Curtin Business School

Evolutionary Algorithms to Solve Agricultural Routing Planning

Amalia Utamima

This thesis is presented for the degree of
Doctor of Philosophy
of
Curtin University

December 2020

Declaration

To the best of my knowledge and belief, this doctoral thesis contains no material previously published by any other person except where due acknowledgement has been made.

This thesis contains no material that has been accepted for the award of any other degree or diploma in any university.

I warrant that I have obtained, where necessary, permission from the copyright owners to use any third party copyright material reproduced in the thesis (questionnaires, artwork, unpublished letters etc.), or to use any of my own published work such as journal articles in which the copyright is held by another party such as the publisher.

Amalia Utamima
13 December 2020

Abstract

This doctoral thesis is presented as a hybrid thesis containing several chapters and sections from published papers by the author.

The optimisation of the agricultural process has gained importance in recent years; that is, to increase harvested yield and to reduce the cost and time required for agricultural operations in the domain of field logistics. Route planning, a part of agricultural operations, is essential in field logistics to minimise the cost of operating agricultural machinery. Agricultural Routing Planning (ARP) addresses the route planning of machines to undertake several agricultural tasks on the farmer field.

This research aims to review the previous research on ARP, to formulate an extension of the ARP and to develop effective Evolutionary Algorithms that can be competitively applied to ARP. The outcomes of this research are several Evolutionary Algorithms that can generate effective route planning for the machines. The outcomes will impact on the research community with the development of new algorithms as well as the dissemination of findings. This study is significant as it is expected to improve the management of agricultural machinery, to minimise the total cost and the settling time for completing field operations, and to produce routing plans.

ARP belongs to the class of NP-hard problems (NP is defined as nondeterministic polynomial time); therefore, an exact optimisation is not feasible concerning run time constraints and so Evolutionary Algorithms are utilised. In addition to defining an extended ARP, this study presents several Evolutionary Algorithms to address ARP. Several metaheuristic algorithms from the literature are used as a benchmark against the proposed Evolutionary Algorithms.

There are two main stages to this thesis. In the first stage, the author studied the previous ARP datasets, created a general mathematical model to represent the datasets and developed a new hybrid Evolutionary Algorithm to solve the model. To obtain datasets for the agricultural routing problem, the author gathered data from previous publications describing different fields. In the second stage, the ARP was extended so as to consider several constraints at the same time to make it closer to the real case. The motivation for an extended ARP is because despite the variations of ARP, the capacitated application with multiple machines and multiple fields has not been covered extensively. Hence, this study extended the ARP to include multiple crop fields and multiple machines with different capacities and provided the mathematical model to represent the problem.

Keywords: agriculture, Agricultural Routing Planning, Evolutionary Algorithm, field logistics

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ACO	Ant Colony Optimisation
ARP	Agricultural Routing Planning
CPP	Coverage Path Planning
DSS	Decision Support Systems
EDA	Estimation Distribution Algorithm
EEDA	Evolutionary Estimation of Distribution Algorithm
EHNS	Evolutionary Hybrid Neighbourhood Search
ENDA	Evolutionary Surrounding Search
EvoLovebird	Evolutionary Lovebird Algorithm
GA	Genetic Algorithm
NP	Nondeterministic Polynomial Time
PSO	Particle Swarm Optimisation
SA	Simulated Annealing
TS	Tabu Search
TSP	Travelling Salesman Problem
VNS	Variable Neighbourhood Search
VRP	Vehicle Routing Problem

Glossary of Terms

Term	Definition
Agricultural operations	The use of land to produce an agricultural commodity for commercial purposes, including but not limited to, cultivation and tillage of the soil, application of pesticides and fertilisers, irrigation, production, pruning, growing, harvesting and processing of any agricultural commodity.
Arable farming	The growing crops in fields, which have usually been ploughed before planting.
cBest/currentBest	The current best solution found in a generation/iteration in the algorithm.
Candidate solution	The possible solutions generated by the algorithm.
Field(s)	Farmer field. The fields operated by the farmer and considered in the optimisation
Field operation	Agricultural operation in the farmer field.
gBest/globalBest	The best solution found so far through a generation/iteration in the algorithm.
Headland area	Crop-free area is used by agricultural vehicles/machines to execute manoeuvres to change direction and transfer to another track.
Headland distance	The distance travelled by the machine(s) in the headland area.
Incapacitated machine	A machine without capacity constraint.
Machine(s)	A vehicle used to perform an agricultural operation in the farmer field.
Manoeuvres/Turn	The machine's turn in the headland area.
Non-working distance	The distance travelled by the machine(s) when not performing an agricultural activity (while performing a manoeuvre, or travelling from/to the depot, or to another field).
Track	Lane in a field that may contain crops.

List of Publications

FoR Code: 0806

No.	Title	Type, Publisher	Year, Vol., Page	Rank	Authors
1	Optimisation of agricultural routing planning in field logistics with Evolutionary Hybrid Neighbourhood Search	**Journal,** *Biosystems Engineering,* Elsevier	2019, vol. 184, pp. 166–80	**Scimago Q1, ERA rank A**	**Amalia Utamima**, Torsten Reiners, Amir H. Ansaripoor
2	Evolutionary estimation of distribution algorithm for agricultural routing planning in field logistics	**Conference,** Procedia Computer Science, Elsevier	2019, vol. 161, pp. 560–67	**B**	**Amalia Utamima**, Torsten Reiners, Amir H. Ansaripoor
3	Decision making for farmers : A case study of agricultural routing planning	Australasian **Conference** of Information System	2019	**A**	**Amalia Utamima**, Torsten Reiners, Amir H. Ansaripoor
4	Automation in agriculture: A case study of route planning using an evolutionary lovebird algorithm (*Winner Best Presentation*)	International **Conference** Proceeding Series, ACM Digital Library	2020	Scopus indexed	**Amalia Utamima**, Torsten Reiners, Amir H. Ansaripoor
5	The agricultural routing planning in field logistics (in Contemporary Approaches Strategy of Applied Logistics)	**Book Chapter,** IGI Global	2018, pp. 261–83	-	**Amalia Utamima**, Torsten Reiners, Amir H. Ansaripoor, Hasan Seyyedhasani

Acknowledgements

This doctoral thesis would not have been completed without the excellent people who have contributed their valuable guidance and assistance. It is a pleasure to express my sincerest gratitude to all of them in my humble confession.

First, I would like to state my highest gratitude to my supervisors, Dr Torsten Reiners and Dr Amir Ansaripoor, who have supported and guided me throughout my thesis and the completion of my doctoral study with their advice, knowledge, patience and sincerity. They inspired me to always have great passion for my learning and research.

My best regards also goes to the LPDP (Indonesian Endowment Fund for Education) of Ministry of Finance, Republic of Indonesia, for supporting my tuition fee and living allowance in Australia. I also want to give thanks to all the lecturers and staff in the former School of Information Systems and Curtin University, who gave assistance during my study.

Finally, I dedicate this PhD thesis to my husband, Hudan Studiawan, my beloved daughter, Athiyya Faiqa, and to my parents, Ibu Qudsiati and Bapak Budi Musana, for their continuous encouragement throughout my doctoral study.

Chapter 1. Introduction

1.1 Background

Developments addressing agricultural field operations and management are generally associated with improved seeds, the use of chemicals, increased resistance to diseases and pests, and water utilisation (Jones et al., 2017). These factors theoretically increase the productivity of the cultivated areas; however, they do not resolve the deployment of machinery and humans to perform field operations or to transport the produce from the field to storage.

The agricultural sector also confronts a growing number of logistical challenges. The optimisation of resource utilisation (fuel, fertiliser and water) and the time needed to manage and perform field operations on increasing areas of farmland have become an integral part of farm operations, motivated by climate change and the demands of sustainable operations. Changes in environmental conditions impact the timeline for applying fertilisers and water or harvesting crops. At the same time, to reduce the expense and usage of carbon dioxide pollutants requires the optimisation of vehicle paths and subsequent use by reducing the driven distance (Al-Zamil & Saudagar, 2018).

A number of benefits can be derived from optimising the use of humans and machines, such as a reduction in time and thus an increase in profits. The cost of agricultural machinery is reduced, fewer seasonal workers need to be hired and less time is required for field operations. This last factor is critical as the size of farms increases, yet the timeframe for performing field operations remains the same.

Route planning in field logistics is one of the main tasks in the management of agricultural machinery (Bochtis et al., 2014). The optimised operation involves Agricultural Routing Planning (ARP), which is used to manage the logistics of agricultural machines in relation to their field movements and other farm locations.

ARP studies typically do not provide enough in-depth analysis to generalise research and develop algorithms across multiple cases. In general, the lack of benchmarking across a collection of datasets limits the comparability of algorithms

and publicity of the transfer of findings to other instances. Thus, the generalisation of ARP and the collection of datasets from previous research is essential.

In ARP, agricultural machines are restricted by several constraints, such as their capacity to hold the harvest crop or goods to be used in the field, their turning radius, the width of tracks and fuel consumption. Some researchers have developed initial models to characterise ARP, but these models are generally limited to the simple consideration of constraints (Bochtis & Vougioukas, 2008; Seyyedhasani & Dvorak, 2017). Furthermore, despite the variations of ARP, the capacitated application with multiple machines and multiple fields has not been covered extensively and the focus has been on one field. Hence, the study in ARP for multiple fields and multiple machines represents a research gap.

ARP with multiple fields and multiple machines is closer to real cases where farmers generally have multiple connected fields and operate more than one machine on those fields. An inclusion of those properties in the ARP problem as well as algorithms to produce an effective routing plan will help farmers to save driving distance, time, and cost of doing the field operations.

A major challenge of route planning in the field is that the problem is NP-hard (Oksanen & Visala, 2009). Therefore, getting an optimal solution is computationally intractable. Routing planning problems are commonly faced with this obstacle, which has forced the development of algorithms that can deliver good (near-optimal) solutions in a reasonable amount of time.

1.2 Objectives

The principal objective of this research is to extend the ARP problem and to develop Evolutionary Algorithms to find near-optimal solutions. This study views ARP from an operational research perspective. More specifically, the research is intended to:

1. Collect ARP instances from previous research as benchmark data and identify a general mathematical model to represent them.

2. Define the extended ARP that considers multiple crop fields and multiple machines with different capacities and provide a mathematical model to represent the problem.
3. Generate the instances of the extended ARP.
4. Develop and configure the Evolutionary Algorithms for ARP
5. Compare the proposed Evolutionary Algorithms and other algorithms from the previous literature.

1.3 Contribution and Significance

The optimisation of route planning to harvest crop fields is important in order to minimise the total cost of operating agricultural machines. The first contribution of this study is made by defining an extended ARP that deals with multiple fields, tracks, capacities and turning manoeuvres. The second contribution is the development of mathematical formulations to represent the general dataset and extended ARP. Furthermore, recent algorithms are not able to solve the given problems in the literature in terms of solution quality and time. Thus, the third contribution of this research is the exploration of new algorithms to address the extended version of ARP. These new hybrid algorithms are benchmarked against existing algorithms in the literature.

ARP optimisation aims to minimise the total distance driven to manage fields, reduce the number of machines and synchronise multiple machines. By solving the proposed ARP with the suggested algorithm, this study is significant as it is expected to improve the management of the machinery fleet, to minimise the total cost and the settling time for doing field operations in multiple crop fields, and to produce better routing plans. Furthermore, the outcomes of this study will impact on the research community with the development of new algorithms as well as the dissemination of findings.

1.4 Research Phase

During the doctoral study, a literature review was conducted to identify the ARP problem, notice the main algorithms on ARP and find gaps in the literature. There are two main phases to this study:

I. Data Collection, Analysis and Algorithm Development

In this first phase, the author collected the datasets from previous ARP research and analysed them. The author then developed an algorithm to solve the data. The main output of this phase is detailed in journal manuscript one (*Biosystems Engineering*).

II. Extension to the Problem and Algorithm Development

In this second phase, the author formalised the extended ARP and built an algorithm to deal with the problem. The main output of this stage is detailed in journal manuscript two (*Annals of Operations Research*).

After the main phases are completed, the author conducted a comparative study of the algorithms built in this study. The author also observed the possible future implementation of ARP in the form of DSS and other field operations.

1.5 Thesis Outline

This doctoral thesis is presented as a hybrid thesis containing several chapters and sections from published papers by the author. The thesis is organised as follows.

- Chapter 2. Literature Review

This chapter contains a review of the previous studies related to ARP. The details of the research gaps are also indicated in this chapter.

- Chapter 3. Overview of the General Problem

This chapter gives an overview of the ARP problem to be solved in this study. The representation of the problem and the description of the machine manoeuvres are listed in this chapter.

❖ Chapter 4. Optimisation of Dataset in ARP

This chapter observes the optimisation of the previous dataset in ARP, which is based on the author's journal publication 'Optimisation of agricultural routing planning in field logistics with Evolutionary Hybrid Neighbourhood Search', *Biosystems Engineering*, 2019, vol. 184, pp. 166–80.

❖ Chapter 5. Extensions to the ARP

This chapter explores the extension of ARP that considers multiple fields and multiple machines with different capacities. The chapter is based on the author's journal submission to *Annals of Operations Research*, second review.

❖ Chapter 6. Evolutionary Algorithms for ARP

This chapter lists the Evolutionary Algorithms built during this research. The comparison of the best three algorithms is also provided.

❖ Chapter 7. Practical Application

This chapter describes the possible practical application (future implementation) of ARP, based on the author's conference publications (Entry 4 and 5 in the table on Page xv).

❖ Chapter 8. Conclusion

This chapter lists the limitations of the research and concludes the whole study.

Chapter 2.Literature Review

This chapter reviews previous studies related to ARP as well as optimisation, with a focus on Evolutionary Algorithms. Section 2.1 presents an overview of the Vehicle Routing Problem (VRP). Section 2.2 explores the optimisation studies in agriculture, including VRP. Section 2.3 investigates the previous studies of ARP and provides an overview of their contributions in the literature. Section 2.4 describes a variety of algorithms and methods used in previous studies. Finally, Section 2.5 states the research gaps identified in the current literature.

2.1 Vehicle Routing Problem

In logistics, various attributes (such as the high variability of demand, constraints in delivery times, uneven distribution over large areas, and the dimension and complexity of orders) increase the planning and operation time and cost, and significantly impact on the efficient and effective flow of products as well as on customer satisfaction. The operations research literature labels the problem concerning the distribution of goods between depot and customers, among others, as the Vehicle Routing Problem (VRP), with different constraints (Cordeau et al., 2002; Fisher, 1995).

VRP defines the routes with the minimum cost from one depot to a set of geographically dispersed locations, such as cities, shops and warehouses. The VRP solution aims for the selection of a series of routes, each carried out by a single vehicle. The vehicle(s) begins and finishes at its depot. Thus, all customer requirements and all operational constraints are met while minimising the global transport costs. The customer requirements refer to transport quantity, type of goods and dimensions. The transport of goods is done by vehicles whose composition and size can be fixed or defined in accordance with the requirements of the customers. The operational constraints depend on the nature of the goods being transported and the characteristics of the vehicles (Toth & Vigo, 2014). Routes must be produced in a specific direction so each location is visited once by exactly one vehicle (Vidal et al., 2019).

The road network, used for the transport of goods to the depot and customer locations, is generally described by a graph. The arcs and the vertices in the graph represent the road sections and the road junctions, respectively. Every arc has an associated value, which can reflect either the length or the travel time during which the arc is traversed (Toth & Vigo, 2014).

Figure 1 illustrates the VRP with three vehicles, 15 vertices (locations), 28 arcs (paths) and a depot. The lines (grey, blue, green and pink) indicate the available arc that connects the vertices, while the number beside the arc indicates the distance to traverse that arc. Several grey arcs show the available paths not chosen by the vehicles. The shortest paths of the first, second and third vehicles to cover all vertices are coloured in blue, green and pink, respectively. The routes of each vehicle begin and end at the depot.

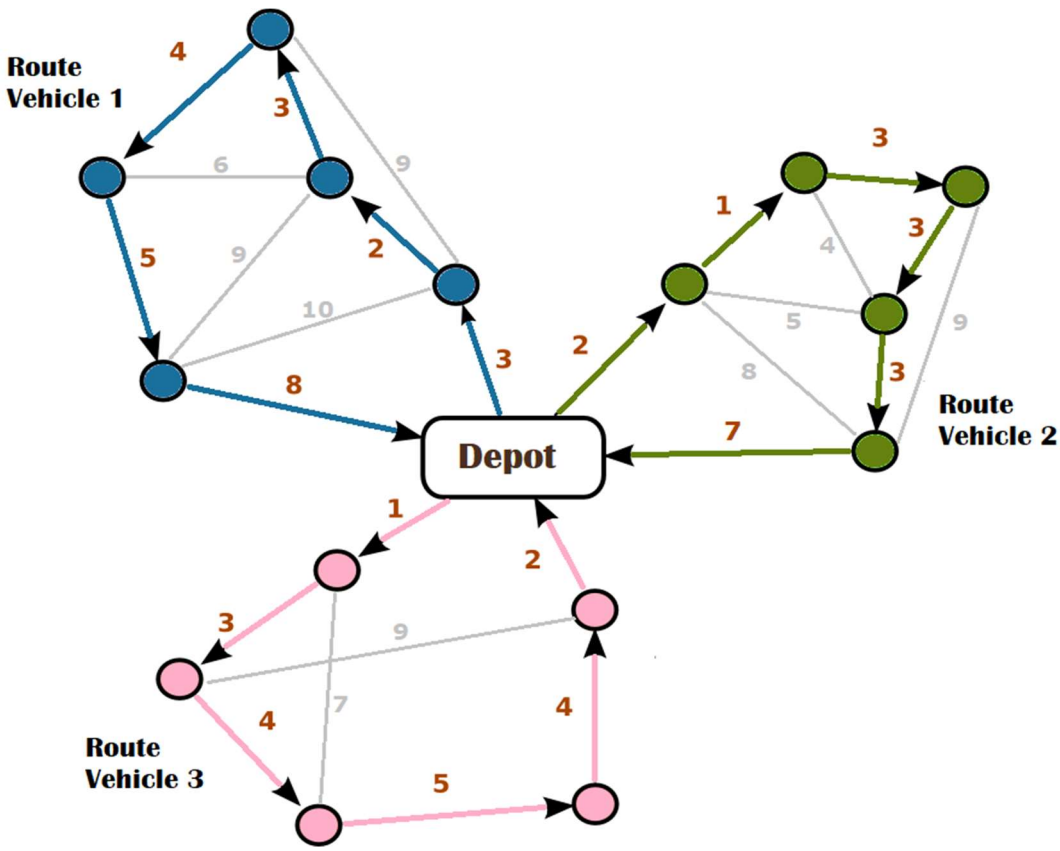


Figure 1. Illustration of the Vehicle Routing Problem

In VRP, the cost is calculated by sum the shortest path(s) that starts from a vertex and arrives at another vertex in the road graph. The objectives considered in VRP include:

- Minimising global transport costs or the global distance travelled by the vehicles.
- Minimising the global travel time.
- Minimising the costs associated with the vehicles.
- Minimising the number of vehicles (or drivers) needed to service all the customers.
- Balancing the travel time and vehicle load (Toth & Vigo, 2002).

