rear area of the second polytunnel will be allocated with a drop-off point respectively. Here, we use the serial number, starting from 0, to indicate the position of

the areas where new drop-off points could be allocated. We assume the top side is polytunnel heads and the bottom areas are polytunnel rears. The length of the chromosome could be any integer between 1 and the size of polytunnels (in the example of Fig. 2, the size is 3).

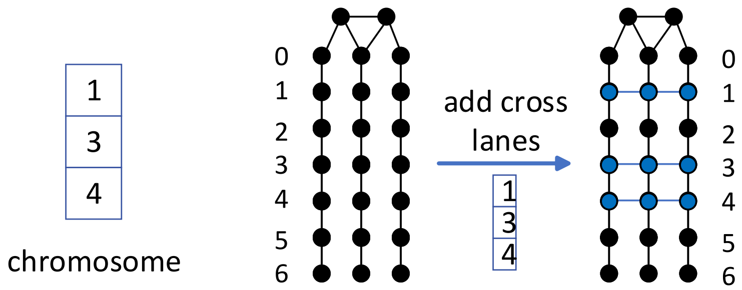


Fig. 1: Example of a chromosome and using it for adding cross lanes on the topological map.

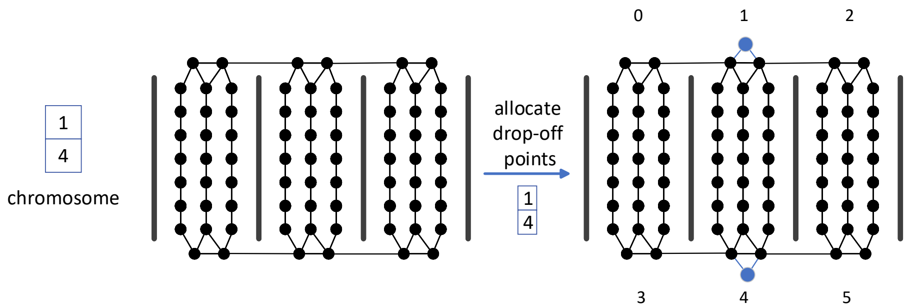


Fig. 2: Example of a chromosome and using it for allocating drop-off points on the topological map.

Crossover and Mutate Crossover is a recombination operator that occurs after reproduction. Its purpose is to create new individuals for the next generation by combining parts of existing ones. Individuals are randomly chosen based on a predefined probability (crossover rate). With two chromosomes (i.e., two integer vectors with many attributes), part of the vectors are exchanged. If any of the two new chromosomes contain duplicated attributes, then the crossover operator would be cancelled. For example, [1 3 5 7] and [2 1 4 6] are two chromosomes and the middle two attributes are selected to exchange. Then the crossover would be restored as the newly generated chromosome [1 1 4 7] contains two same attributes 1.

The mutation operation is applied to perform a random modification on a chromosome's attributes with a small-predefined probability (mutation rate). An independent probability is set for each attribute to be flipped. When each attribute mutates, the new attribute is randomly selected from the remaining data sets, ensuring no duplicated attribute would be generated.

Control parameters

– Population size. The size of the population determines the number of chromosomes and determines how much genetic material is available in the genetic search process. It must be noted that a smaller population size covers a smaller region of the search space, meaning it may not

be a representative sample of the solution. Therefore, a small population size reduces the likelihood of finding the global optimum. On the other hand, a larger population size significantly increases the computational complexities and CPU time for each iteration of the GA. The population size depends on the properties and complexity of the current problem. In this work, the most time-consuming process is evaluating the fitness of each chromosome, i.e., simulating the multi-robot systems delivering logistics tasks. Among the two topological maps used for evaluation, the fitness calculation per chromosome is approximately 5 s for the smaller map and approximately 6 minutes for the larger map.
– Generation size. This is the number of iterations that the GA needs to run before converging to a solution for a given optimisation problem. Generation size is one of the criteria to stop the GA.
– Crossover rate. The crossover rate determines the frequency at which the crossover operator is applied to the population chromosomes, thus producing a new population. The higher the crossover rate, the more individuals are introduced to the new population. The crossover rate is usually set between 0.5 and 1.0, and in our example, 0.5 was chosen after extensive testing.
– Mutation rate. The mutation rate determines the changing probability of genetic value in chromosomes. It should be noted that each attribute of the chromosome is also controlled by an individual probability. Mutations introduce new areas of unexplored search space. However, the mutation rate should not be too high, as it increases the randomness of the search. The mutation rate is usually around 0.5 and the individual rate is below 0.2, and in our example, after multiple trials, 0.5 and 0.1 were selected respectively.
– Tournament size. This is the number of individuals participating in each tournament. In each tournament, select the best individual among the tournament size randomly chosen individuals, for population size times. Then a new generation of the population is generated.