VRP is a generalisation of the former problem named the Travelling Salesman Problem (TSP, Bochtis & Sørensen, 2009). In TSP, one salesperson has to visit all the customers exactly once and with the shortest path. However, TSP does not consider capacity constraints whereas most VRP do consider capacity constraints such that the cumulative demands of all locations on a given route must not exceed the capacity of the vehicles (Toth & Vigo, 2002).

The additional constraints result in instances of VRP (Bochtis & Sørensen, 2010). For example, the limitation of the maximum length of the path (distance constrained VRP), every location should be serviced within a specific time constraint (VRP with time windows), and the stochastics demand of the customers (VRP with stochastics demand). Other additions are VRP with multiple depots and VRP with different capacity of the vehicles (Koç et al., 2016).

VRP is one of the most challenging optimisation problems, and over the years, a large number of different methods have been developed. VRP belongs to the NP-hard problem, which makes it too difficult to solve with a conventional method. Thus, heuristics or metaheuristics approaches are needed to solve VRP (Archetti & Speranza, 2014; Cordeau et al., 2002; Pereira & Tavares, 2009).

2.2 Coverage Path Planning in Field Logistics

VRP in field logistics is also known as coverage planning (Bochtis & Sørensen, 2009). The coverage planning problem is labelled as Coverage Path Planning (CPP) (Oksanen & Visala, 2009). CPP aims to minimise the distance of a route that passes over all points of a region (Galceran & Carreras, 2013). In

agriculture, CPP has been used in different applications. The distinction of the applications is found in the operations on the field, the field configuration or the objectives. CPP has been explored in aerial coverage optimisation (Barrientos et al., 2011; Valente et al., 2013), crops harvesting (Amiama et al., 2015; He et al., 2018; Sethanan & Neungmatcha, 2016) and biomass collection (Akhtari et al., 2014; Gracia et al., 2014). Several CPP publications focus on the generation of paths or tracks inside fields (Bochtis et al., 2013; Graf Plessen & Bemporad, 2017; Jin & Tang, 2010; Oksanen & Visala, 2009)

Table 1 lists examples of research on CPP in agriculture detailing their features and specifications. The first variation is the CPP for aerial mini-robots. In this variation, both Valente et al. (2013) and Barrientos et al. (2011) optimised CPP in vineyard parcels for mini-robots. Valente et al. (2013) focused on the implementation of their metaheuristic approach to the problem, while Barrientos et al. (2011) focused on the practical experimentation in the field using a heuristics method. Barrientos et al. (2011) used two steps in their experiments: 1) data collection using the aerial robots scanning the field; and 2) the coverage planning of the field using the collected data. Valente et al. (2013) proposed a metaheuristic algorithm that could improve the CPP solution proposed by Barrientos et al. (2011).

The second variation is the CPP for harvesting different crops. Amiama et al. (2015) developed a decision support system for identifying the routes that provide reduced travelling distances for maize silage harvest operations. Sethanan and Neungmatcha (2016) explored the harvesting sugarcane field operation to minimise the travelling distance in one harvesting period. The authors generated seven samples following realistic scenarios of a general sugarcane field in Thailand. Kittilertpaisan and Pathumnakul (2017) extended the research on CPP by considering three harvesting periods in sugarcane fields. The authors claimed that their work could benefit the integration of a harvester routing design and plan the cultivation of a new crop. Both He et al. (2018) and He and Li (2019) conducted the optimisation of time in harvesting problems in the wheat field. He et al. (2018) examined a case with moist farmland, while He and Li (2019) considered the multi-trip VRP.

Table 1. Examples of CPP in agriculture

No.	Variations	Examples of Research	Specifications
1	Aerial/mini-robots	Barrientos et al. (2011)	Propose a strategy to cover a crop field utilising aerial robots for data collection
		Valente et al. (2013)	Propose a new approach to improve coverage trajectories
2	Crop harvesting	Amiama et al. (2015)	A decision support system to find the shortest distance in harvesting maize silage fields
		Sethanan and Neungmatcha, (2016); Kittilertpaisan and Pathumnakul (2017)	Minimise the travelling distance in the sugarcane field
		He et al., (2018); He and Li (2019)	Optimise the time in harvesting wheat fields
3	Biomass collection	Akhtari et al. (2014)	Minimise the cost of delivering forest biomass to the heating plant gate
		Gracia et al. (2014)	Minimise the distance in biomass collection
4	Field-track generation	Bochtis and Vougioukas (2008), Bochtis et al. (2013)	Develop B-patterns to generate field tracks
		Oksanen and Visala (2009)	Develop a greedy algorithm to determine the best driving direction and the subfields
		Jin and Tang (2010)	Use geometric model and divide-and-conquer strategy for CPP
		Graf Plessen and Bemporad (2017)	Generate reference trajectory planning for autonomous machine

The third variation of CPP is the biomass collection. Gracia et al. (2014) used the biomass harvesting and collection problem as a case study to improve

operational efficiency. The authors modelled the problem as a VRP and used multiple vehicles with capacity constraints. Meanwhile, Akhtari et al. (2014) developed a model to ensure that the total demand for biomass was met from different supply areas. The authors minimise the cost of delivering forest biomass to the heating plant and assess the optimum monthly stream of biomass to the facility.

Some studies focused on generating the tracks inside the agricultural field, as detailed in the fourth variation. In these studies, the field is assumed not to have any developed tracks, hence it can be traversed from any driving direction. The authors proposed different approaches to generate the tracks to provide an efficient path to be traversed by vehicles. Oksanen and Visala (2009) dealt with the shape of agricultural fields. They used two kinds of greedy algorithms to generate the tracks by determining the best driving direction and the selection of subfields. Jin and Tang (2010) analysed a similar problem and used a geometric model with the divide-and-conquer strategy for arable farming. The authors demonstrated that the proposed method could produce better generated tracks compared to the farmer's solution and previous research. They claimed the generated tracks can reduce the distance the vehicles cover in turning manoeuvres.

Bochtis and Vougioukas (2008) and Bochtis et al. (2013) proposed the use of B-patterns for coverage planning. The authors used B-patterns to simulate the operational performance for different combinations of fields and characteristics of machines. Several years later, Graf Plessen and Bemporad (2017) analysed the case to control the agricultural machine by generating reference trajectory planning. The smooth reference trajectory is used to guide the autonomous and slow-moving agriculture machine.

2.3 Agricultural Routing Planning

Several problems of CPP in agriculture consider fields with tracks and the manoeuvres performed by the vehicles. CPP within this scope can be understood as Agricultural Routing Planning (ARP). In ARP, each field consists of several tracks, as illustrated in Figure 2.

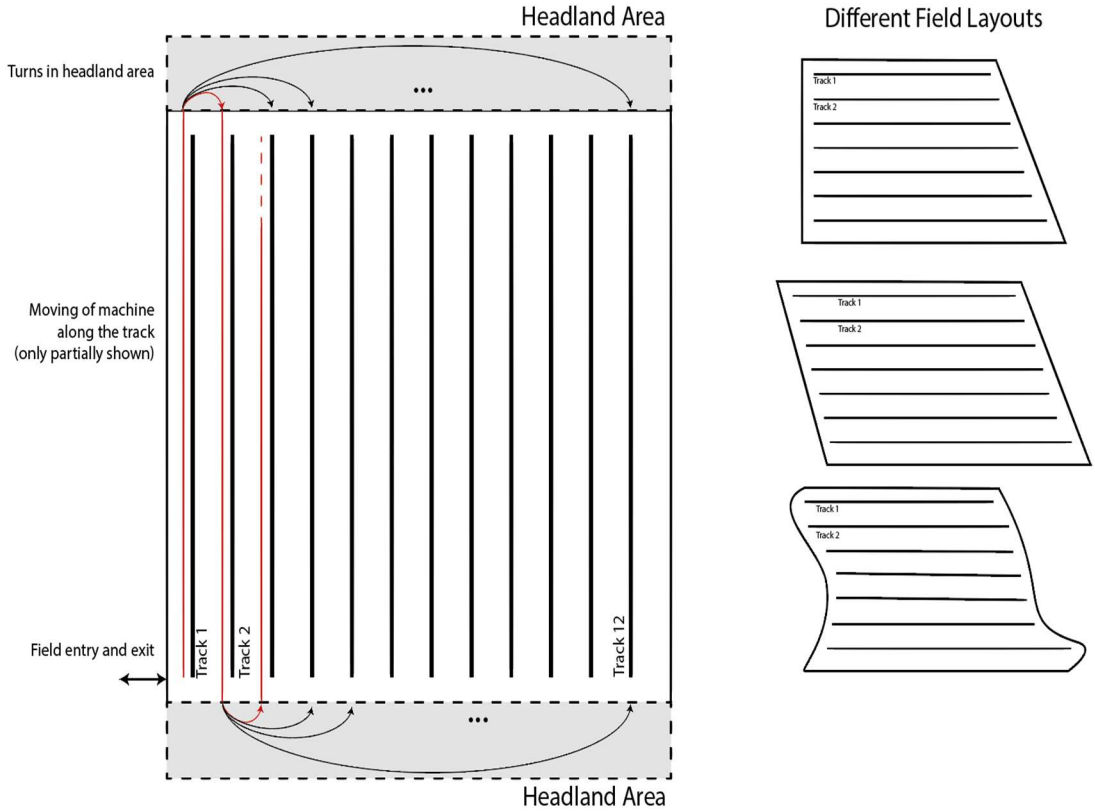


Figure 2. Detailed illustration of a field in ARP (Utamima et al., 2019c)

In Figure 2, the tracks have a headland area at each end to allow agricultural vehicles to execute manoeuvres to change direction and to transfer to another track (Bochtis & Vougioukas, 2008; Utamima et al., 2019c). Each track in the field needs to be visited exactly once by the agricultural vehicle. A field can have different shapes or layouts; see the illustration on the right side of Figure 2.

ARP is about the minimisation of the distance travelled by agricultural vehicles (called machines hereafter) to do field operations in all tracks in the field (Utamima et al., 2018). This problem has been adjusted concerning different objectives (e.g. to optimise time, use of multiple machines or minimise input costs) and constraints (e.g., machine capacity, various fields or barriers).

The following sections discuss the different focuses of ARP: 1) the minimisation of non-working distance; 2) the minimisation of time; and 3) the applications to different cases.

2.3.1 Minimisation of Non-Working Distance

Bochtis and Vougioukas (2008) first proposed the optimisation of the headland distance. The goal was to reduce the distance travelled in the headland area. This distance is called the 'non-working distance' because the agricultural machine does not perform an agricultural activity while performing a turning manoeuvre. The concept of non-working distance can be used both in a real field or hypothetical field. Hameed et al. (2011) and Spekken and Bruin (2013) used real fields, while Seyyedhasani and Dvorak (2017), Bakhtiari et al. (2013) and Conesa-Muñoz et al. (2016b) used a generated hypothetical field. Most studies in ARP use the same concept of headland distance or non-working distance, as introduced by Bochtis and Vougioukas (2008).

Table 2 shows examples of studies to minimise the non-working distance in the field. The first variation is the implementation of the B-pattern by Bochtis et al. (2013) before optimising the non-working distance. B-pattern refers to 'algorithmically-computed sequences of fieldwork tracks for agricultural machines completely covering an area that do not follow any predetermined standard motif (as in the conventional fieldwork patterns), but are a result of an optimisation process under one or more selected criteria' (Bochtis et al., 2013). Hameed et al. (2011) use B-patterns for generating the tracks and then use a metaheuristic algorithm to minimise the non-working distance of the routing of agricultural machines in the field. The results demonstrate that both studies can produce more efficient routes compared to the conventional fieldwork pattern. Bakhtiari et al. (2013) adapted B-patterns and optimised (using a metaheuristic method) an extension of ARP where the harvester unloads at a stationary facility located outside the field area. Both Hameed et al. (2011) and Bakhtiari et al. (2013) observed the operation in one field and one machine without capacity constraints using metaheuristic algorithms.

Table 2. Examples of research to minimise the non-working distance

Variations	Examples of Research	Important Findings
B-pattern implementation	Hameed et al. (2011)	Generate the driving angle and optimise the non-working distance and compare it with a conventional approach
	Bakhtiari et al. (2013)	Optimise path planning of a case where the harvester unloads at a stationary facility located outside the field area
Obstacles inside the field	Hameed et al. (2013)	Navigate agricultural robots in agricultural operations
	Zhou et al. (2014)	Use TSP concept in the real field to represent the problem
	Graf Plessen (2019)	Observe full and partial field CPP

The second variations are due to the presence of obstacles inside a field. Zhou et al. (2014), Hameed et al. (2013) and Graf Plessen (2019) considered the presence of obstacles in several shapes of the agricultural field with one machine. Zhou et al. (2014) introduced the ARP with multiple barriers and used the TSP concept to represent the problem. The authors used both open field and real field with multiple obstacles and minimised the non-working distance of the machines. Similarly, Hameed et al. (2013) optimised the non-working distance for agricultural operations involving obstacles for the navigation of agricultural robots. Graf Plessen (2019) minimised the non-working distance in full and partial field coverage in irregular-shaped fields with obstacles. The partial field coverage targets in spraying and fertilising applications with larger working widths for in-field operating vehicles.

2.3.2 Minimisation of Time

Beside optimisation in the non-working distance, several studies focused on the minimisation of time to complete field operations (Seyyedhasani & Dvorak, 2017, 2018a; Spekken & de Bruin, 2013). The time minimisation also reflects on secondary goals, as minimising distance implies a cost saving for fuel while minimising the completion time will also reduce the working hours required for the operators.

Table 3 lists the research to minimise the time required to perform the operation inside fields. Jensen et al. (2012) developed an ARP that involves the generation of optimal in-field and inter-field paths to be followed by a supporting unit (transport vehicle) cooperating with a primary unit (harvester vehicle). Both minimisation of time and non-working distance are considered in the research.

Table 3. Examples of studies to minimise the time to perform field operations

Examples of Research	Specifications
Jensen et al. (2012)	Minimisation of time and non-working distance in-field and inter-field paths of the fields
Seyyedhasani and Dvorak (2018a, 2018b, 2017)	Minimise the time of the last vehicle to finish its field operation
Spekken and de Bruin (2013)	Minimise servicing time of the vehicles

Seyyedhasani and Dvorak (2017) minimised the total time for the last vehicle to finish the field operation. The authors used the multiple VRP concepts without capacity constraint in their problem and solved a hypothetical rectangular field and an irregular-shaped field. Seyyedhasani and Dvorak (2018a) extended the idea of their previous work to become a dynamic multiple depot VRP. The simulation in the authors' research illustrates the possibility to update field routes (dynamically) for the vehicles, while the start and stop depots can be at different locations. Meanwhile, Seyyedhasani and Dvorak (2018b) used the concept from their

previous work (Seyyedhasani & Dvorak, 2017) to reduce fieldwork time with experiments in the real field with real tractor. The authors claimed the computer model could accurately represent field working times of different routings.

Spekken and Bruin (2013) also reduced the fieldwork time by minimising the servicing time of the routing on agricultural machines when doing the non-working procedures on six fields having different shapes. Seyyedhasani et al. (2019) focused more on the method in ARP, which is to make a probability equation using a logistic model to predict the routing algorithm selection.

2.3.3 Scenarios in ARP

Multiple authors adapted the ARP problem to other scenarios and applications. Table 4 lists examples of previous studies with different variations in situations or applications. Both Conesa-Muñoz et al. (2016a) and Gonzalez-de-Soto et al. (2015) considered a weed control operation in a field and optimised the travelled distance for the given field operations.

Conesa-Muñoz et al. (2016a) included the same input cost and capacitated machines. The authors used a single field and three fields. Each field has weed patches that need to be sprayed by chemicals. Gonzalez-de-Soto et al.'s (2015) field also has similar weed patches to the one in Conesa-Muñoz et al. (2016a). The differences are that Gonzalez-de-Soto et al. (2015) focused on the reduction of fuel consumption of the robotic tractor for both the weed and pest control in the field.

Jensen et al. (2015) examined the capacitated field operation of liquid fertiliser application using a single machine. The authors demonstrated that the optimised plans could achieve a lower travel distance compared to the conventional plans being used during the operations. Edwards et al. (2017) considered the case of neutral material operations where there is no flow of material into or out of the field (e.g. cultivation, mowing). Hence, the authors did not consider the capacity of the machines and only used one machine. The authors developed a method to optimise the travelled distance of mowing operations in the 12 instances of the field.

In orchard operations, both Bochtis et al. (2015) and Zhou et al. (2015) minimised the distance required for the field operations. Bochtis et al. (2015) looked at general mowing in orchard operation and minimised the routing plans. The authors

examined the case with one field and an incapacitated machine. Zhou et al. (2015) focused on operations in potato fields where the simulation results showed significant savings of both distance and time for the analysed cases. The authors utilised a capacitated machine.

Table 4. The different applications of studies in ARP

Application	Reference	Objectives
Weed control operation	Conesa-Muñoz et al. (2016a)	Reduce the non-working distance by considering the input cost of the machine
	Gonzalez-de-Soto et al. (2015)	Reduce the fuel consumption of the robotic tractor
Fertilising	Jensen et al. (2015)	Observe the capacitated field operation of liquid fertiliser with one machine
Neutral material operations	Edwards et al. (2017)	Develop a tool to optimise the travelled distance of mowing operations
Orchard operations	Bochtis et al. (2015)	Consider mowing in orchard operation and minimised the routing plans
	Zhou et al. (2015)	Consider operation in potato fields
Wildlife avoidance	Bochtis et al. (2014a)	Integrate the route planning of the machine and the modelling of animal behaviour

Observing wildlife avoidance, Bochtis et al. (2014a) integrated the operation of agricultural machines and the modelling of animal behaviour. The authors observed two instances of field constellations and an incapacitated machine. The output provided an algorithmic generation of an operationally feasible plan for the agricultural machine that has been tested against suspected animal escape reactions.

2.4 Algorithms for Agricultural Route Planning

Several studies used heuristics algorithms in ARP. Among others, the heuristic algorithms used are the Clarke-Wright (Seyyedhasani & Dvorak, 2017) and the greedy algorithm (Oksanen & Visala, 2009). Besides heuristic algorithms, most research suggested metaheuristic approaches to address the route planning problem. The most common algorithms applied to the problem are the Genetic Algorithm (Gracia et al., 2014; Hameed et al., 2013; Nazarahari et al., 2019) and Tabu Search (Graf Plessen, 2019; Seyyedhasani & Dvorak, 2017, 2018a).

Both Bochtis and Vougioukas (2008) and Seyyedhasani and Dvorak (2017) used the Clarke-Wright algorithm for ARP. This method is used to compute traversal sequences for an agricultural machine that operates on parallel tracks. This approach showed a reduction of non-working distance compared to naive approaches previously applied on farms (Bochtis & Vougioukas, 2008). However, the Clarke-Wright solution quality for rectangular fields could not be achieved in other irregular-shaped fields (Seyyedhasani & Dvorak, 2017). Hence, the authors employed Tabu Search (TS).

Table 5 lists the previous studies that used metaheuristic methods to optimise ARP. Besides Genetic Algorithm and TS, other evolutionary and metaheuristics algorithms found in the literature include Ant Colony Optimisation (Bakhtiari et al., 2013; Zhou et al., 2014), Simulated Annealing (Conesa-Muñoz et al., 2016a), and Particle Swarm Optimisation (Sethanan & Neungmatcha, 2016).

The Genetic Algorithm, Ant Colony Optimisation and Particle Swarm Optimisation can be categorised as Evolutionary Algorithms (Yu & Gen, 2010). Section 2.4.1 reviews Evolutionary Algorithms and Section 2.4.2 reviews TS and other metaheuristics used in ARP.

Table 5. Research on the metaheuristics method for route planning

Algorithms	Examples of Studies	Objectives
Genetic Algorithm	Gracia et al. (2014)	Optimise the route of vehicle for biomass collection
	Hameed et al. (2013)	Optimise the route planning of field robots
	Nazarahari et al. (2019)	Optimise route planning of multiple mobile robots
Tabu Search	Seyyedhasani and Dvorak (2017)	Reduce the fieldwork time
	Seyyedhasani and Dvorak (2018a)	Optimise the dynamic routing of the vehicle
	Graf Plessen (2019)	Optimise the route planning in full and partial field coverage
Ant Colony Optimisation	Bakhtiari et al. (2013)	Optimise the route planning of capacitated field operation
	Zhou et al. (2014)	Optimise the route planning of field with obstacles
Simulated Annealing	Conesa-Muñoz et al. (2016a)	Optimise the route planning in herbicide application
Particle Swarm Optimisation	Sethanan and Neungmatcha (2016)	Optimise the route planning in sugarcane field operation

2.4.1 Evolutionary Algorithms

Evolutionary Algorithms are inspired by the evolution of the organism, based on Charles Darwin's Theory of Evolution. In the literature, evolutionary processes are applied to a large variety of different problems and therefore became

significantly adapted and modified; however, the main concept was kept as follows. Evolutionary Algorithms are characterised by the presence of a group of individuals (population) subjected to population pressure, which contributes to natural selection (i.e. survival of the fittest) and results in an improvement of the population's average fitness. Fitness is an indicator of the degree of adaptation of the species to the environment. The higher the fitness, the more the individual adapts and fits the environment (Câmara & Câmara, 2015).

Generally, Evolutionary Algorithms focus on a subset of structures characterised by the biological evolutionary process. The convergence of Evolutionary Algorithms is not influenced by variations of the objective function. They can be utilised not only as a substitute for conventional optimisation but also expanded and enhanced (Yu & Gen, 2010). Evolutionary Algorithms may also tackle single and multi-objective optimisation, which is likely to involve more than one discipline (Knuth, 2007). They have significant benefits over traditional methods since they can be implemented simultaneously with different variable types (Galvan et al., 2003).

Evolutionary Algorithms conduct optimisation with the ability to evolve. According to Yu and Gen (2010), Evolutionary Algorithms have the following three key characteristics:

1. Population based

   Evolutionary Algorithms create a group of candidate solutions, called a population, to optimise the problem. The population is a fundamental principle of the evolutionary process.

2. Fitness oriented

   Every solution is called an individual in a population. Every individual has its own characteristics and performance evaluation (fitness value). Evolutionary Algorithms select fitter individuals with a higher probability. This is the main principle of optimisation and its convergence to local or global optima.

3. Variation driven

Individuals must experience a series of different operations to mimic gene changes in genetics. The operations include several variations of the recombination process (e.g. crossover and mutation) that are key to finding a solution.

Figure 3 shows the general flowchart of an Evolutionary Algorithm. The algorithm starts with the generation of a new population, and the fitter individuals are selected. The selected individuals are recombined with specific operations. The fitness evaluation is performed to detect the best individual. If one of the stopping criteria of the algorithm is met, the algorithm will end, otherwise it will repeat the process from the selection of fitter individuals. The examples of stopping criteria are the maximum number of generations/iterations reached or if there is no improvement in the solution after several generations.

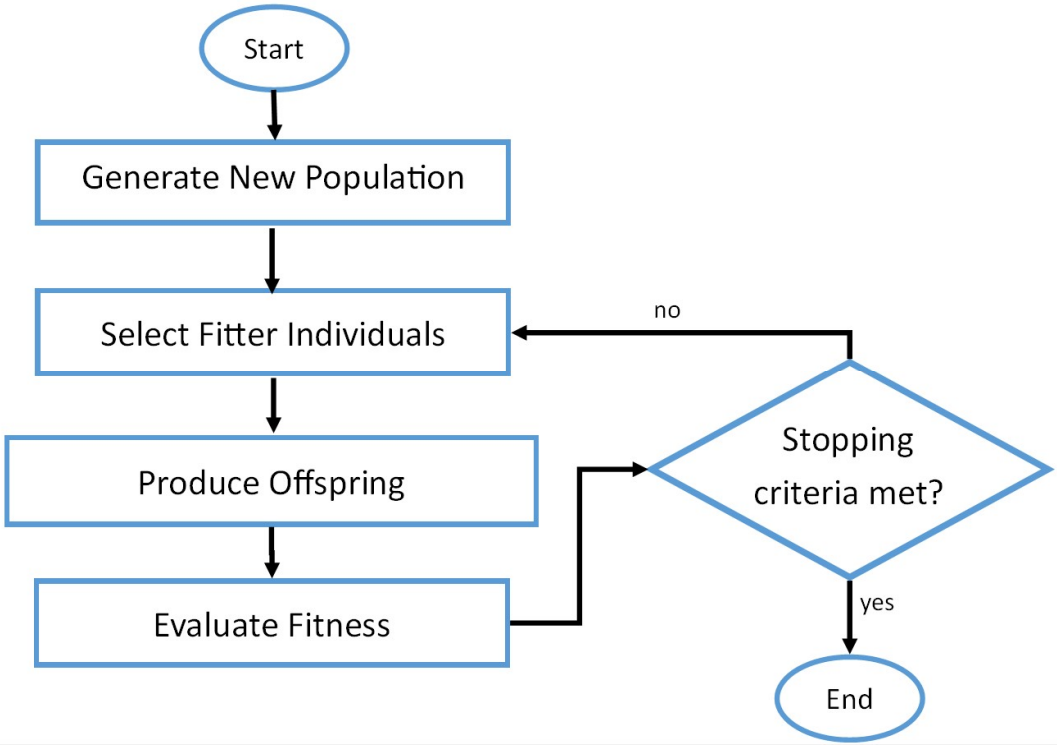


Figure 3. General flowchart of an Evolutionary Algorithm (adapted from Voratas Kachitvichyanukul (2012))

An increased performance of Evolutionary Algorithms is usually obtained by using hybrid approaches. An example of a hybrid approach in Evolutionary Algorithms is the combination of an Evolutionary Algorithm with a Local Search (Jung & Moon, 2002). Local Search can be applied to the best individual of an

Evolutionary Algorithm to find improvement in the solution. Local Search will be explained further in Section 2.4.2.2.1.

The Genetic Algorithm is the most well-known Evolutionary Algorithm. Sections 2.4.1.1 to 2.4.1.3 give a short description and an example of the application of the Genetic Algorithm, Ant Colony Optimisation and Particle Swarm Optimisation, respectively.

2.4.1.1 Genetic Algorithm

The Genetic Algorithm (GA) is an optimisation method based on the concept of genetics and natural selection. GA uses a population formed of individuals to evolve to a state that maximises fitness under the defined selection procedure (Haupt & Haupt, 2004). In general, GA simulates a natural evolution process that generates individuals by selecting the best candidates from the current generation to be applied crossovers and mutations. GA is an effective optimisation technique with possible application to many types of problems (Gen & Cheng, 2000). This is possible as the representation of the individuals and the operators is adaptable and requires only fragments of knowledge. GA only optimises values of the objective function, not their derived function or other auxiliary knowledge (Sivanandam & Deepa, 2008). The following terminology is used in GA:

- **Chromosome**

  An individual of the population (species). Different generations contain a potentially different population. Every chromosome portrays a possible solution to the problem.

- **Genes**

  A chromosome is composed of genes (features/characters). Genes are located at specific places on chromosomes called loci (positions).

- **Fitness**

  The objective function/value that defines whether a solution is excellent. The fittest chromosomes with the best fitness will survive.

Figure 4 illustrates one generation of a GA. A generation in GA contains a new population with several chromosomes. Generally, the algorithm starts from the generation of a new population and continues with the selection of chromosomes with a better fitness value. The higher the fit of a chromosome, the higher its chances to be selected. Crossover and mutation are applied to the chosen chromosomes with a given probability rate. The crossover process cuts two chromosomes at one point, and the halves are spliced together to create new chromosomes. Meanwhile, the mutation process changes the value of genes in a small group of chromosomes. Crossover is significant because it allows characteristics from two different parents to be mixed, while mutation makes a few random changes, which can be an excellent way to quickly explore the search space (Sivanandam & Deepa, 2008).

After crossover and mutation, the fitness evaluation detects the best solution, which is recorded. The whole process in Figure 4 is repeated until a stopping criterion is met. The process of evolution in a population of chromosomes over multiple generations represents a search for a good and feasible solution.

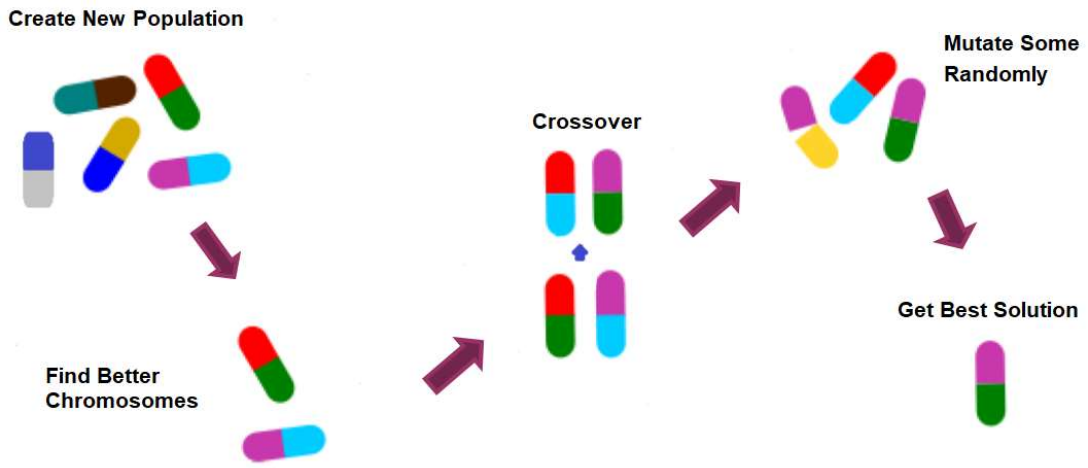


Figure 4. Illustration of a generation in a Genetic Algorithm

In the context of ARP, Gracia et al. (2014) aimed for a reduction to the total distance travelled in biomass transport by using GA and Local Search to adjust the exploration and exploitation of the method. Hameed et al. (2013) worked on the navigation of field robots by using GA to obtain relatively low computational times compared to an off-line planning system. GA was utilised to optimise the harvest plan of sugarcane to make the harvesting point as close as possible to the ideal (Florentino et al., 2018). Nazarahari (2019) used an extended GA to solve the route

planning of multiple mobile robots. The extended GA used five arithmetic crossover and mutation operators to find the optimal route.

2.4.1.2 Ant Colony Optimisation

Ant Colony Optimisation (ACO) is an Evolutionary Algorithm inspired by the behaviour of ants (Barbosa, 2013). It is a search technique inspired by the pheromone trail behaviour of actual ant colonies. Once the ants eat, they walk back to the nest through the forest marking the path with a pheromone, a chemical indicator, as a guide for their nest-mates to the food source. Individual ants might first follow different pathways to the same food source. Yet, the shortest trail will accumulate the highest concentration of pheromone, increasing the probability of being chosen. In ACO, a group of agents, or artificial ants, look for good solutions to a given optimisation problem (Adrian et al., 2015). The following terminology is used in ACO:

- **Ant**

  An individual in a colony or population for different iterations. Every ant represents a possible solution to the problem.

- **Pheromone**

  A trail deposited by ants with food (representing a solution) as they return to their nest; the pheromone attracts other ants along the trail to find the food.

- **Evaporation**

  A variable that decreases all trail values over time, to prevent unlimited trail accumulation over a component. The evaporation (in the trail) will enable the ant colony to slowly forget history and direct the searching in new directions. This process will avoid convergence to a locally optimal solution.

The objective is to find the shortest path in a graph containing vertices and edges. In ACO, every edge is associated with a parameter called a pheromone, which used and modified by the ants to travel the graph and find the solution (Dorigo & Stützle, 2004).

At every iteration, each ant constructs a solution (the path with no revisited vertices) by using a stochastic process of pheromone bias to select the next vertex and, thus, the path. At the end of an iteration, the pheromone values are adjusted (increase frequency while evaporating) proportionally to the level of the quality of the solutions produced by each ant as a stimulus in the path selection in the next iteration (Panigrahi et al., 2011).

In ARP applications, Zhou et al. (2014) used ACO for a problem containing a single field with obstacles. The authors demonstrated that ACO could produce better solutions compared to the farmers' manual operations. Bakhtiari et al. (2013) also applied ACO to optimise the non-working distance of capacitated field operation. A comparison of the ACO optimal plan with the conventional operator-generated plan shows savings in the non-working distance.

2.4.1.3 Particle Swarm Optimisation

Particle Swarm Optimisation (PSO) is inspired by the social behaviour of birds flocking or fish schooling. PSO is initialised with a group of random solutions and an optimum solution is investigated by updating swarms (Panigrahi et al., 2011). The following terminology is used in PSO:

- **Particle**

  Individuals in a swarm or population for different iterations/generations. Each particle represents a possible solution to the problem.

- **Position**

  A position vector or location vector that represents the position of a particle in space.

- **Velocity**

  A variable to direct the movement or motion of a particle's position. It is similar to local minimisers that use derivative information because velocity is a position derivative.

Generally, the concept in PSO is that the rest of the swarm will be able to follow quickly if one particle sees a desirable path to go (e.g. to food or safety). The

particles fly through the problem space by following the current optimum particles. To support the exploration of the search space, each particle typically has a certain level of randomness in its movement so that the swarm's movement has a particular explorative ability. The explorative ability means that the swarm should be influenced by the rest of the swarm but it should also explore to some extent independently (Sivanandam & Deepa, 2008).

Every particle is provided with a position and a velocity within the multidimensional space. A swarm of particles with randomly initialised positions and velocity would move towards the (local) optimal position along a path that is iteratively updated based on the current best position of each particle and the global best position of the whole swarm. The particle that has the current best position (solution) from a swarm of particles in an iteration is termed cBest. The particle that has the best position of the whole swarm in the iterations so far is termed gBest. Each particle continuously focuses and refocuses its search effort based on both cBest and gBest. This behaviour mimics the adaptation of a biological agent in a swarm: an individual judges its position based on specific objective requirements, compares itself with the others and imitates the best in the whole swarm. PSO adjusts the velocity vector for each particle and then applies that velocity to the position or values of the particle (Haupt & Haupt, 2004).

Every particle's fitness is evaluated based on an objective value. If a particle's fitness is better than cBest, the position vector for the particle is saved as the new cBest. If the fitness is better than gBest, the position vector for the particle is kept as the new gBest. The particle's velocity and position are updated based on cBest and gBest. The swarming process is best illustrated by following the path of one of the particles until it reaches the global minimum (Clerc, 2010).

In ARP applications, Sethanan and Neungmatcha (2016) employed PSO with several structures for route planning in a sugarcane field operation; it provided competitive results compared to the basic GA.

2.4.2 Other Metaheuristics and Heuristics Methods

2.4.2.1 Neighbourhood Search

2.4.2.1.1 Local Search

Local Search is a heuristics approach of finding approximate solutions to difficult optimisation problems. The general idea is to start from an initial solution by examining neighbouring solutions and then search for successive improvements. Local Search is based on an iterative exploration of the neighbourhood structure $N(x)$ of the current solution x. $N(x)$ uses combinatorics or mutation operator to define the neighbours, for example, swap, insertion/add and interchange (Vaessens et al., 1998).

Beginning with the initial solution x, Local Search looks for a better solution x' from $N(x)$ for each iteration. If x' is found, it will be set as the new solution x. Local Search will finish after finding a solution x in a way that the local optimum is already achieved concerning $N(x)$. The most popular search strategies are: 1) first improvement or greedy, where an improved solution being identified in $N(x)$ is directly set to be the solution for the next neighbourhood; and 2) best enhancement, which selects the best of all solutions in $N(x)$ to be the new solution x (Hansen et al., 2017).

2.4.2.1.2 Variable Neighbourhood Search

Variable Neighbourhood Search (VNS) is a solution refining approach that conducts a search inside a solution space that repeatedly alternates the neighbourhood structures (Mladenović & Hansen, 1997). VNS embeds a Local Search heuristic for solving the problem. VNS is based on incremental adjustments in the neighbourhoods, either in the descent phase (to find a local minimum) or in the disturbance phase (to escape from the local optima) (Hansen et al., 2010, 2017).

Let $N = \{N_1(x), N_2(x), \ldots\ldots, N_k(x)\}$ be a set of various neighbourhood structures. The basic version of the VNS method begins with the exploitation of $N_1(x)$ according to a specified search strategy for this structure. Once the exploitation of $N_1(x)$ has been completed (no improvement options found after several iterations), the exploitation of $N_2(x)$ begins. If improvement is found at the end of the exploitation of $N_2(x)$, the neighbourhood search $N_1(x)$ shall be resumed. Otherwise,

the next neighbourhood would be exploited (i.e. $N_3(x)$, and so on). This cyclical and systemic alternation of neighbourhoods continues until $N_k(x)$ is explored without improving the current solution or until any of the stopping criteria are met (Marinho Diana & de Souza, 2020)

2.4.2.2 Tabu Search

Tabu Search (TS) is a Local Search algorithm traversing the solution space by moving iteratively from the current solution to a (best) solution in its neighbourhood. The TS process begins with a feasible solution and moves towards a neighbouring solution such that an optimal or near-optimal solution is obtained after a number of moves. TS improves the search by not re-evaluating points already visited in search space. Such a point is considered 'tabu' to be rechecked (for several iterations). This method makes use of flexible memory structures with restrictions and aspiration criteria that exploit the search spaces (Glover & Laguna, 1997).

The general approach is to avoid entrapment in loops by preventing or penalising moves considered to be a tabu solution. The main elements of TS are the tabu list, tabu tenure and aspiration criteria. The tabu list contains several tabu moves or solutions that are prohibited from being chosen for the next iterations. Tabu tenure is the number of iterations in which the tabu move is regarded to remain tabu. Aspiration criteria are standard to accept an improved solution, even if it is created by a tabu move (Vaessens et al., 1998).