3.2 Discrete Event Simulation

In this paper, we use Discrete Event Simulation (DES) as a tool for evaluating the chromosomes generated by GA. These chromosomes are representations of topological modifications.

In a DES system model, the basic unit that changes its state is called an entity and they compete for limited resources [20]. The entities described in this DES are human workers and robots on a farm. Resources are elements that provide a certain service, usually with limited capacity. In this agricultural in-field logistics scenario, robots are examples of resources providing transport services and their number is limited. Also, the drop-off points where picked fruit trays are transported to and temporally stored by robots, are resources with limited resources that only allow one robot to use at any moment. Steps performed or executed on an entity are called operations. In the case of a robot assistant, assisting a designated worker at a specific location is an operation. An event is a set of operations that occur at a certain time instance and result in a change in the system state, which is composed of the states of each entity. DES skips different event times and updates the state of entities and the entire system. Therefore, it allows running simulations very fast, allowing the efficient evaluation of GA chromosomes. More details of the human pickers and robots performing picking and delivering tasks can be found in [21] and [22]. The process-based discrete-event simulation framework we used is SimPy [23].

3.3 Genetic Algorithm based Topological Optimisation (GATO) Algorithm

When introducing a mobile robotic fleet to an environment, the simplest approach is to develop the robot's navigation autonomy without altering the environment's topology. However, this results in sub-optimal performance as the topology wasn't designed with robot operations in mind. On the other hand, constructing the environment's topology from scratch to optimise it for the robots is challenging and costly. A compromise between these two extremes is to make minimal changes to the topology to enhance the efficiency of the fleet's operation. However, determining the precise modifications from the available options remains challenging.

Algorithm 1 GATO

$t \leftarrow 0; t \in N$
$GenerateInitialPopulation(P(t))$
while $iteration < allowedIteration$ **do**
    $evaluateInDES(P(t))$
    $P'(t) \leftarrow selectBestIndividual(P(t))$
    $P"(t) \leftarrow selectPopForCrossover(P'(t))$
    $P(t+1) \leftarrow mutatePopulation(P"(t))$
    $t \leftarrow t+1$
end while
return $fittestMemeberOfPopulation(P)$

The proposed solution in this work is the Genetic Algorithm based Topological Optimization (GATO) algorithm for MRS in logistics tasks. This algorithm uses a GA to iteratively find high-quality topology modifications, and a Discrete Event Simulation (DES) of picking and logistics operations as a low-cost computational fitness function to evaluate the quality of each candidate solution. The GATO algorithm is designed to find the best locations of cross lanes and drop-off points and it can be adapted to handle more complex topology modifications.

The pseudocode for the GATO algorithm is shown in Algorithm 1, where the termination criterion is set by a fixed number of iterations. In generating the initial population, each individual is represented as a one-dimensional list with a set number of genes (topological nodes). The DES is run for each individual to simulate the picking and transporting tasks, producing a result consisting of simulation time and deadlock number. This result is then multiplied by a negative 2×1 weight to obtain the fitness of the individual, as illustrated in (1).

$$f = \left[t_{sim}\ n_{dl}\right] \times \left[w_1\ w_2\right]^T \tag{1}$$

where f means fitness, t_{sim} is the simulation time, n_{dl} means the number of deadlock. The weight vector, $[w_1\ w_2]$, in our approach is used to optimise multiple objectives, with a focus on minimising both the simulation time and deadlock number. During the mutation and crossover stages, the topological nodes of each individual can be altered or swapped.

4 Evaluation

We evaluated the topology optimisation performance of the proposed GATO algorithm with two strawberry farms: a small farm – Clockhouse farm transportation map (CHF transportation, see Fig. 3c) with four polytunnels of approximately 130m in length and 30m in width, and map of a commercial strawberry production field – Clockhouse farm Vanity map (CHF Vanity, see Fig. 3a) with twenty polytunnels of approximately 240m in length and 150m in width. The CHF transportation map has about 823±14 nodes and the vanity map has about 5541±14 nodes. For the transportation map, each evaluation of the GA individual takes about 5 seconds, overall about 400 evaluations, take about 2000 s, i.e., 33min. For the vanity map, each evaluation of the GA individual takes about 5 min. Overall for about 130 evaluations, it takes about 650 min.

4.1 Experimental setup

The main parameters of GA are presented in Table 1. To get a high-quality solution in a reasonably short time and acknowledge the fact that having a large population leads to the accuracy of getting an optimal solution, we select a partial population from the complete data set. The complete populations are presented in Table 2. For the Vanity map, the population size is $C_{85}^{n_c}, n_c \in [1, 5]$, i.e., 85-32801517, when adding cross lanes. The maximum size is huge, which means it is not practical to use the full size and get an optimum solution. In this paper, we use a size of 30 or 50.

Table 1: GA parameters

Parameter	Value	Note
N_POP	50/30	the number of initial individual population
N_GEN	6	the number of generations to run
INDPB	0.1	the probability of mutate each attribute/gene of the individual
N_LANE	1-5	the number of gene (cross lane) that each individual consists
N_DROP_OFF	1-4	the number of gene (drop-off point) that each individual consists
CXPB	0.5	the probability with which two individuals are crossed
MUTPB	0.50	the probability for mutating an individual
N_TOU	10	the parameter for selecting individuals for breeding the next generation: each individual of the current generation is replaced by the 'fittest' of N_TOU individuals drawn randomly from the current generation
WEIGHTS	[1.0, 4.0]	$[w_1 \ w_2]$ in formula (1), used to vary the importance of each objective one against another, a minimising fitness is built using negatives weights

Table 2: GA population size

Population	CHF transportation	CHF vanity
Cross lanes	$C_{46}^{n_c}$	$C_{85}^{n_c}$
Drop-off points	$C_{8}^{n_d}$	$C_{40}^{n_d}$
Note	$n_c \in [1, 5]$, no. of cross-lanes	$n_d \in [1, 4]$, no. of drop-off points

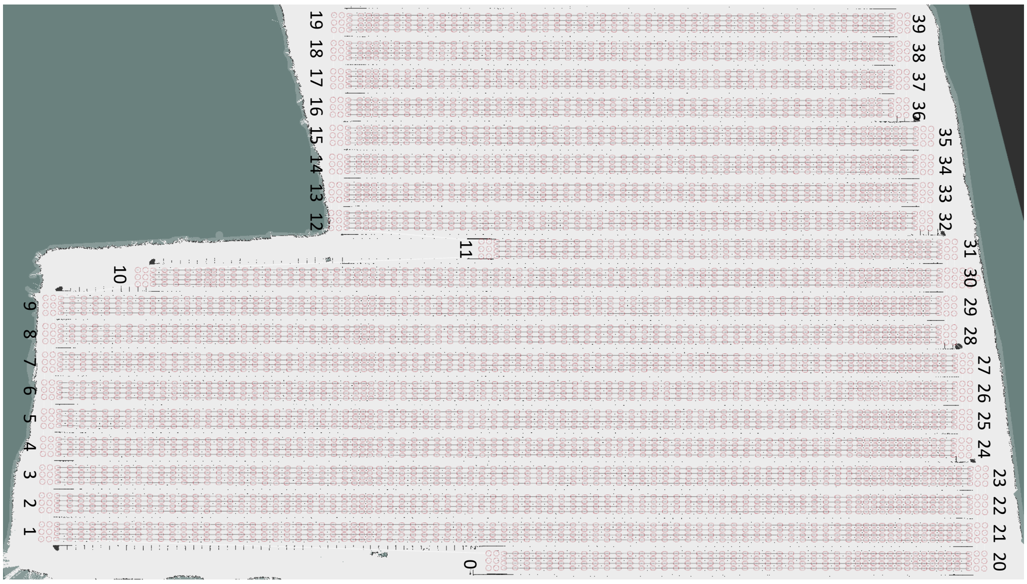
4.2 Adding cross lanes

When using GATO for optimising topology, we make the following assumptions:

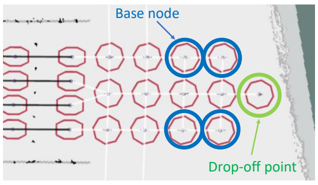
1. There are 4 pickers and 4 robots performing picking and delivering tasks in cooperation. Whenever a picker has full trays, a call request will be sent to an idle robot and the robot will come to the picker and deliver the trays to the drop-off point.
2. The cross lane begins from one side of the farm to the end of another side, crossing the whole farm. When adding the cross lane, if the two neighbour nodes are not aligned, the GATO will link the nearest node from another row.
3. The fruit yield will not be affected by the added cross lanes. However, in the real strawberry farm, the raised beds used for growing strawberries have to be removed for letting the robots cross the lanes.
4. The maximum cross lanes that we can add is 5 and the minimum is 1. This range is selected for the field we consider in this work. For a different field with polytunnels of a different length, the range of cross-lanes one could explore can be different from this. Adding too many cross-lanes can reduce the overall performance due to increased graph complexity and corresponding computational costs.