The first step in the decision process of TS is constructing several *moves* (candidate solution) in the neighbourhood. For the next step, pick a *move* and check whether the *mov*e is tabu (the *move* is tabu if listed in the tabu list). If the *move* is not tabu, accept the *move* to traverse to the chosen neighbour (centre for the next iteration), otherwise, check the aspiration criteria of the *move*. If a tabu *move* meets the aspiration criteria, accept the *move*, otherwise refuse the *move*. If the new solution from an accepted *move* is better than the current solution, update the solution. The decision process will be repeated for a number of given iterations (Ji & Tang, 2004).

In ARP applications, Seyyedhasani and Dvorak (2017) and Seyyedhasani and Dvorak (2018a) used TS to decrease the fieldwork time and dynamic routing of

multiple machines in agricultural operations. TS was selected for its capacity to compel the improvement technique to widen the search and prevent local optima. The outcomes demonstrate that in irregular-shaped fields, TS can enhance the effective field capacity more successfully than the Clarke-Wright method (Seyyedhasani & Dvorak, 2017). TS was also employed by Graf Plessen (2019) to minimise the non-working distance in full and partial field coverage in irregular-shaped fields. The authors highlighted the use of the tabu list in TS for better exploration of the algorithm and low computation runtimes in TS.

2.4.2.3 Simulated Annealing

Simulated Annealing (SA) simulates the annealing process in which a solid material is heated above its melting point and then slowly cooled to create a crystalline lattice that minimises its energy probability distribution. This crystalline lattice, made up of millions of perfectly arranged atoms, is an example of nature seeking an ideal structure. However, the rapid cooling or squeezing of the material slows down the crystal formation, and the product becomes an amorphous mass with a higher than optimum energy level. The secret to crystal formation is to regulate carefully the rate of temperature change. SA starts at a high temperature, which decreases exponentially. The slower the cooling, the better it is for finding the right solution (Laarhoven & Aarts, 2013).

Every iteration of the SA algorithm replaces the previous solution with a random 'close' solution. A cooling schedule controls the probability of accepting this solution. The cooling schedule depends on the difference between the corresponding function values and the global temperature parameter, which is gradually decreased during the process. The temperature is usually lowered slowly so that the algorithm has a chance to find the potential best valley before trying to get to the lowest point in the valley (Kirkpatrick et al., 1983).

SA can also be defined as a global search algorithm based on a Local Search system. SA can consider the worst result of a Local Search with a specific probability. SA uses a Local Search to discover the neighbourhood given by the current solution x. The Local Search will suggest a new solution y based on a defined acceptance probability of y taking the place of x (Yu and Gen, 2010).

SA hybridised with Mix-opt (mutation operators) has been used to plan the route of autonomous machines in herbicide application (Conesa-Muñoz et al., 2016b, 2016a). Mix-opt is used to accelerate the convergence of SA in the optimisation of ARP.

2.5 Research Gaps

Thus far, previous research tends to be limited to isolated case studies investigating the optimisation of small and real-world examples. Most of the works present different real-world or hypothetical fields without observing previous fields' data from the literature. The individual cases were addressed with no comparative benchmarks across different datasets. Therefore, the data from the previous studies needs to be collated to become benchmark datasets.

Table 6 shows the previous ARP literature, which varies in constraints and details. The table shows the studies that consider both field tracks the machine manoeuvres.

The second column of Table 6 lists the authors of the previous ARP studies. The third and fourth columns of Table 6 show the number of instances and number of fields in that instance, respectively. The number of instances refers to the number of different fields considered in the previous study. The fifth column of Table 6 lists the adjacency of the fields in the instances, while the sixth and seven columns refer to the multiple machines and capacity constraints. The last column refers to the presence of obstacles or barriers in those instances.

As shown in Table 6, most current studies on ARP focus on dealing with a single field, and there is limited research on ARP with multiple fields in an instance. The fourth column of Table 6 shows that none of the studies considers fields' adjacency. Furthermore, most of the ARP research utilised a single machine without capacity constraints. Hence, this study is the first to address several constraints simultaneously in ARP.

Table 6. The variations in ARP planning problems that consider both field tracks and machine manoeuvres

No.	Authors	No. of instances	Number of fields	Fields' adjacency	Multiple machines	Machine's capacity	Different capacity
1	Graf Plessen (2019)	3	1	no	no	no	no
2	Seyyedhasani, Dvorak, and Roemmele (2019)	1	1	no	yes	no	no
4	Seyyedhasani and Dvorak (2018a)	1	1	no	yes	no	no
5	Seyyedhasani and Dvorak (2018b)	3	1 & 2	no	yes	no	no
6	Paraforos et al. (2018)	1	1	no	no	no	no
7	Edwards et al. (2017)	12	1	no	no	no	no
8	Plessen and Bemporad (2017)	1	1	no	no	no	no
9	Seyyedhasani and Dvorak (2017)	1	1	no	yes	no	no
10	Conesa-Muñoz et al. (2016a)	2	1 & 3	no	yes	yes	no
11	Conesa-Muñoz et al. (2016b)	1	1	no	yes	yes	no
12	Bochtis et al. (2015)	1	1	no	no	no	no
13	Jensen, Bochtis, and Sørensen (2015)	3	1	no	no	yes	no
14	Zhou et al. (2015)	3	1	no	no	yes	no
15	Gonzalez-de-Soto et al. (2015)	1	1	no	yes	no	no
16	Zhou et al. (2014),	2	1	no	no	no	no
17	Bochtis et al. (2014)	2	1	no	no	no	no
18	Hameed et al. (2013)	1	1	no	no	no	no
19	Bochtis et al. (2013)	2	1	no	no	no	no
20	Spekken and Bruin (2013)	3	1	no	no	yes	no
21	Bakhtiari et al. (2013)	2	1	no	no	no	no

No.	Authors	No. of instances	Number of fields	Fields' adjacency	Multiple machines	Machine's capacity	Different capacity
22	Hameed et al. (2011)	2	1	no	no	no	no
23	Bochtis et al. (2010)	1	1	no	no	no	no
24	Jin and Tang (2010)	7	1	no	no	no	no
25	Oksanen and Visala (2009)	12	1	no	no	no	no
26	Bochtis and Vougioukas (2008)	2	1 & 3	no	no	no	no

Several studies have focused on real-world scenarios and the implementation of standard algorithms rather than considering specific ARP configurations in algorithm development (Jensen et al., 2015; Seyyedhasani & Dvorak, 2018a). As mentioned earlier, most of the previous research used existing metaheuristic methods for ARP. The nature of ARP of being a NP-hard problem implies the need to create and implement advance algorithms, not only to enhance the quality of solutions but also to manage the efficiency of the algorithm itself. Preliminary experiments showed that solutions of several instances in previously published studies were not optimal and the proposed algorithms in this study can find solutions with a lower objective function value

To conclude, there are several research gaps, as indicated earlier. These are:

1. No study on benchmark dataset is found in the literature.
2. Most of the algorithms in ARP are still limited to the standard algorithms.
3. Research using multiple fields and multiple machines in an instance is very limited.
4. No research has been found that considers all the constraints in Table 1 (fields' adjacency, multiple machines, machine's capacity) at the same time.

Chapter 3. Overview of the General Problem

As mentioned earlier, this research focuses on CPP in the agricultural field, which has tracks, and considers machine manoeuvres. The problem has been labelled as Agricultural Routing Planning (ARP). ARP aims to optimise the logistics of the machines in relation to their in-field movements and to other farm locations such as the depot or machine shed. The term 'machine' refers to the agricultural vehicle used by farmers to perform the agricultural operation in the field. This chapter presents a general description of ARP in Section 3.1, a representation of ARP in Section 3.2, the basic mathematical model of ARP in Section 3.3, and the manoeuvres of the machine in Section 3.4.

3.1 Description of the General Problem

In ARP, the field consists of several tracks or lanes. The tracks may contain crops, as illustrated in Figure 5. The machine can traverse each track sequentially or skip one or more tracks to do the field operation in the tracks. Each of the parallel tracks of a given, possibly different, length must be crossed exactly one time by a machine. The number of machines is possibly more than one with or without a fixed capacity. The capacity refers to the maximum volume of a machine to be filled with harvested crops.

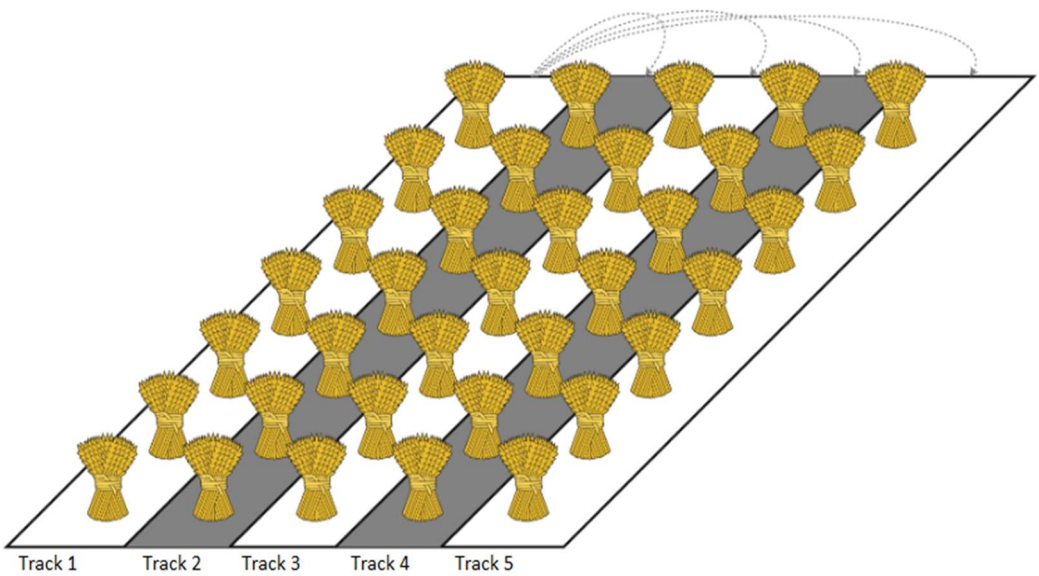


Figure 5. A field with five tracks of crops in ARP

The objective of ARP is to determine the sequence of tracks to minimise the cost of operations relative to the travelled distance. Cost, distance and time as a metric for the ARP's objective are positively correlated. The further the distance and the longer the time to complete the field operation, the higher the operating costs of the machines.

3.2 Representation of the Problem

Figure 6(a) illustrates a field with 12 tracks that need to be traversed by the machine(s). The parallel tracks of a given, possibly different, length must be traversed by a given number of machines with or without a defined capacity. The capacity refers to the maximum volume of a machine to be filled with harvested crops. The tracks have a so-called headland area at each end (marked in grey in Figure 6) to allow machines to perform manoeuvres to change direction or shift to another track. In ARP, the tracks represent nodes or vertices on a graph. Each node must be visited by exactly one machine. The arcs connecting the two nodes represent a way for the machine to switch from one track to another. In some instances, the machines start and end at a specific location outside the tracks, such as the farm depot. The goal is to find the minimum length of all trips for a group of machines to cover all tracks.

The optimisation algorithm assigns machines to specific routes based on their capacities. For example, if a machine cannot carry any more harvested crop, it must return to the depot to unload before beginning the next trip. This study assumes a uniform amount of harvest per distance travelled. Therefore, the machine capacity can be defined by the maximum distance that can be travelled on a single trip.

Figure 6(b) provides an example of two potential solutions to the problem. The first alternative (dotted edges) shows a tour where one machine is capable of harvesting the whole area. The second alternative utilises two machines, each of which covers a segment of the map. One machine covers the green part (Tracks 1–7) while the other one is traversing the remaining part (Tracks 8–12).

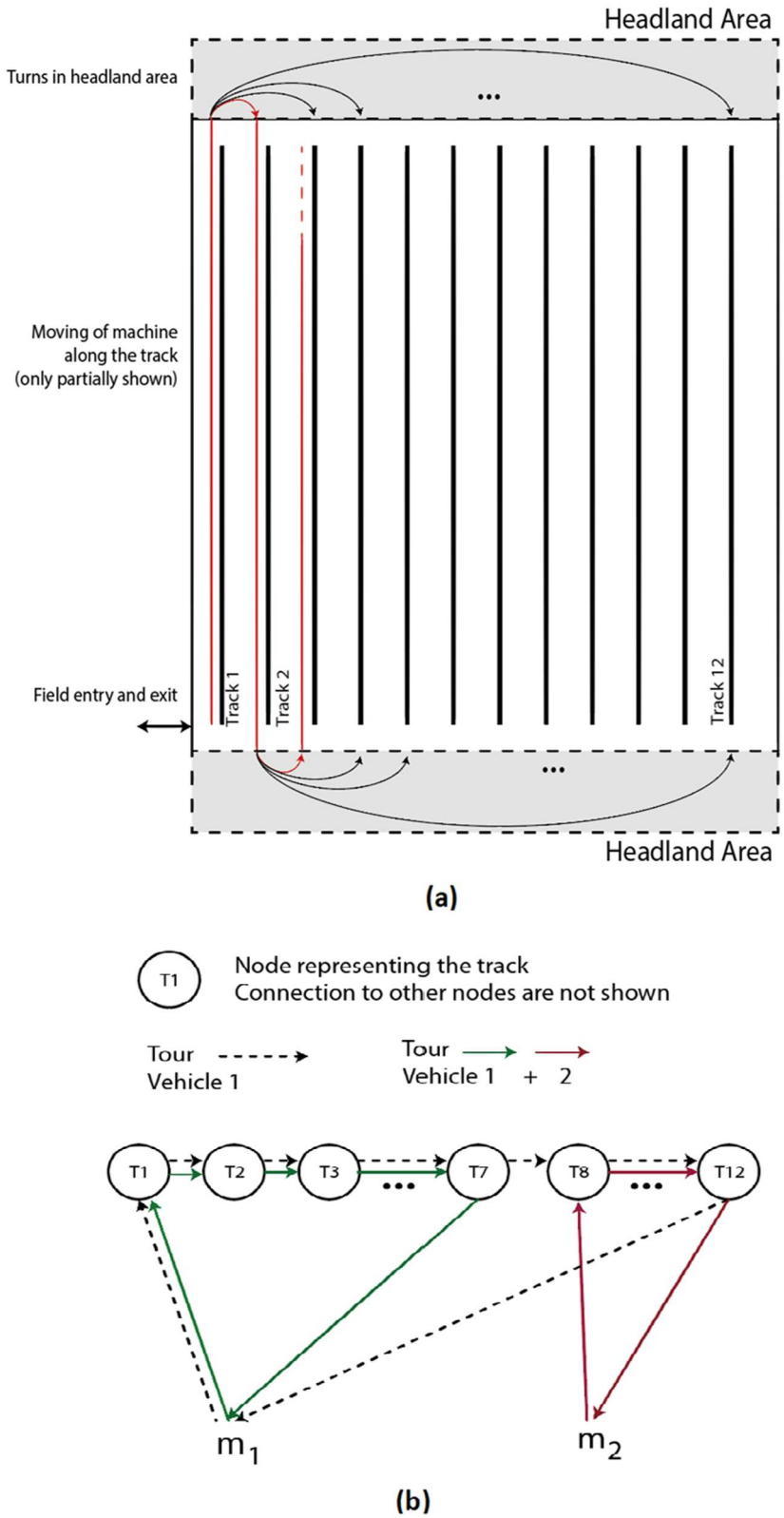


Figure 6. Illustration of (a) ARP in the field with 12 tracks; and (b) the possible solutions to the problem

3.3 Number of Fields

ARP can consider the agricultural operations in a single field or multiple fields. The illustration of the problem with a single field can be found in Figure 6(a). The majority of the previous research in ARP focuses on a single field. The details of the problem with a single field can be found in Chapter 4.

An example of problems with multiple fields is shown in Figure 7. The left side of Figure 7 is a case with two fields, while the right side is a case with four fields. The blue line indicates the tracks while the red line is the field border. The green square represents the field entrance, while the red square is the depot. The details and the representation of the problem with multiple fields can be found in Chapter 5.

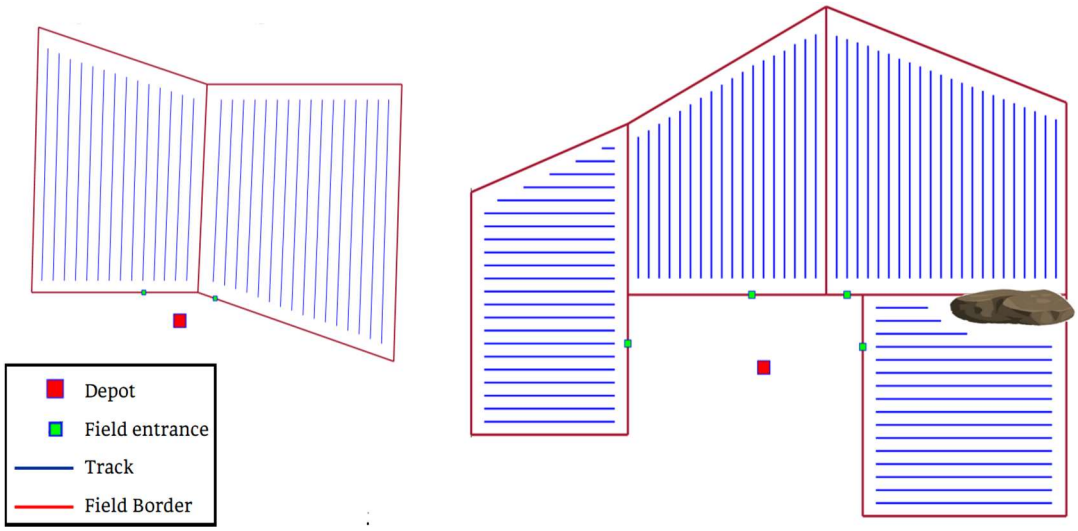


Figure 7. The illustration of the ARP problem with multiple fields

3.4 General Mathematical Model

The following mathematical formulation represents a basic ARP model (one vehicle, unlimited capacity, one field). Table 7 provides the notations of the general mathematical model for ARP.

Table 7. The variables of the basic mathematical model of ARP

Variable	Definitions
G	Graph $G = \{N, A\}$ representing the ARP
N	Set of nodes in Graph G
S	Subgraph of Graph G, $\forall S \subseteq N$
A	Set of arcs in Graph G
i, j	Nodes (tracks) index ($i, j = 0,1,2, \dots, N$)
c_{ij}	Value associated with arcs $(i,j) \in A$ representing distance, cost or time to transit from i to j
x_{ij}	Decision variable, $x_{ij} = 1$ if the machine moves from Nodes i to Nodes j, otherwise $x_{ij} = 0$

$$f = min\left(\sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij}\right) \quad (1)$$

Subject to:

$$\sum_{i \in V} x_{ij} = 1 \text{ , } j \in V: i \neq j \quad (2)$$

$$\sum_{j \in V} x_{ij} = 1 \text{ , } i \in V: i \neq j \quad (3)$$

$$\sum_{j \in V} x_{ij} = \sum_{j \in V} x_{ji}, \quad i \in V \quad (4)$$

$$\sum_{i \epsilon S} \sum_{j \in S} x_{ij} \leq \|S\| - 1, \forall S \subseteq V, \|S\| \geq 1 \quad (5)$$

Equation (1) is an objective function aimed at reducing the total cost, time or distance defined by a series of arcs. The constraints of this objective function are determined by Equation (2) through to Equation (5). Equations (2) and (3) ensure that each node is visited only once. Equation (4) guarantees that when the machine reaches the vertex, it also leaves the vertex. Equation (5) is the sub-tours elimination restriction that excludes every disjoint sub-tour from the solution.