In order to investigate how the number of cross-lanes influences the performance of MRS, we run the GATO with 1/2/3/4/5 lanes respectively and summarised the results in Fig. 4.

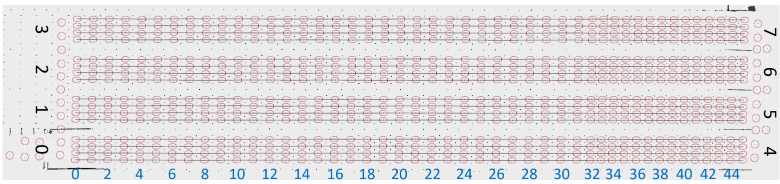
Fig. 4a shows the overall performance of the MRS performing logistics tasks against different numbers of cross lanes with CHF transportation map. As shown in Fig. 4a, it surprisingly indicates that using more cross-lanes does not mean better performance. The fitness curve shows that adding two lanes to the CHF small map achieves the best comprehensive performance. Though 2 cross lanes do not bring the least deadlock (the blue solid line), they ensure the shortest simulation time (the red solid line). The deadlocks curve shows that using more cross lanes leads to a higher likelihood of encountering deadlocks as robots tend to converge on the shortest routes to the same destination. Moreover, when planning a new route, the robots try to avoid paths occupied by other robots, which may result in a longer planned route and an increased travel time. With fewer cross lanes, robots are separated by rows and travel on different routes. As shown in Fig. 5, the deadlocks heatmap shows that using 5 cross lanes brings more deadlocks than using 2 cross lanes. The right



(a) Aerial view of the CHF vanity map

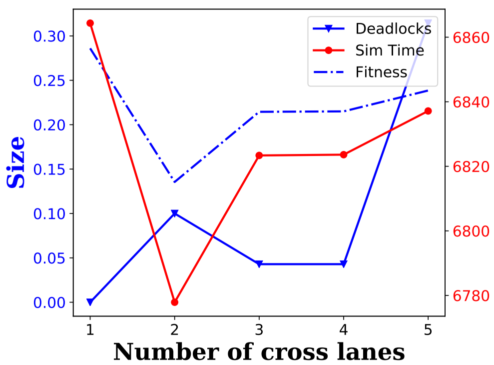


(b) Drop-off and base

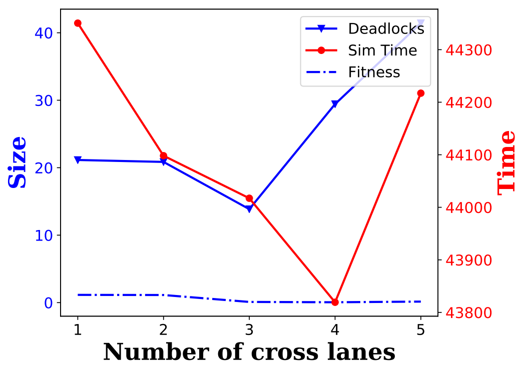


(c) Aerial view of the CHF transportation map

Fig. 3: Topological representation of the CHF vanity and transportation map. (a) The potential drop-off points of CHF vanity map are marked with 40 numbers (0-39); (b) example of a typical drop-off point which consists of a drop-off node, four base nodes for robot waiting when the drop-off point is occupied by another robot, and some nodes for navigation; (c) the potential drop-off points of CHF transportation map are marked with 8 numbers (0-7) and the potential cross-lane columns are marked with 46 numbers (0-45). A drop-off point could be allocated at the position of any black number and a cross lane could be added at the column of any blue number. All topological maps have head lanes at both ends.

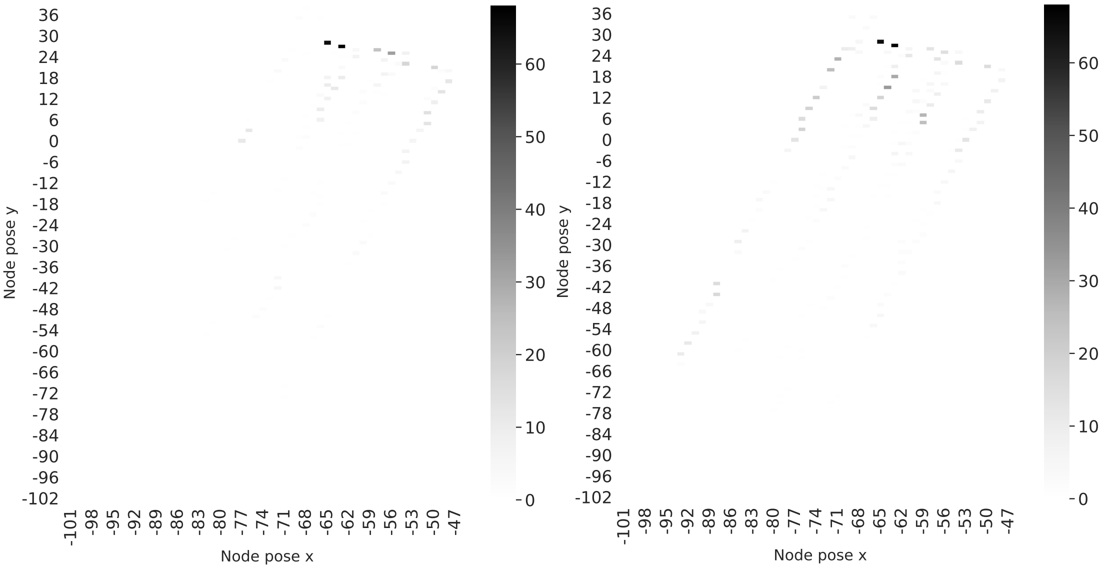


(a) CHF transportation map



(b) CHF Vanity map

Fig. 4: MRS performance when adding cross lanes: deadlocks and sim time



(a) Adding 2 cross lanes

(b) Adding 5 cross lanes

Fig. 5: Heatmap distribution of the accumulated deadlocks over the GA generations with the CHF small map after adding cross lanes: (a) with two cross lanes added, (b) with five cross lanes added. The horizontal and vertical axes indicate the positions of deadlocks. The grey bar on the right displays the frequency of deadlocks. Dark blocks imply many deadlocks, white blocks imply no deadlocks, and shallow grey blocks imply very few deadlocks.

figure shows more deadlocks occurring throughout the map, despite sharing the same peak number of deadlocks and location (the two dark black nodes at the top edge of both figures).

When adding cross lanes on a much larger map, the trends of deadlocks, sim time, and fitness are very similar, i.e., after the turning points: the more cross lanes, the more deadlocks and more sim time, as well as the fitness, as shown by the solid blue line in Fig. 4b and Fig. 4a. Different from the smaller CHF transportation map, the turning point of the fitness and sim time delay from 2 cross lanes to 4 cross lanes, while the turning point of the deadlocks advances from 4 cross lanes to 3 cross lanes. The two figures suggest that the number of cross-lanes that are optimal for a field is depended on the actual size of the field. Here, for the small transportation map, 2 cross lanes are a good solution and for the large vanity map, 4 cross lanes are a good solution.

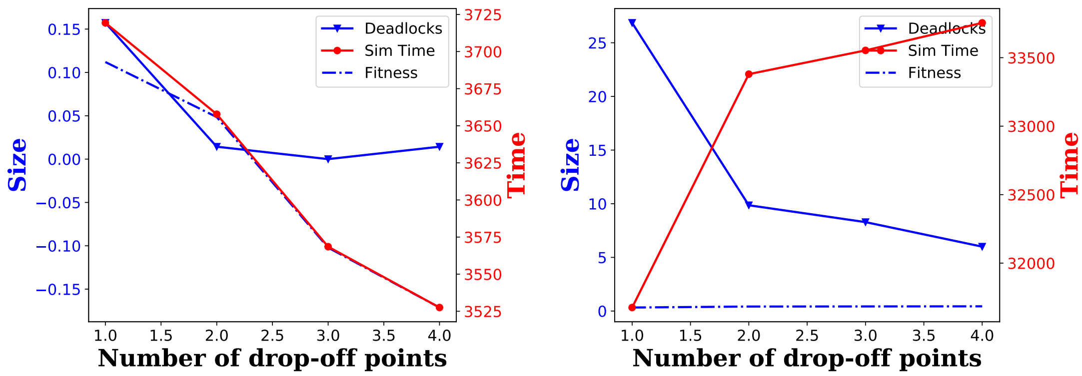
4.3 Allocating drop-off points

When using GATO for optimising the drop-off points, we make the following assumptions:

1. There are 4 pickers and 4 robots performing picking and delivering tasks in cooperation.
2. The minimum number of drop-off points is 1 and the maximum is 4.