A range of assumptions are utilised in the formulation of mathematical models to describe fieldwork operations through the ARP. First, when a machine traverses a track, it will complete the field operation for that track. Second, the field

is assumed to be situated on flat land or surface, and the tracks are assumed to be straight and parallel. Third, the headland area is considered wide enough for the machine to manoeuvre.

The problem can also vary with the individual capacity for each vehicle to accommodate heterogeneous machines with a variety of handling characteristics assuming they operate on the same paths. The maximum distance inside the track will depend on the maximum capacity of a machine. The modification and extension of the mathematical model can be found in Sections 4.3 and 5.2.

3.5 Machine Manoeuvres

Figure 8 illustrates the four kinds of machine manoeuvres considered in this study: (a) bulb, (b) flat, (c) flatΘ and (d) bulbΘ. The formula to calculate the length of the manoeuvres is based on Jin and Tang (2010). The brown machine represents the current location of the machine, while the grey machine represents the machine before manoeuvring.

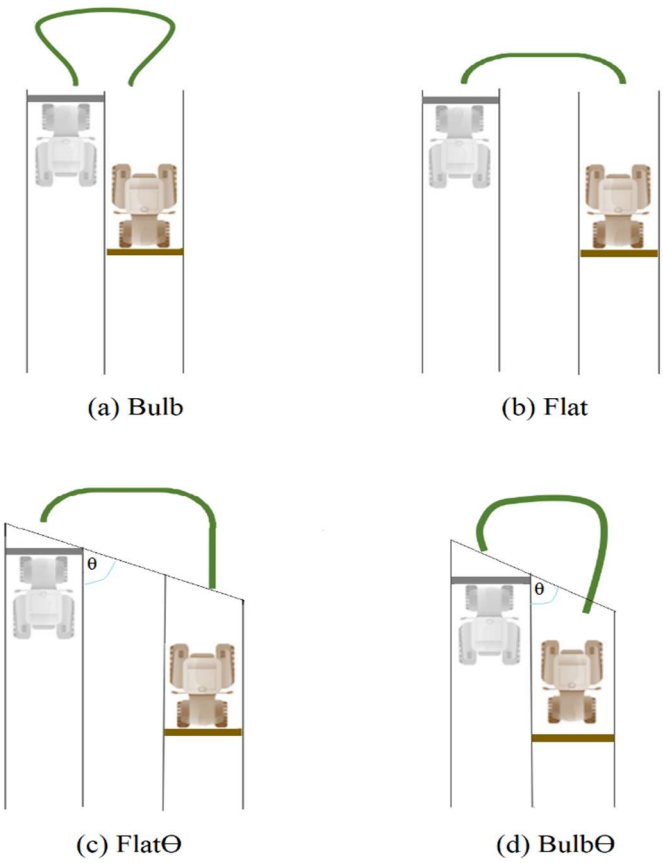


Figure 8. The four types of machine manoeuvre considered in this research

Figure 9 illustrates the condition of the manoeuvres, adapted from Bochtis and Vougioukas (2008). The top side of Figure 9 shows an ARP problem and the connection of operating width (w) and the minimum turning radius of a machine (r), while on the bottom side is a corresponding graph that represents each track as a node on the graph.

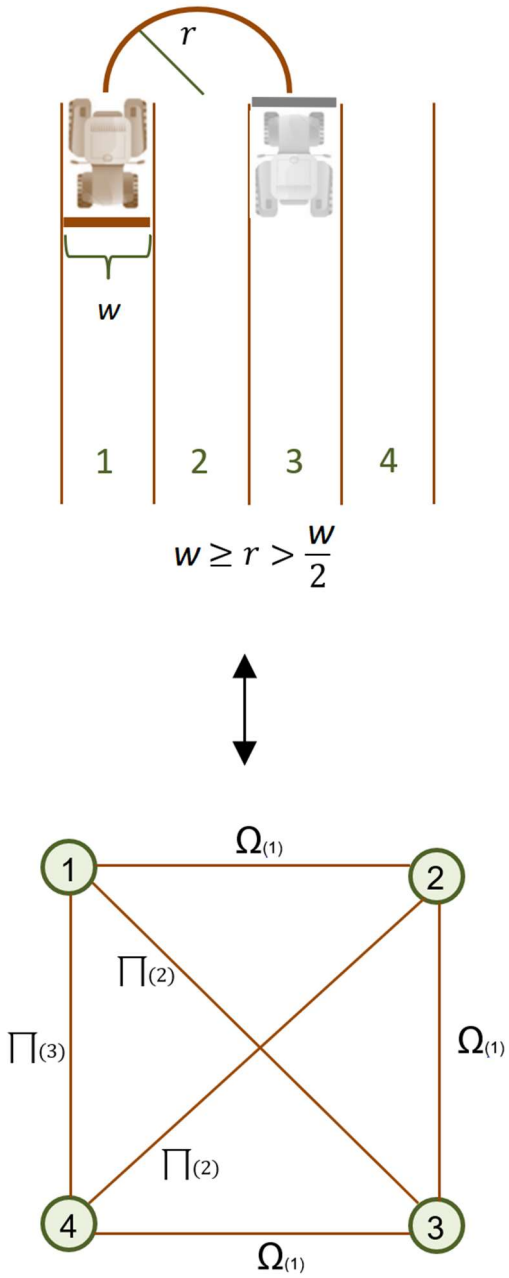


Figure 9. Illustration of (top) the problem where required to skip a track to do a flat turn; and (bottom) the related turning forms of the machine (adapted from Bochtis and Vougioukas (2008))

In Figure 9, since $w > r$ and $r > w/2$, the flat (Π) turn can be made when a machine skips at least one track. Otherwise, the bulb (Ω) turn will take place. Figures 8(c) and 8(d) describe the flatθ and bulbθ turns, which happen when the previous track and the next track are aligned with an acute angle $\theta < 90$.

The formulas to calculate the length of the manoeuvres are listed in Table 8 (Jin & Tang, 2010). Every track is symbolised with a number. The *i* refers to the current track, *j* is the next track, and |*i*-*j*| is the absolute distance between *i* and *j*. For example, if *i* = *3* and *j*= *1*, than |*i*-*j*| = *2*.

Table 8. The type of manoeuvre and the formula to calculate the length required for that manoeuvre

Type of Manoeuvre	Calculation Formula
Flat (Π)	$\mathrm{r}\left(3\pi - 4\sin^{-1}\left(\frac{2\mathrm{r} + \lvert i-j\rvert w}{4r}\right)\right)$
Bulb (Ω)	$\lvert i-j\rvert w + (\pi - 2)\mathrm{r}$
FlatΘ (Πθ)	$\lvert i-j\rvert w(1+\cot\theta) + r(\pi - 2)$
BulbΘ (Ωθ)	$\pi r + \frac{4r^2 - \lvert i-j\rvert w(4r + \mathrm{w}\cot^2\theta + \mathrm{w})}{4r - 2\mathrm{w}\lvert i-j\rvert} \times \sin^{-1}\frac{\lvert i-j\rvert w(4r\cot\theta - 2\mathrm{w}\cot\theta)}{4r^2 - \lvert i-j\rvert w(4r + \mathrm{w}\cot^2\theta + \mathrm{w})}$

The ARP with a capacitated machine is close to VRP, while the general ARP without considering capacity constraint is similar to TSP (Bochtis & Sørensen, 2009). Since both VRP and TSP are considered as NP-hard (Bochtis & Sørensen, 2009; Wang et al., 2019), then we can assume that ARP also belongs to NP-hard problems.

Chapter 4. Optimisation of Dataset in ARP

This chapter is written as the author's journal article 'Optimisation of agricultural routing planning in field logistics with Evolutionary Hybrid Neighbourhood Search' published in *Biosystems Engineering*, vol. 184, 2019. The full paper can be found in Appendix A. Sections 4.3 to 4.5 are taken directly from the article (Utamima et al., 2019c), while Sections 4.1 and 4.2 are enriched with additional information.

As indicated earlier, previous ARP studies that focus on benchmark datasets are not found in the literature. The lack of benchmarking for the collection of datasets limits the comparability of algorithms and the encouragement of the transfer of findings to other instances. Hence, Utamima et al.'s (2019) study is the first to gather problem datasets from previous publications that describe different field layouts. The layouts were derived either from those given in the literature or by requesting them from the previous authors. The collected datasets are presented in Section 4.1 and 4.2.

4.1 Rectangular Fields

The rectangular field is the general type most researchers use in ARP studies (Bochtis & Sørensen, 2009). The field typically has four sides, with headland areas in the north and south of the field. Table 9 presents the details of the dataset containing the rectangular fields identified in the previous publications. The second and third columns of Table 9 list the problem codes and the number of tracks (problem size) in the field. The 'Track Width' and 'Track Length' columns contain the width and length of the tracks, the 'Type of Manoeuvre' column refers to the types of manoeuvres that can be used in this particular dataset, the 'Machine' column details the number of machines that are used—note that all cases use the homogeneous type of machine(s), the 'Turning Radius' column refers to the machine's turning radius, and the last column lists the references to the data.

Table 9. The dataset that contains rectangular fields

No.	Problem Code	Size	Track Width (metres)	Track Length (metres)	Type of Manoeuvre	Machine (Homogeneous type)	Turning Radius (metres)	References
1	8rt	8	2.89	30	Flat, bulb	1 (single)	3.50	(Bochtis & Vougioukas, 2008)
2	12rt_a	12	2.5	40	Flat, bulb	1 (single)	3.50	(Conesa-Muñoz et al., 2016a)
3	12rt_b	12	2.5	70				
4	20rt	20	2.5	80				
5	90rt	90	10	100	Flat, bulb	3 (multiple)	7.00	(Seyyedhasani & Dvorak, 2017)

4.2 Irregular-Shaped or Non-Convex Fields

The irregular-shaped or non-convex field is a field that varies in shape from trapezium to curved; see Figure 10 for an illustration of irregular-shaped fields. These fields are closer to real cases and general farm layouts (Zhou et al., 2014).

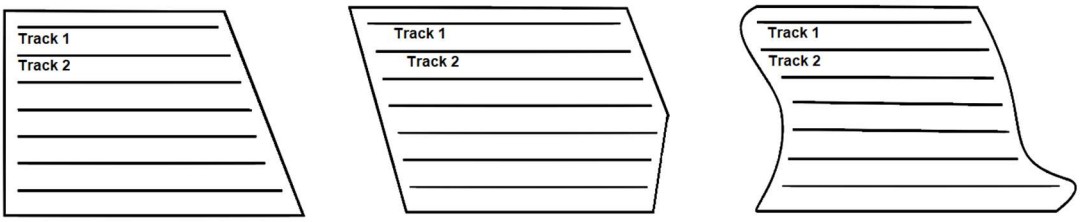


Figure 10. The illustration of the irregular-shaped fields

Table 10 listed the datasets with irregular-shaped fields used in previous studies. Problem 1 is a field with 37 tracks, observed in Conesa-Muñoz et al. (2016a) and Hameed et al. (2011). The length of the field varies between 19.89 and 206.8 metres, with three kinds of manoeuvres and operated by a single machine. Problems 2 and 3 use the same field but with 74 tracks. The difference between this and Problem 1 is that Problem 2 only considers two manoeuvres, as presented in Seyyedhasani and Dvorak (2018b), while Problem 3 considers four manoeuvres. Problem 4 is an H-shaped field, as described in Oksanen and Visala (2009).

Table 10. The datasets that contain the irregular-shaped

No	Problem Code	Size	Track Width (metres)	Track Length (metres)	Type of Manoeuvre	Machine (Homogeneous type)	Turning Radius (metres)	References
1	37nc	37	9	Vary between 19.89 and 206.8	Flat, bulb, flatΘ	1 (single)	6.00	(Conesa-Muñoz et al., 2016a; Hameed et al., 2011)
2	74nc	74	19	Vary between 54.08 and 671.56	Flat	3 (multiple)	9.77	(Seyyedhasani & Dvorak, 2018)
3	74nc2	74	19		Flat, bulb, flatΘ, bulbΘ	3 (multiple)	9.77	-
4	62Hs	62	2.5	Vary between 75 and 210	Flat, bulb	1 (single)	2.30	(Oksanen & Visala, 2009)

4.3 Mathematical Model

Table 11 listed the variables used in the mathematical model. The mathematical model to solve the collected datasets is formulated as a binary integer programming model. The group of tracks in a field are assumed to be a set of nodes and arcs in Graph *G*. The detailed description of parameters is given in the Nomenclature. The binary decision variables (x_{ij}^{m}) indicate the movement of the machine(s) between two nodes.

Table 11. The description of variables used in the mathematical model

Variable	Description
G	Graph $G = \{N, A\}$ representing the field's layout
N	Set of nodes in Graph G
A	Set of arcs in Graph G
S	Subgraph of Graph $G, \forall S \subseteq N$
n	Number of tracks in the field
i, j	Nodes indices (i, j = 0, 1, 2, …, n), 0 is the depot
T	Set of tracks (t = 1, …, n)
l_t	Length of track t
M	Set of machines available at the depot
m	Machine $m, m \in M$
r	The minimum turning radius of a machine
ω	The effective operating width of a machine
d_{ij}	The manoeuvre's degree between track i and j
θ	The degree of tilt between the previous and next track, note that $0 < \theta \leq 90$
Q	the maximum distance that can be travelled by a machine(s) (the machines are assumed homogeneous)
$\Omega(d_{ij})$	The bulb type of manoeuvre turn executed by a machine in the headland area
$\Pi(d_{ij})$	The flat type of manoeuvre turn executed by a machine in the headland field area
$\Pi\Theta(d_{ij})$	The flat type of manoeuvre turns with θ degree, executed by a machine in the headland field area if the previous track and the next track are aligned with an angle θ
$\Omega\Theta(d_{ij})$	The bulb type of manoeuvre turns with θ degree, executed by a machine in the headland field area if the previous track and the next track are aligned with an angle θ
x_{ij}^{m}	The decision variable, equal to 1 if machine m moves from node i to node j $(i, j \in A)$, otherwise it is equal to 0

The objective function (Equation 1) minimises the non-working distance of the machines. The total distance (Equation 2) is a sum of the total tracks' length and the minimisation of the non-working distance of the machines. The non-working distance refers to the distance that is traversed by a machine when moving to the headland areas or specific locations outside the field, for example, the warehouse or garage.

A machine can perform four kind of manoeuvres. Constraints (3)–(7) specify the four different manoeuvres: flat (Π), bulb (Ω), flatΘ (ΠΘ) and bulbΘ (ΩΘ). Every track in a field is needed to be visited once, hence Constraints (8)–(9) ensure that every node is visited only once by the machine. Constraint (10) guarantees that if a machine enters a node (a track), it will also leave that node again (Flow constraint). Constraint (11) excludes disjoint sub-tours from a solution, hence there will be only one tour for each machine. Constraint (12) restricts the maximum distance for a machine. The last Constraint (13) specifies that the decision variable is a binary number.

$$z = \min\left(\sum_{i\in N}\sum_{j\in N}\sum_{m\in M} d_{ij}.x_{ij}^{m}\right) \tag{1}$$

$$Total\ Distance = \sum_{t\epsilon T} l_t + \min\left(\sum_{i\in N}\sum_{j\in N}\sum_{m\in M} d_{ij}.x_{ij}^{m}\right) \tag{2}$$

s.t.

$$d_{ij} = \begin{cases} \Pi(i,j),\ if|i-j| \le \frac{2\mathrm{r}}{w}\ \wedge\ \theta = 90 \\ \Omega(i,j),\ if|i-j| > \frac{2\mathrm{r}}{w}\ \wedge\ \theta = 90 \\ \Pi\Theta(i,j), if|i-j| \le \frac{2\mathrm{r}}{w}\ \wedge\ \theta < 90 \\ \Omega\Theta(i,j), if|i-j| > \frac{2r}{w}\ \wedge\ \theta < 90 \end{cases} \tag{3}$$

$$\Pi(i,j) = |i-j|\cdot w + (\pi - 2)r \tag{4}$$

$$\Omega(\mathrm{i},\mathrm{j}) = \mathrm{r}\left(3\pi - 4\sin^{-1}\left(\frac{2\mathrm{r}+|i-j|\cdot w}{4\mathrm{r}}\right)\right) \tag{5}$$

$$\Pi\Theta(i,j) = |i-j|\cdot w(1+\cot\theta) + r(\pi - 2) \tag{6}$$

$$\Omega\Theta(i,j) = \pi r + \frac{4r^2 - |i-j|w(4r + \mathrm{w\,c}\quad{}^2\theta + \mathrm{w})}{4r - 2\quad|i-j|} \times \sin^{-1}\frac{|i-j|w(4r\cot\theta - 2\mathrm{w}\cot\theta)}{4r^2 - |i-j|w(4r + \mathrm{w\,c}\quad{}^2\theta + \mathrm{w})} \tag{7}$$

$$\sum_{m\in M}\sum_{i\in N} x_{ij}^{m} = 1, \quad i,j \ne 0,\ j \in N: i \ne j \tag{8}$$

$$\sum_{m\in M}\sum_{j\in N} x_{ij}^{m} = 1, \quad i,j \ne 0,\ i \in N: i \ne j \tag{9}$$

$$\sum_{m\in M}\sum_{i\in N} x_{ij}^{m} = \sum_{m\in M}\sum_{j\in N} x_{ji}^{m}, \quad i,j \in N \tag{10}$$

$$\sum_{i\epsilon S}\sum_{j\in S} x_{ij}^{m} \le \|S\| - 1, \forall S \subseteq N, \|S\| \ge 1, m \in M \tag{11}$$

$$\sum_{i\in N}\sum_{j\in N} l_i x_{ij}^{m} < Q, m \in M,\ i,j \in N \tag{12}$$

$$x_{ij}^{m}\epsilon\{0,1\} \tag{13}$$

4.4 Evolutionary Algorithms to Solve the Datasets

The proposed algorithm Evolutionary Hybrid Neighbourhood Search (EHNS) adapts an evolutionary technique and combines it with Mutation-based Neighbourhood Search and Tabu Search. In general, a starting population of the solution is modified over multiple generations by selecting individual solutions, which are mutated and improved by a hybridised algorithm to form the next generation. The mutation operator that acted as a neighbourhood search is a strong technique to obtain an effective solution (Conesa-Muñoz et al., 2016b).