Fig. 6a shows the overall performance of the MRS performing logistics tasks against different numbers of drop-off points with the transportation map. The rise in drop-off points leads to a decline in the number of deadlocks and simulation time, implying that more drop-off points result in better performance. However, increasing the number of drop-off points to more than four would be impractical from an operational point of view, as the fruits from these locations have to be transported to cold storage.

Fitness is calculated as the weighted sum of deadlocks and simulation time (Equation 1), which are normalised using Equation 2 and 3. A decrease in the sum does not necessarily indicate a decrease in the individual components. As shown by the deadlock curve (blue solid line), the last point increases while fitness decreases (blue dashed line). The fitness drops below 0, due to



(a) CHF transportation map

(b) CHF vanity map

Fig. 6: MRS performance when allocating drop-off points: deadlocks and sim time

the boundary sim time (maximum sim time and minimum sim time in Equation 2) we used for normalising is gained from one drop-off point, as shown by the red solid line. The fitness becomes negative when the simulation time is shorter than the minimum boundary time. The number of deadlocks is also normalised in a similar manner (Equation 3) to balance their impact on the fitness used by the GA to optimise the topological map. It should be noted that the normalised sim time and deadlocks are only used for calculating the fitness, not for plotting Fig. 6.

Fig. 6b shows a totally different trend when optimising with the vanity map. The fitness is always maintained at a steady low level, when the simulation time rises, the deadlocks decrease. With more drop-off points, the robots have more choices for delivering goals so the deadlocks decrease. However, sometimes the robot may have to travel further to reach the drop-off point that is not allocated at the best position, so the overall simulation time increases against the number of drop-off points.

$$t_{norm} = \frac{t - t_{min}}{t_{max} - t_{min}} \tag{2}$$

$$n_{norm} = \frac{n - n_{min}}{n_{max} - n_{min}} \tag{3}$$

where t is the average minimum simulation time and n is the average minimum number of deadlocks that happened in each GA iteration. min and max are the predefined boundaries according to practical experience.

4.4 Discussion

The best solutions of optimising topology are presented in Table 3. When using one cross lane for optimising CHF transportation map, the best cross lane is 42. As we can see from Fig. 3c, 42 is near the rear side (pickers pick from left to right) of the polytunnel. When using two cross lanes for optimising CHF transportation map, the best cross lanes are 39 and 45. If using five cross lanes, the best locations are 38, 26, 21, 35 and 27. Comparing the distribution of the best lanes with transportation and vanity, we see that the best solutions are scattered instead of clustered. Besides, the best lanes are more likely to be on the rear side as the density of the nodes where pickers demand logistic assistance is higher on this side.

For the drop-off points, the conclusion is more obvious. As we see from Table 3, for the CHF vanity, the best drop-off points are 31, 10, 9, 11, 12, 4, 3. Most of them are clustered in the

Table 3: Best individuals

Strategy	CHF transportation	CHF vanity
Cross lanes	[42]	[14]
	[39, 45]	[19, 12]
	[17, 5, 4]	[10, 15, 0]
	[3, 1, 15, 41]	[0, 8, 9, 18]
	[38, 26, 21, 35, 27]	[12, 19, 34, 31, 45]
Drop-off points	[2]	[31]
	[1, 7]	[10, 9]
	[3, 2, 6]	[4, 11, 10]
	[2, 0, 4, 7]	[11, 12, 10, 3]

middle area of the farm except for 3 and 4, refer to Fig. 3a. However, for the smaller map, CHF transportation, the cluster is not obvious as the distance between drop-off points are very small.

To evaluate how much the proposed solution, optimised by GATO, can improve the performance of MRS, we compared it with the performance when the algorithm is not employed. The comparison results are presented in Table 4. For the two chosen maps, Clockhouse Transportation and Vanity, we compared the simulation time and deadlocks using three scenarios respectively: default, adding cross lanes, and allocating drop-off points. The default scenario means that the GATO algorithm is not used, and picking and transporting tasks are simulated directly using DES. Meanwhile, adding cross lanes and allocating drop-off points are the two proposed strategies of GATO used to optimise the topological map for MRS logistics tasks. All scenarios were run 10 times with random seeds respectively.

The results show that GATO substantially improved performance on both small and large maps in terms of reducing task completion time and deadlocks. For the small transportation map, adding cross lanes saved about 4.3% of time and reduced about 96.2% of deadlocks. Allocating drop-off points further improved the performance by 50.2% and 99.3% respectively. Moreover, the optimisation performance was more notable on the larger Vanity map, especially in terms of simulation time; GATO achieved 49.5% and 61.0% of improvement respectively.

Table 4: Results of GATO compared with those on the unmodified topological map

Map	Scenario (s)	Time (s)	Deadlocks
Clockhouse transportation	Default	7079.1 $\pm$ 400.4	2.9 $\pm$ 3.7
	Adding cross lanes	**6777.6** $\pm$ 83.3	**0.1** $\pm$ 0.01
	Allocating drop-off points	**3527.1** $\pm$ 98.2	**0.02** $\pm$ 0.00
Clockhouse vanity	Default	86797.5 $\pm$ 625.4	115.2 $\pm$ 23.7
	Adding cross lanes	**43812.1** $\pm$ 271.3	**31.2** $\pm$ 6.7
	Allocating drop-off points	**33836.6** $\pm$ 279.5	**6.1** $\pm$ 3.3

5 Conclusions

In this paper, we propose a GA-based topological optimisation algorithm GATO for the autonomous optimising topology of agricultural fields for the efficient robotic fleet deployment of reducing congestion and improving overall transporting efficiency. Specifically, we consider the autonomous topology optimising problem related to where cross lanes should be considered and where drop-off points should be allocated. The cross lanes improve the efficiency by allowing the robot to change rows in the middle of the polytunnels instead of travelling from the row head to the row rear for changing rows. The optimised drop-off points shorten the total delivery distance that the robots need to travel. Crucially, we discretise the searching space of GA by labelling the position of cross lanes and drop-off points as finite integer numbers. The GA generates the population from the finite integers and evaluates the fitness of the solution with DES. Then we get the best individual among all the individuals which is also a good quality solution for optimising the topological map. This method allows the GA to find a high-quality or even optimum solution when using the complete population in a computationally efficient way.

Finally, we empirically evaluate our GATO in discrete event simulations with a small farm map and a commercial farm map. Compared with manual topological optimisation in our previous work [4], the proposed autonomous algorithm explores more potential solutions and has the ability to find better solutions or even optimum solutions. This paper further expands the field of autonomous topological optimisation, from optimising of base nodes in our previous work [22] to the proposed method of optimising the cross lanes and drop-off points. The GATO algorithm has the ability to significantly improve facilities by designing new ones or modifying existing ones to efficiently accommodate autonomous robots, resulting in overall process improvements.

Acknowledgements

This work has been funded by InnovateUK grants 51367 ”Robot Highways” and 10041179 ”Agri-OpenCore”.

References

1. Guang-Zhong Yang, Jim Bellingham, Pierre E Dupont, Peer Fischer, Luciano Floridi, Robert Full, Neil Jacobstein, Vijay Kumar, Marcia McNutt, Robert Merrifield, Bradley J Nelson, Brian Scassellati, Mariarosaria Taddeo, Russell Taylor, Manuela Veloso, Zhong Lin Wang, and Robert Wood. 2018. The grand challenges of Science Robotics. Sci. Robot. 3, 14 (2018), eaar7650.
2. David Christian Rose, Jessica Lyon, Auvikki de Boon, Marc Hanheide, and Simon Pearson. 2021. Responsible development of autonomous robotics in agriculture. Nat. Food 2, 5 (2021), 306–309.
3. Tom Duckett, Simon Pearson, Simon Blackmore, Bruce Grieve, Wen-Hua Chen, Grzegorz Cielniak, Jason Cleaversmith, Jian Dai, Steve Davis, Charles Fox, and Others. 2018. Agricultural robotics: the future of robotic agriculture. arXiv Prepr. arXiv1806.06762 (2018).
4. Gautham Das, Grzegorz Cielniak, Francesco Del Duchetto, Zuyuan Zhu, James Heselden, Johann Dichtl, Marc Hanheide, Simon Pearson, Jaime Pulido Fentanes, Adam Binch, Michael Hutchinson, and Pal From. 2022. A Unified Topological Representation for Robotic Fleets in Agricultural Applications. Manuscript Submitt. Publ. (2022).
5. Ayoub Insa Corréa, André Langevin, and Louis-Martin Rousseau. 2007. Scheduling and routing of automated guided vehicles: A hybrid approach. Comput. & Oper. Res. 34, 6 (2007), 1688–1707.
6. Tatsushi Nishi, Yuichiro Hiranaka, and Ignacio E Grossmann. 2011. A bilevel decomposition algorithm for simultaneous production scheduling and conflictfree routing for automated guided vehicles. Comput. & Oper. Res. 38, 5 (2011), 876–888.
7. Valerio Digani, M Ani Hsieh, Lorenzo Sabattini, and Cristian Secchi. 2019. Coordination of multiple AGVs: a quadratic optimization method. Auton. Robots 43, 3 (2019), 539–555.
8. Brian P. Gerkey and Maja J. Matarić. 2004. A Formal Analysis and Taxonomy of Task Allocation in Multi-Robot Systems. The International Journal of Robotics Research 23, 9 (2004), 939–954.
9. Alessandro Farinelli, Nicolo’ Boscolo, Elena Zanotto, and Enrico Pagello. 2017. Advanced approaches for multi-robot coordination in logistic scenarios. Robotics and Autonomous Systems 90 (2017), 34–44.
10. Beatriz A. Asfora, Jacopo Banfi, and Mark Campbell. 2020. Mixed-Integer Linear Programming Models for Multi-Robot Non-Adversarial Search. IEEE Robotics and Automation Letters 5, 4 (2020), 6805–6812.