In EHNS, each iteration begins with the calculation of the objective function of every candidate solution. Then, the roulette wheel selection is performed. The roulette wheel selection assumes a higher probability of the survival of candidate solutions with a better fitness. Here, the probability of selecting a candidate solution is proportional to its fitness (Lipowski & Lipowska, 2012; Utamima et al., 2015).

The Mutation-based Neighbourhood Search follows the previous stage. Figure 11 shows the pseudocode for this Mutation-based Neighbourhood Search. The procedure is repeated for *max_iteration* times. In each iteration, the roulette wheel selection will choose a candidate to be applied with the mutation operator.

```
Procedure Mutation_based_Neighbourhood_Search {
   for 1 until max_iteration {
      Old ← Roulette_Wheel_Selection)
      k = rand*(3)
      switch k {
           [p1, p2] ← Select_2_different_points()
           case 1: New ← Flap(p1,p2)
           case 2: New ← Interchange(p1,p2)
           case 3: New ← Slide(p1,p2)
      }
   }
}
```

Figure 11. Pseudocode for the Mutation-based Neighbourhood Search

Three kinds of mutation operators are used, as illustrated in Figure 12. For each operator, two points (*p1* and *p2*) are randomly picked between *1* and *n* (with n being the size of the candidate solution), being used as input parameters for the following operators:

- *Flap*: flips the sequence between the points *p1* and *p2* (Figure 12 (a))
- *Interchange*: swaps the elements at position *p1* and *p2* (Figure 12 (b))
- *Slide*: moves the element at *p1* to *p2* and shifts the remaining points forward (starting from *p1* + 1 until *p2*) (Figure 12 (c)).

Old	1	2	3	4	5	6	7	8	9	10
			p1 *				p2 *			
New	1	2	4	5	6	7	3	8	9	10

(a)

Old	1	2	3	4	5	6	7	8	9	10
			p1 *				p2 *			
New	1	2	7	4	5	6	3	8	9	10

(b)

Old	1	2	3	4	5	6	7	8	9	10
			p1 *			p2 *				
New	1	2	4	5	6	3	7	8	9	10

(c)

Figure 12. Illustration of Flap (a), Interchange (b), and Slide (c) operators

The new candidates from the neighbourhood search will replace the previous candidates in the replacement step. Next, the current best solution (*cBest*) from the Mutation-based Neighbourhood Search is checked to determine whether it is better than the previous best solution (*prevBest*) so far. If so, then *prevBest* will be replaced by *cBest*. Otherwise, Tabu Search will explore the local neighbourhood around the solution *cBest*. The pseudocode for Tabu Search is listed in Figure 13. First, the parameters, which are the tenure and the number of iterations, are initialised. In each iteration, the list *swap_list* is constructed. It contains a set of *moves* (the swap of two tracks) and the respective distances. Next, Tabu Search checks whether a *move* is tabu and whether the distance of every *move* (*tabu_dist*) is worse than *tabu_Sol*. A penalty is imposed if a *move* matches these conditions. Subsequently, Tabu Search updates the *tabu_List* and *tabu_Sol* as well. At the end of Tabu Search's iteration, a *tabu_Sol* will contain the best solution. Then, *tabu_Sol* is

compared with the *gBest* (the global best solution found so far in the EHNS's generations). If *tabu_Sol* obtains a better solution, then it will replace the value of *gBest*. Otherwise, the *gBest* will remain the same.

```
Procedure Tabu_Search (Candidate_Solution) {
     Initialization of Parameters
     tabu_Sol ← Candidate_Solution
     for 1 until tabu_iteration {
         Construct swap_list
         Calculate tabu_dist for every move in swap_list
         if move is taboo and move's tabu_dist > tabu_Sol {
              give move penalty
         }
         Update tabu_List(tenure)
         Update tabu_Sol
     }
}
```

Figure 13. Pseudocode for Tabu Search

Several evolutionary strategies have been proposed to retain the best solution while maintaining diversity in the population. This study applies an evolutionary strategy called elitism (Nayeem et al., 2014; Sivanandam & Deepa, 2008), which retains a group of the best solutions over all generations. Elitism records ten per cent of the best solutions in each generation to preserve them from extinction and copy this group to the new generation (Nayeem et al., 2014). Then, a randomisation technique (Haupt & Haupt, 2004) is employed to diversify the population. The randomisation technique, called scramble, will bring a small group of solutions into a random order. This scrambled group is also copied to ten per cent of the new generation.

Figure 14 shows the last step of EHNS, which is the modification of the new generation that contains the evolutionary strategy. The number of solutions selected for both elitism and scramble are initially set to ten per cent each of the population (*ElitismSize* and *ScrambleSize*). Lines 2 to 5 in Figure 14 list the elitism process. The process starts with obtaining the best candidate in the current generation (*best*). The process continues to check whether *best* is better than the current candidate with the highest fitness (*max(EliteGroup)*) in the *EliteGroup* (the group of best candidate

solutions) and replaces it if all conditions are met. Eventually, the solutions in the *EliteGroup* replace the originally selected solutions. Next, the scramble process is applied, as shown in Lines 9 to 11. The result is copied back into the population, replacing the selected solutions.

```
1  Procedure Modify_New_Generation {
2      best ← get_BestCandidate()
3      if best < max(EliteGroup){
4          EliteGroup[max(EliteGroup)] ← best
5      }
6      for i=1 until ElitismSize {
7          New_Population[i] ← EliteGroup[i]
8      }
9      for j=i+1 until i+ScrambleSize {
10         New_Population[j] ← rand(n)
11     }
12 }
```

Figure 14. Pseudocode for the modification of the new generation in EHNS

4.5 Experimental Results in Datasets

4.5.1 Parameter settings

The parameter configuration is obtained by conducting a two-level factorial design with four factors. Every factor incorporates high and low levels (Montgomery, 2013). The algorithm is run ten times with varying random settings. Each factor is in the given range to calculate an average to compensate for the non-deterministic nature of the algorithm (Guan & Lin, 2016). Table 12 provides the details of all factors and the range of each setting. The settings use *n*, which is the number of tracks in the field, to adapt to the size of the problem. The larger the size of the problem, the greater the number of iterations and the population size. The values in bold indicate better settings for EHNS.

For the datasets introduced in Section 4.1 and 4.2, this study implemented GA, TS, and ACO to compare EHNS against those algorithms. The number of generations or iterations for GA, TS and ACO are set to match those of EHNS. In GA, the crossover rate is set to 0.7, and the mutation rate is 0.3, while TS's tenure is

set to *0.5n* (Ou-Yang & Utamima, 2013). Meanwhile, the number of ants, the weight of pheromones and the evaporation rate in ACO are set to *n, 1,* and *0.01,* respectively (Bakhtiari et al., 2013; Zhou et al., 2014). In this study, the Local Search is hybridised in ACO.

Table 12. Factor settings for EHNS

Factors	Levels		Range
	Low	High	
Number of generations	*30n*	***40n***	*20n-50n*
Population size	*3n*	***5n***	*3n-5n*
Mutation-based Neighbourhood Search iteration	*1.5n*	***2.5n***	*n-3n*
Tabu Search iteration	*0.5n*	***n***	*0.5n-n*

4.5.2 Results

Table 13 represents the experimental results of the EHNS algorithm in comparison with several algorithms found in the agricultural routing optimisation literature. The algorithms used in previous studies are GA, Mix-opt+Simulated Annealing (MS), TS, and ACO in the second, third, fourth and fifth columns of Table 13. The 'Problem Code' column refers to the datasets that are explained in Section 4.1. The values shown in bold in Table 13 indicate the best (and the lowest values) in that row, '-' indicates that there is no reported solution, '*' indicates that the result are obtained from the references, otherwise we implemented the algorithms as described in the corresponding reference (listed below Table 13).

Generally, EHNS successfully achieves either the lowest or the same objective function compared with the other algorithms. As listed in Table 13, EHNS outperforms TS in six problems (20rt, 37nc, 90rt, 74nc, 74nc2, and H-shaped). EHNS obtains the same best solution as MS for the first four problems, while for Problem 37nc, EHNS introduces a new best solution. EHNS achieves better objective function in seven problem instances compared with ACO. Moreover, EHNS outperforms GA in all problems. EHNS presents new best solutions in six of the nine datasets. Hence, this study could find for 56% of the cases an improved combination of tracks saving an average of 10.68% non-working distance compared to other algorithms.

Table 13. The non-working distance comparison

No	Problem Code	Genetic Algorithm (GA) [1, 2]	Mix-opt + SA (MS) [1, 3]	Tabu Search (TS) [4]	Ant Colony Optimisation (ACO)	**Evolutionary Hybrid Neighbourhood Search (EHNS)**
1	8rt	95.767	***94.439**	**94.439**	**94.439**	**94.439**
2	12rt_a	176.451	***146.027**	**146.027**	**146.027**	**146.027**
3	12rt_b	166.451	***145.602**	**145.602**	147.076	**145.602**
4	20rt	250.916	***235.491**	245.916	269.642	**235.491**
5	37nc	*1142.474	*961.470	1088.188	1228.129	**958.930**
6	90rt	2791.680	-	2870.172	3987.434	**2658.474**
7	74nc	5212.590	-	*4416.300	5506.67	**3880.679**
8	74nc2	6064.112	-	5856.542	6911.151	**5197.349**
9	62Hs	562.414	-	559.914	579.914	**479.914**

*The results are obtained from [1] Conesa-Muñoz et al. (2016a), [2] Hameed et al. (2011), [3] Conesa-Muñoz et al. (2016b), and [4] Seyyedhasani and Dvorak (2018a).
-The results are not available in the references.

In Table 13, TS is shown to reach the same objective function as EHNS in three problems; moreover, its solutions are better than GA and ACO in seven problems. ACO performs well in small-sized problems; however, its solution quality decreases in larger problems. GA can obtain better solutions than ACO in six problems; however, GA's solutions are not as good as TS's.

The results of Problems 37nc, 90rt, and 74nc2 are presented in detail in Figures 15, 16, and 17, respectively. Figure 15 shows the optimised path chosen by EHNS when the bulb turn is selected for all the curved area of the field, while the flat turn dominates the rectangular section of the field. The bulbΘ turn requires less turning length than the bulb turn (Jin & Tang, 2010). In contrast, the flatΘ turn needs to skip one or more tracks, and its length is similar to the flat turn with the addition of the different alignment between two tracks. Therefore, the algorithm chooses the bulbΘ turn in the curved area because it tends to cover a shorter distance than the flatΘ.

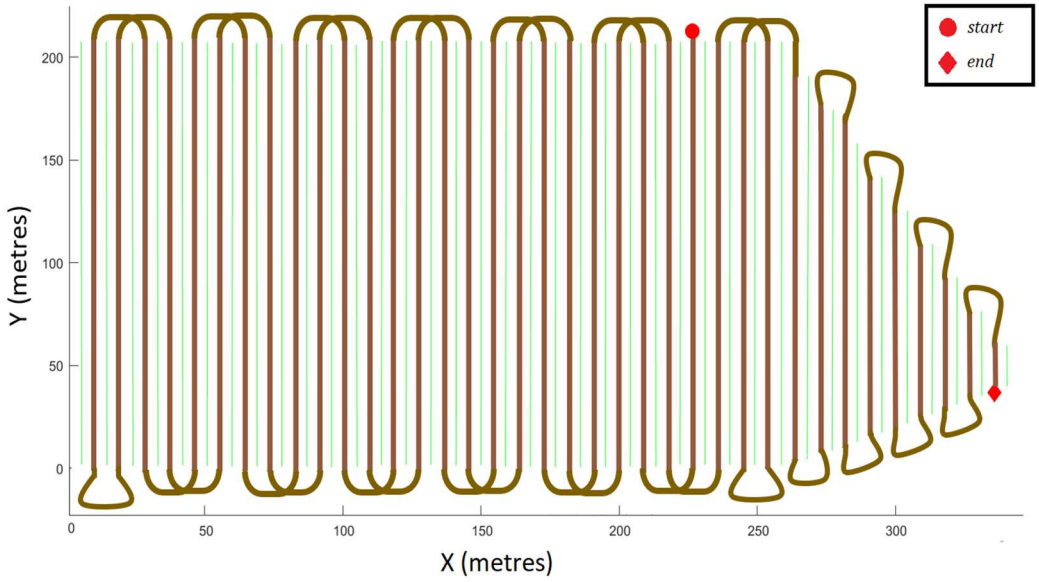


Figure 15. EHNS's manoeuvre results for the machine when solving Problem 37nc

Figure 16 shows the path planning for three machines in a rectangular field of 90rt. Most of the manoeuvres are flat turns. The colour differentiates the route for each machine. Figure 17 shows that 74nc2 path planning is the most complicated problem compared with others. This is because the machine needs to start and end at the depot, and it considers the flatΘ and bulbΘ turn. The bulbΘ turn is predominant in this irregular-shaped field.

Table 14 shows the comparative running times of GA, TS, ACO, and EHNS in four problem instances (20rt, 37nc, 90rt, and 74nc2). Generally, the running time of ACO is the fastest among the algorithms.

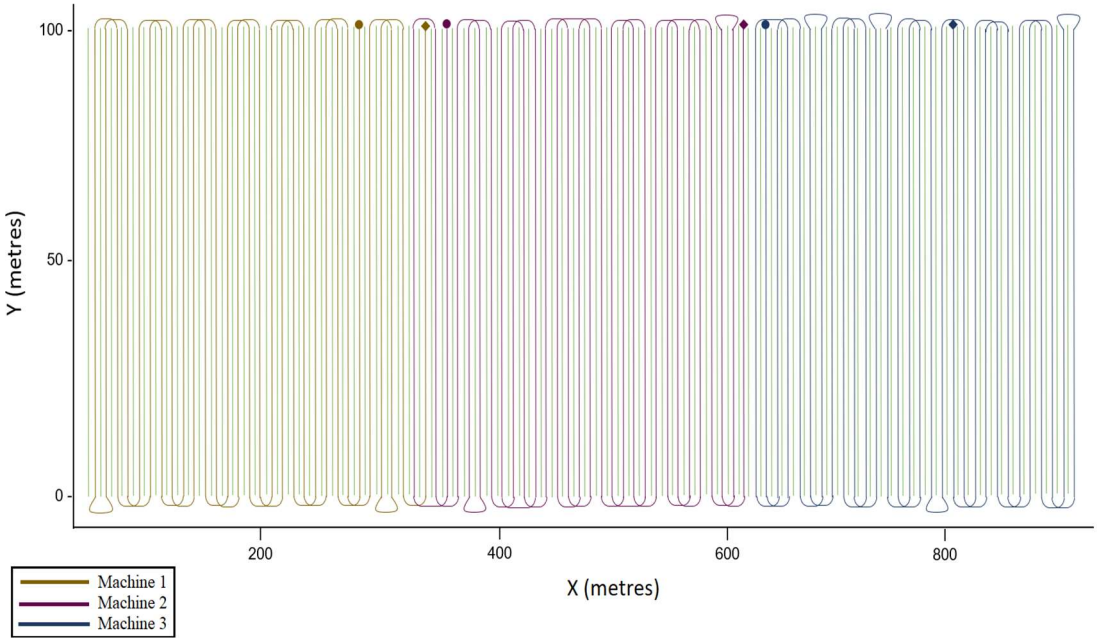


Figure 16. EHNS's manoeuvre results for the machine when solving Problem 90rt

As shown in Table 14, the running time of GA depends on the size of the problem: the greater the number of tracks, the longer the time needed. In contrast, the running times of TS, ACO, and EHNS in Problem 74nc2 are longer than in 90rt. This is because the neighbourhood search aspect checks every move's objective function, and the objective function of Problem 74nc2 contains the flatϴ and bulbϴ manoeuvres, which are more complex than the manoeuvre in Problem 90rt.

Table 14. Runtime comparison of GA, TS, ACO, and EHNS

Algorithm	Running Time (Seconds)			
	20rt	37nc	90rt	74nc2
Genetic Algorithm (GA)	3.551	8.273	23.064	20.463
Tabu Search (TS)	1.768	5.102	14.616	21.188
Ant Colony Optimisation (ACO)	1.649	4.995	13.238	16.308
Evolutionary Hybrid Neighbourhood Search (EHNS)	2.059	5.695	15.633	23.470

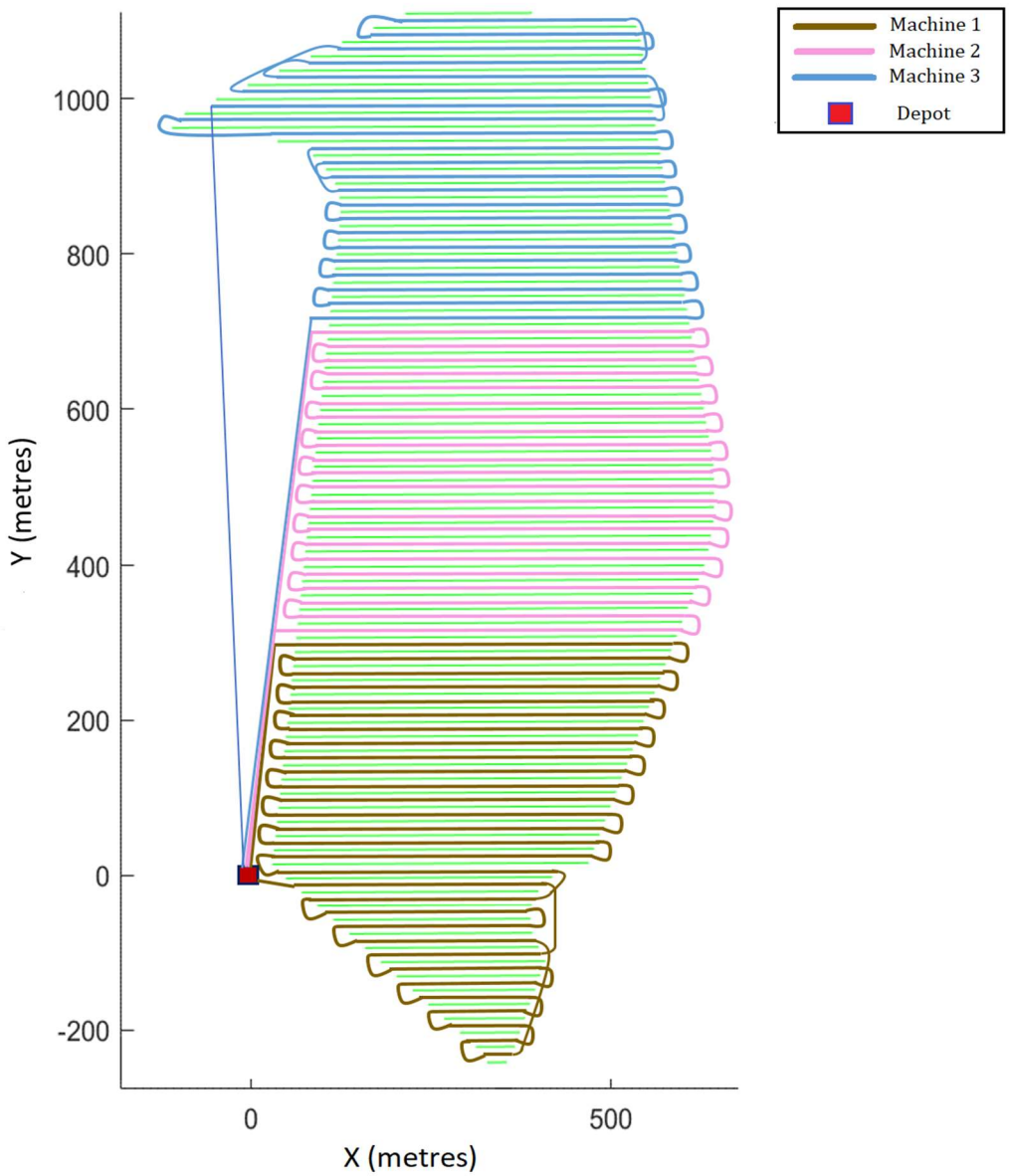
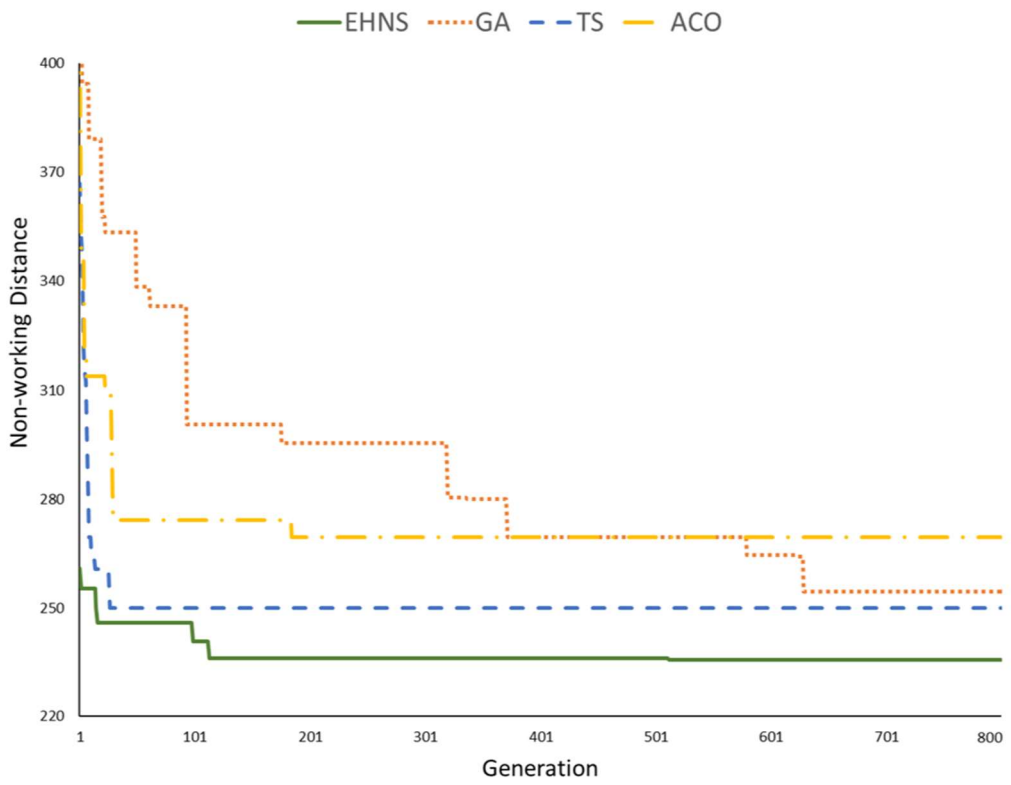
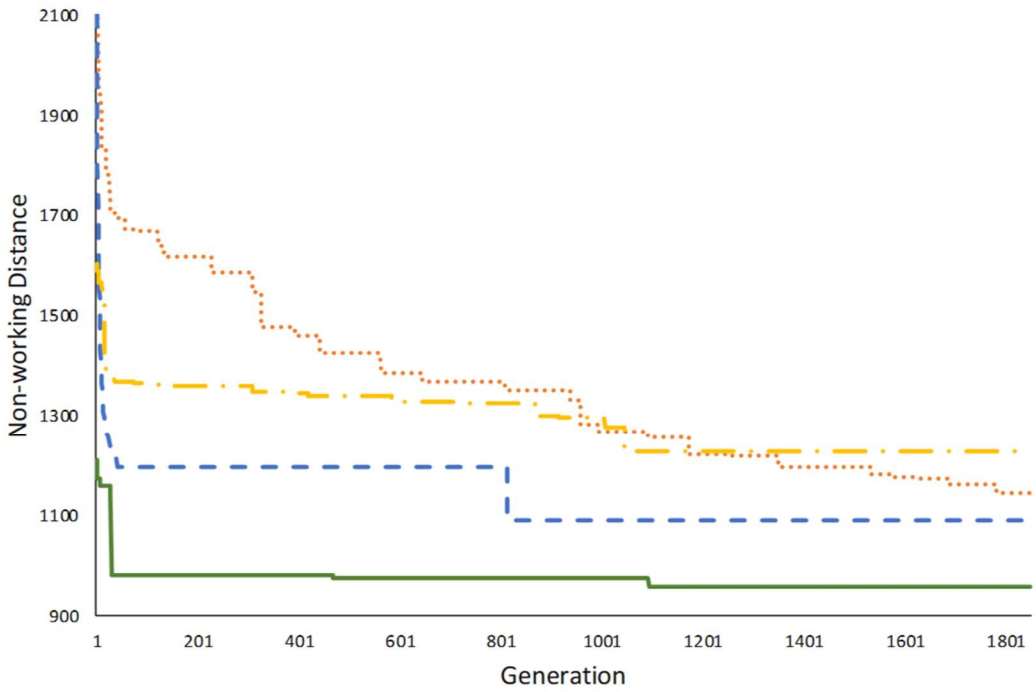


Figure 17. EHNS's manoeuvre results for the machine when solving Problem 74nc2

Figure 18 and 19 present the graphs of the convergence process of EHNS, GA, TS, and ACO when solving Problems 20rt (Figure 18 (a)), 37nc (Figure 18 (b)), 90rt (Figure 19 (a)), and 74nc2 (Figure 19 (b)). The X-axis of Figure 18 and 19 shows the number of generations of the algorithms, while the Y-axis represents the objective function reached.

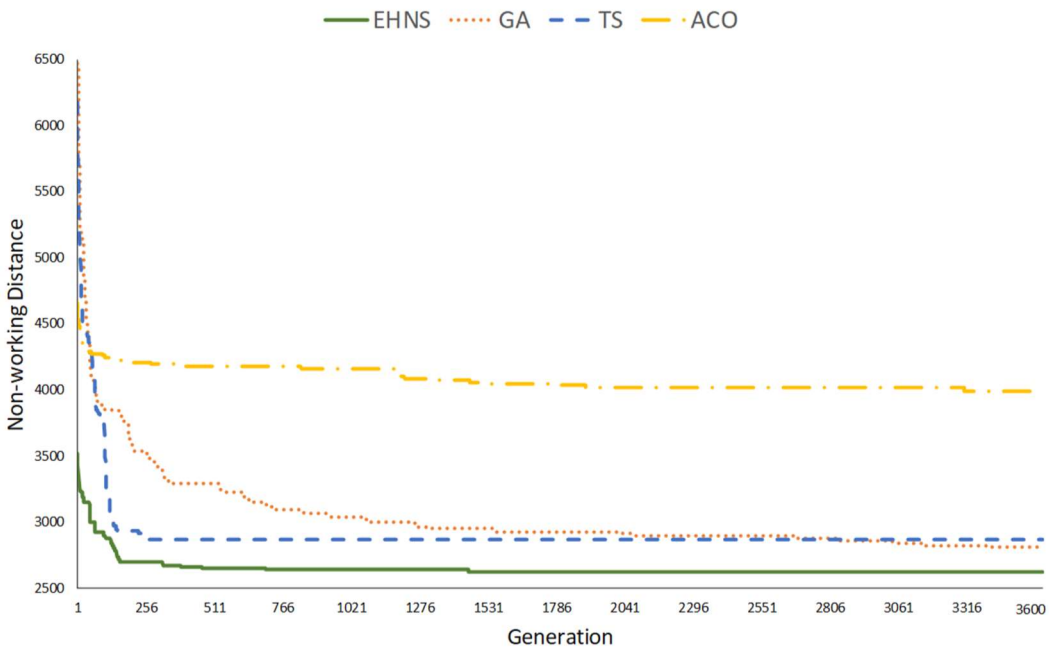


(a). Problem 20rt

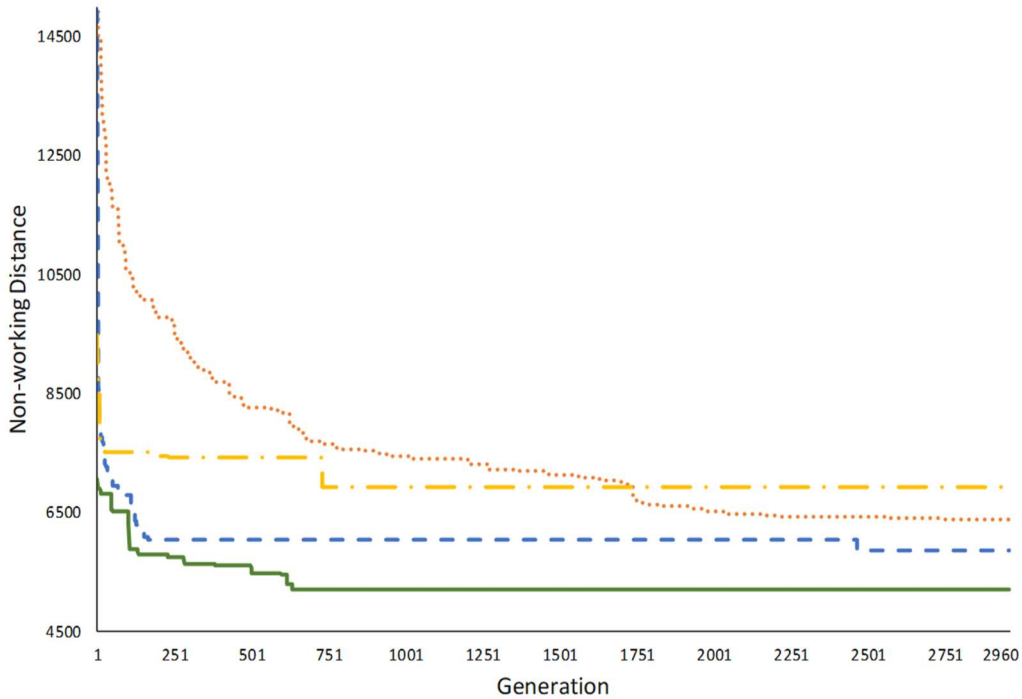


(b). Problem 37nc

Figure 18. Graphs of the convergence process of GA, TS, ACO, and EHNS solving Problems 20rt (a) and 37nc (b)



(a). Problem 90rt



(b). Problem 74nc2

Figure 19. Graphs of the convergence process of GA, TS, ACO, and EHNS solving Problems 90rt (a) and 74nc2 (b)

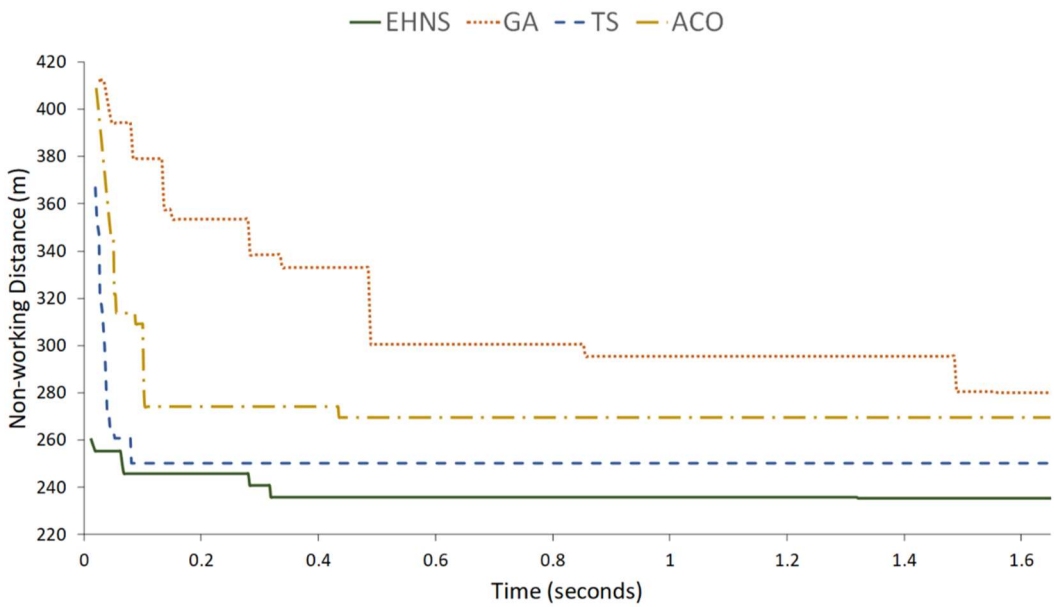
In the four problems presented in Figure 18 and 19, EHNS converges quickly to a good solution, and achieves the best objective function over all iterations compared to GA, TS, and ACO. TS is placed second in terms of solution quality, while GA's solution is placed third. ACO's objective function is better than GA's in

Problems 20rt and 37nc. Although the solutions by ACO show a fast improvement over the initial generations (in Problems 20rt, 90rt, and 74nc2), its final results are not as good as achieved by the GA over all generations.

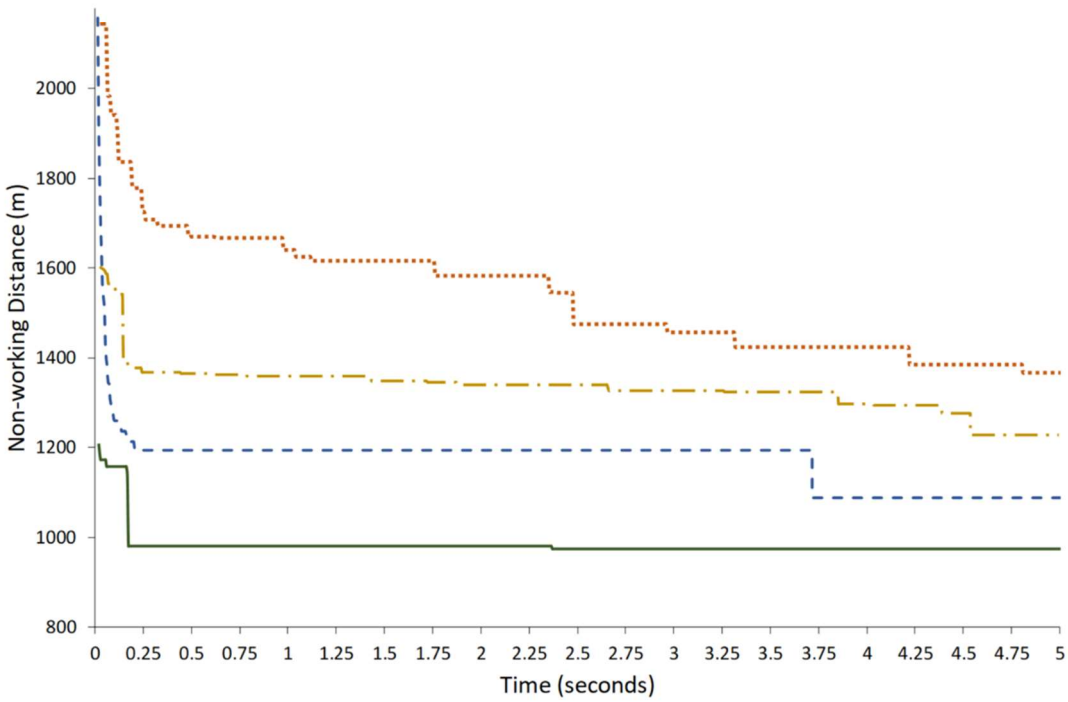
Figure 20 and 21 describe the graphs of the objective function (non-working distance) versus the time needed to achieve the objectives of EHNS, GA, TS, and ACO in four problem instances (20rt, 37nc, 90rt, and 74nc2). The X-axis of Figure 20 and 21 shows the time, while the Y-axis represents the objective function reached. The time is shortened to ACO's running time, which has the shortest running time. For all cases, EHNS achieves the best objective function in the fastest time compared with other algorithms, while TS is placed second.

As shown in Figure 20 (a) and Figure 21 (a, b), TS (the blue dashed line) is often trapped in local optima and produces the same solution when other algorithms are progressing to reach better solutions. TS does sometimes manage to achieve a better solution in the middle of its running (Figure 20(b)). In Figure 20 and 21, ACO (the yellow striped and dotted line) can produce good solutions first; however, its quality decreases, and it becomes trapped in the local optima. GA (the orange dotted line) slowly progresses to obtain better solutions from the beginning to the end of its generation. However, GA's objective function still cannot defeat TS's (Problems 20rt, 37nc, and 74nc2) and EHNS's object function (all problems).

The given running time of TS shown in Figure 20 and Figure 21 acts similarly to that shown in Figure 18 and Figure 19. In less than 150 generations (Figure 18(a)) and less than 0.4 seconds (Figure 20(a)), EHNS starts to converge while obtaining the best solution to Problem 20rt. EHNS always reaches a better solution than the others in Problem 37nc, and EHNS reaches its best solution before reaching 1200 generations (Figure 18(b)) in less than 2.5 seconds (Figure 20(b)). Solving Problem 90rt, EHNS starts to converge in about 500 generations (Figure 19(a)), and in less than 3 seconds (Figure 21(a)) while maintaining the best solutions compared with other algorithms. Finally, EHNS always achieves a superior solution and converges in about 700 generations (Figure 19(b)) with less than 6 seconds running time (Figure 21(b)).