11. Shaurya Shriyam and Satyandra K. Gupta. 2020. Incorporation of Contingency Tasks in Task Allocation for Multirobot Teams. IEEE Transactions on Automation Science and Engineering 17, 2 (2020), 809–822.
12. Jianping Wang, Yuesheng Gu, and Xiaomin Li. 2012. Multi-robot task allocation based on ant colony algorithm. J. Comput. 7, 9 (2012), 2160–2167.
13. Lantao Liu and Dylan A Shell. 2013. Optimal Market-based Multi-Robot Task Allocation via Strategic Pricing.. In Robotics: Science and Systems, Vol. 9. 33–40.
14. P.Th. Zacharia and N.A. Aspragathos. 2005. Optimal robot task scheduling based on genetic algorithms. Robotics and Computer-Integrated Manufacturing 21, 1 (2005), 67–79.
15. Abderraouf Maoudj, Brahim Bouzouia, Abdelfetah Hentout, Ahmed Kouider, and Redouane Toumi. 2016. Distributed multi-agent approach based on priority rules and genetic algorithm for tasks scheduling in multi-robot cells. In IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society. IEEE, 692–697.
16. Hanfu Wang, Weidong Chen, and Jingchuan Wang. 2020. Coupled task scheduling for heterogeneous multi-robot system of two robot types performing complexschedule order fulfillment tasks. Robotics and Autonomous Systems 131 (2020), 103560.
17. O Booij, B Terwijn, Z Zivkovic, and B Krose. 2007. Navigation using an appearance based topological map. In Proc. 2007 IEEE Int. Conf. Robot. Autom. 3927–3932.
18. Štimec, Aleš and Jogan, Matjaž and Leonardis, Aleš. 2008. Unsupervised learning of a hierarcht of topological maps using omnidirectional images. Int. J. Pattern Recognit. Artif. Intell. 22, 04 (2008), 639–665.
19. A. Binch, G.P. Das, J.P. Fentanes, and M. Hanheide. 2020. Context Dependant Iterative Parameter Optimisation for Robust Robot Navigation. In Proc. 2020 IEEE Int. Conf. Robot. Autom.
20. Varga, András. 2001. Discrete event simulation system. In Proc. of the European Simulation Multiconference (ESM'2001). 1–7
21. Gautham P. Das, Grzegorz Cielniak, Johan From, and Marc Hanheide. 2018. Discrete Event Simulations for Scalability Analysis of Robotic In-Field Logistics in Agriculture – A Case Study. In ICRA 2018 Work. Robot. Vis. Action Agric. Brisbane.
22. Zuyuan Zhu, Gautham P. Das, and Marc Hanheide. 2023. Autonomous Topological Optimisation for Multi-robot Systems in Logistics. In The 38th ACM/SIGAPP Symposium on Applied Computing (SAC '23). Tallinn, Estonia.
23. Team SimPy. 02.02.2023. SimPy: Discrete event simulation for Python. https://simpy.readthedocs.io/