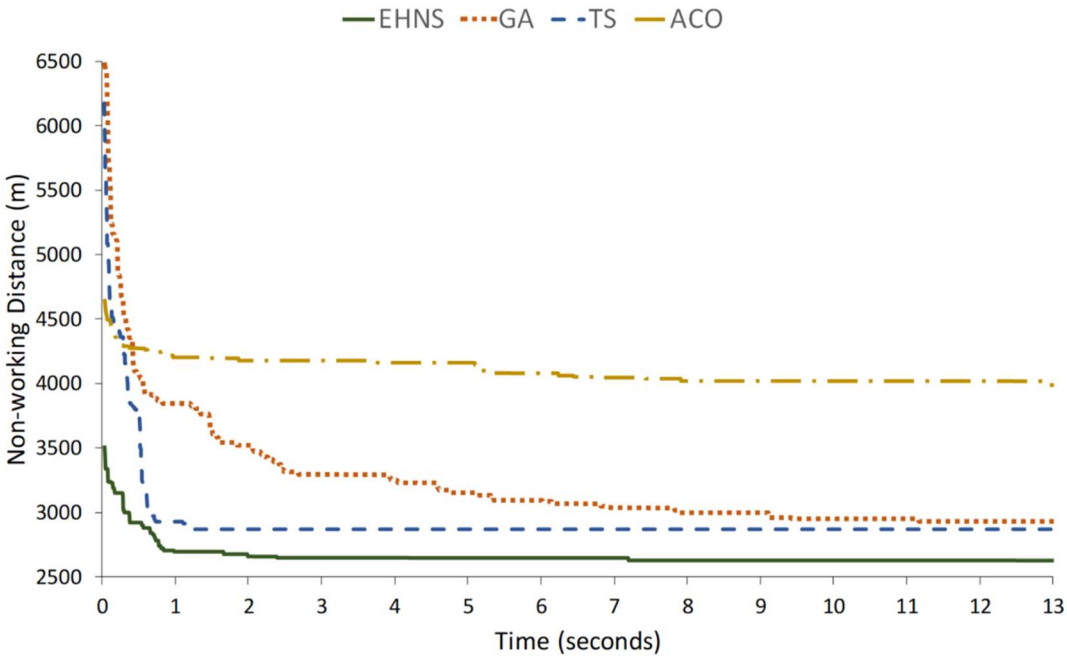


(a). Problem 20rt

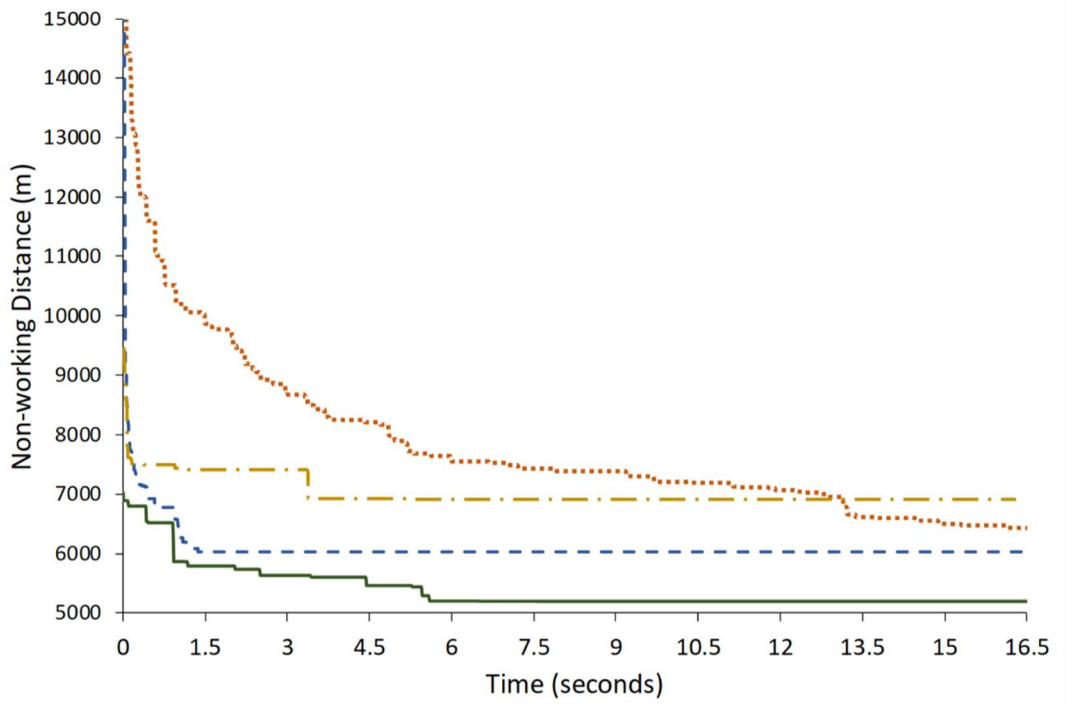


(b). Problem 37nc

Figure 20. Graphs of the objective function (non-working distance) versus the running time of GA, TS, ACO, and EHNS when solving Problems 20rt (a) and 37nc (b)



(a). Problem 90rt



(b). Problem 74nc2

Figure 21. Graphs of the objective function (non-working distance) versus the running time of all algorithms when solving Problems 90rt (a) and 74nc2 (b)

The robustness of EHNS is shown in solving the ARP datasets. EHNS successfully achieves the same optimal solution (as published literature) in four

problems, and introduce new best solutions in the rest of the instances. In all problems, the proposed algorithm (EHNS) always achieves a superior solution compare to GA, TS, and ACO.

4.6 Conclusion

The agriculture literature has introduced several cases of agricultural routing optimisations and provided solutions using different algorithms. This study collected different datasets representing past studies and suggest the application of a new algorithm (EHNS) that outperforms the others used so far. The algorithms used in studies from the literature are Genetic Algorithm, Mix-opt with Simulated Annealing, Tabu Search, and Ant Colony Optimisation.

The experimental results show that EHNS can obtain a better solution for 5 of 9 problems set while achieving the same best-known optimal solutions for the rest of the problem sets compared with the algorithms used in the previous literature. EHNS also successfully save the non-working distance by 10.68% in the improved cases. The results also illustrate that EHNS can maintain the best objective function and the fastest convergence speed compare with GA, TS, and ACO.

Chapter 5. Extensions to the ARP

As indicated in Section 2.5, there are several research gaps in the extension of ARP. First, there is limited research on ARP involving multiple fields and most ARP research utilises a single machine without capacity constraints. Hence, this study is the first to address several ARP constraints simultaneously. Second, the results of experiments conducted in this study show that the solutions given for several datasets in previous research can be improved. Therefore, an improved, more efficient algorithm needs to be developed to enhance the quality and management of the solution.

5.1 Problem Description

Figure 22 illustrates the problem with two fields. The blue line indicates the tracks while the red line is the field border. Each track is differentiated by a number (i.e. Field 1 has 18 tracks (west to east, 1 to 18)). The barriers in the problem are the field borders and the rocks, which cannot be traversed by the machine. Therefore, the machines can enter or exit a field, or move from one field to another only from a specific location or entrance.

In Figure 22, the machines can only enter the field through the field's entrance (green square). The machine's tour starts from and ends at the depot (red square). This research assumes that there are several available machines with different capacities to harvest several fields. The optimisation will assign machines to the given routes based on their capacities.

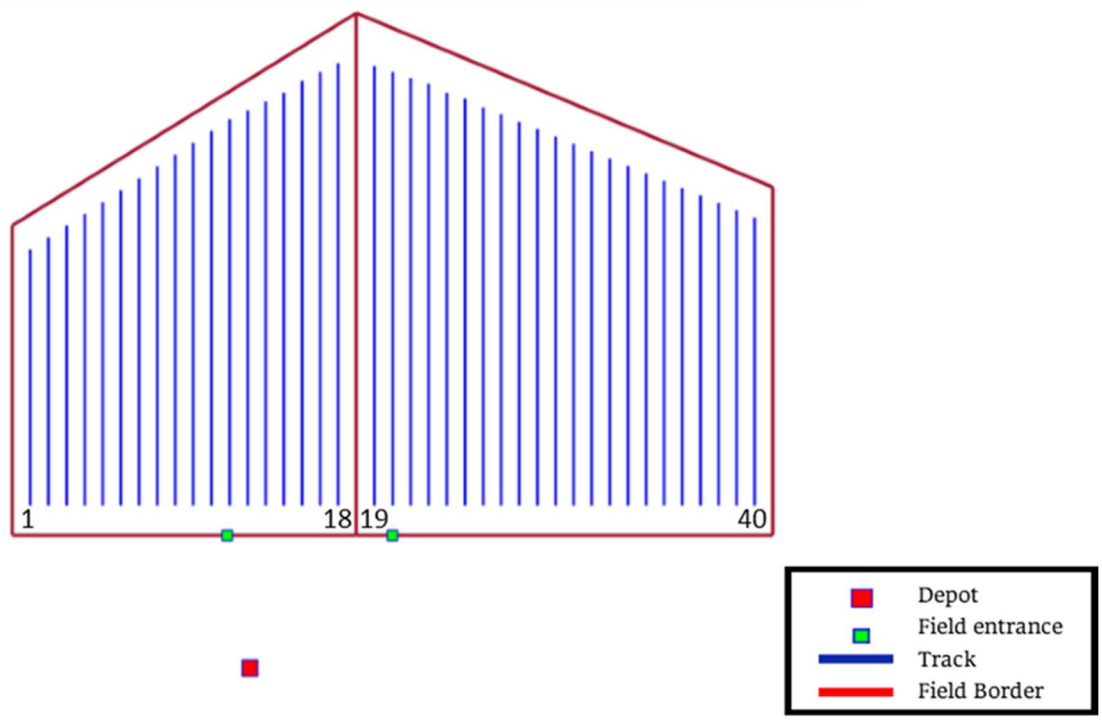


Figure 22. Illustration of a problem with two fields

Figure 23 depicts a machine's trip in two fields and its manoeuvres. There are two adjacent fields and the field in the north can only be entered through the field in the south. The green lines represent the non-working distance. The numbers adjacent to the green lines indicate the sequence of tracks the machine uses. The non-working distances can be differentiated by the manoeuvres (3 to 6 and 9) and the movements between depot and entrances, or the tracks to the entrances (1, 2, 7, 8, 10, 11, 12 and 13).

In Figure 23, the machine starts from the depot and traverses the south field's entrance (1) before harvesting several tracks in the southern field (2–6). The machine then goes to the northern field's entrance (7), harvests two tracks in the northern field (8 and 9) and returns to the entrance (10). Next, the machine harvests a track in the southern field (11) before going back to the depot via the southern field's entrance (12).

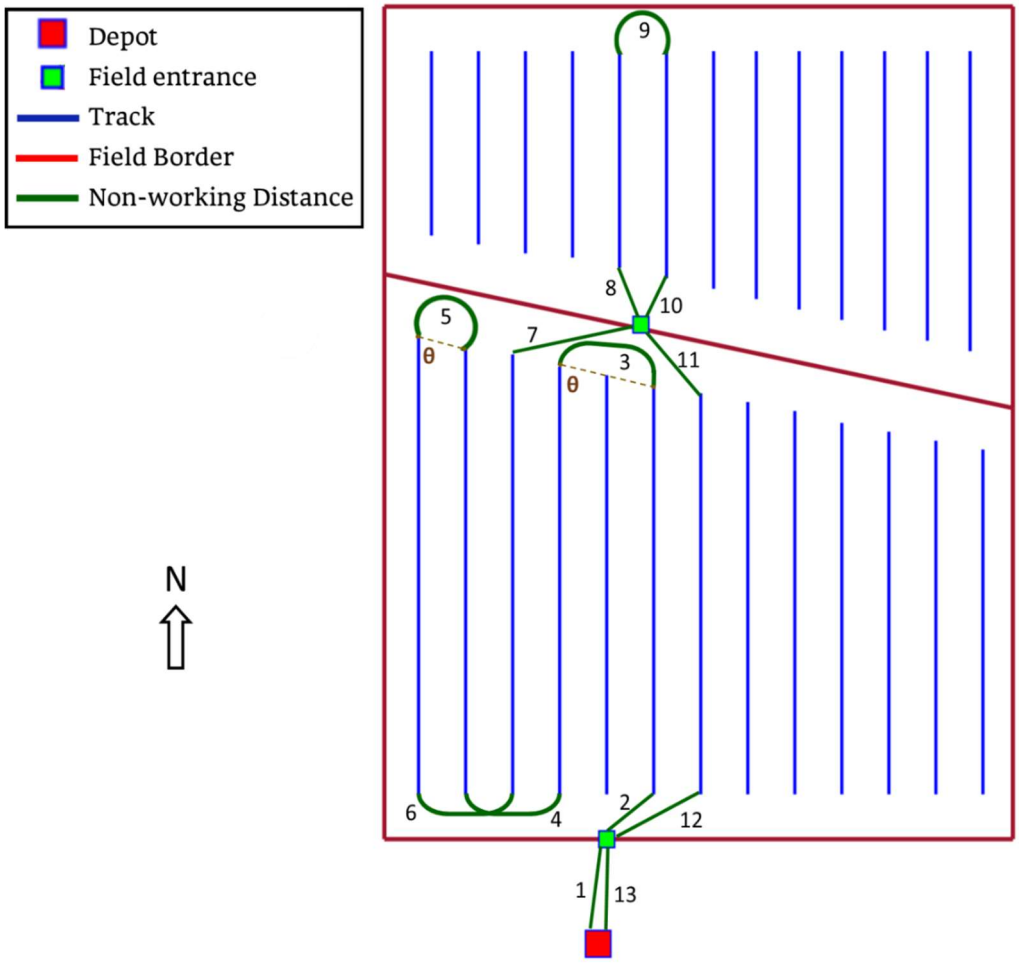


Figure 23. Illustration of a machine's tour and its manoeuvres in two fields

Figure 23 also depicts the four kinds of manoeuvres considered in this study. The assumption in Figure 23 is that the machine needs to skip at least one track to make a flat turn. Hence, if a machine's next track is directly beside the current track, then the bulb or bulbΘ turn will need to be made. In Figure 23, the manoeuvres are flat (4, 6), bulb (9), flatΘ (3) and bulbΘ (5). The flatΘ and bulbΘ turns are made when the current and next track are aligned at an angle θ ($\theta < 90^0$), while the flat and bulb turns are performed when the current and the next track are aligned with $\theta = 90^0$ and have the same Y-axis coordinate in the manoeuvre's area. The conditions of these manoeuvres are listed in Equation (3) in Section 5.2, which is adapted from Utamima et al. (2019) and Jin and Tang (2010).

5.2 Mathematical Model

The mathematical model used to solve the extended ARP is formulated as an integer programming model. The group of tracks in the fields are assumed to be a set of vertices and edges in Graph G. Table 15 describes the parameters used in the mathematical model. The decision variables are listed in Table 16.

Table 15. Parameters of the mathematical model

Notation	Description
G	Graph $G = \{V, E\}$ representing the tracks in the fields
V	Set of vertices in Graph G
S	A subgraph of Graph G, $\forall S \subseteq V$
E	Set of edges of Graph G
i,j	Vertex index (i, j = (0), 1, …, n), where 0 is the depot
F	Set of fields (f=1, …, F)
a,b	Fields index (a, b = 1, …, F)
T	Set of tracks (t=1, …, T) in the fields
δ_{ft}	The distance of track t to the entrance of field f
d_{ab}	The distance between two fields (from entrance field a to entrance field b)
D_f	The distance from field f to depot
h_{ab}	A binary variable, equal to 1 if there is a direct entrance from the field a to field b ($a, b \in F$), otherwise 0
l_{ft}	the length of track t in field f
K	The set of machines available at the depot, each referenced as k, $k \in K$
r	Minimum turning radius of a machine
w	Effective operating width of a machine
ψ	The driving direction of the machine, a binary variable equal to 0 and 1
$g_{ij\psi}$	A manoeuvre's degree between track i and j and driving direction ψ
$\theta_{ij\psi}$	The degree of tilt between the previous and next track, note that $0 < \theta \leq 90$
$\Omega(g_{ij\psi})$	The bulb type of manoeuvre turn executed by a machine in the headland area and

Notation	Description
	driving direction ψ
$\Pi(g_{ij\psi})$	The flat type of manoeuvre turn executed by a machine in the headland field area and driving direction ψ
$\Omega\theta(g_{ij\psi})$	The bulb type of manoeuvre turns with a $\theta_{ij\psi}$ degree. executed by a machine in the headland field area if the previous track and the next track are aligned with an angle $\theta_{ij\psi}$ and driving direction ψ
$\Pi\theta(g_{ij\psi})$	The flat type of manoeuvre turns with a $\theta_{ij\psi}$ degree, executed by a machine in headland field area if the previous track and the next track are aligned with an angle $\theta_{ij\psi}$ and driving direction ψ
Q_k	The capacity of a machine $k \in K$
P_{ft}	The volume of crops harvested in track t of field f

Table 16. Decision variables

Decision Variable	Description
x_{fij}^{k}	Equal to 1 if machine k moves from vertex i to vertex j in field f, otherwise 0 $(i, j \in E)$
y_{ab}	Equal to 1 if there is a tour from the field a to field b, otherwise 0 $(a, b \in F)$ (after harvesting field a the tour continues to b)
z_{ft}	Equal to 1 if there is a tour from the entrance of field f to track t, otherwise 0
s_f^k	Equal to 1 if machine k's first visited track is on field f, otherwise 0
τ_f^k	Equal to 1 if machine k's last visited track is on field f, otherwise 0

The objective function, given in Equation (1), is the distance aiming for a minimisation of the sum of all included edges in the solution, the distance travelled by each machine to and from the depot, the distance of the tour between fields, and the distance between the entrance to the first and last track to be harvested. The total distance travelled for harvesting the crop fields is found with Equation (2), which is the sum of the distance between the fields, the length of each track in each field, and the objective function.

The constraints of this objective function are determined by Equations (3) to (16). Equations (3) to (7) are used to compute the distance the machine travels to make a turning manoeuvre. Equations (8) and (9) specify that each route should start and end at the depot (Vertex 0). Equations (10) and (11) ensure that each node is visited only once. Equation (12) provides that when a machine enters a vertex, it will also leave that vertex. Equation (13) is a sub-tour limitation that eliminates any disjoint sub-tours from a solution. Equation (14) ensures that the amount of harvested crop does not exceed the capacity of a machine. Finally, Equations (15) and (16) ensure that each field is visited only once.

Where:

$$z = \text{Min}\left(\sum_{f\epsilon F}\sum_{i\in V}\sum_{j\in V}\sum_{k\in K} g_{ij}x_{fij}^{k} + \sum_{f\in F}\sum_{k\in K} D_f(s_f^k + \tau_f^k) + \sum_{a\in F}\sum_{b\in F} d_{ab}y_{ab}h_{ab} + \sum_{f\epsilon F}\sum_{t\in T} z_{ft}\delta_{ft}\right) \tag{1}$$

$$dist = \sum_{f\epsilon F}\sum_{t\epsilon T} l_{ft} + z \tag{2}$$

Subject to:

$$g_{ij} = \begin{cases} \Pi(i,j,\theta_{ij\psi}), if|i-j| \leq \frac{2\text{r}}{w} \wedge \theta_{ij\psi} = 90° \\ \Omega(i,j,\theta_{ij\psi}), if|i-j| > \frac{2\text{r}}{w} \wedge \theta_{ij\psi} = 90° \\ \Pi\theta(i,j,\theta_{ij\psi}), if|i-j| \leq \frac{2\text{r}}{w} \wedge \theta_{ij\psi} < 90° \\ \Omega\theta(i,j,\theta_{ij\psi}), if|i-j| > \frac{2r}{w} \wedge \theta_{ij\psi} < 90° \end{cases} \tag{3}$$

$$\Omega(\text{i},\text{j},\theta_{ij\psi}) = \text{r}\left(3\pi - 4\sin^{-1}\left(\frac{2\text{r}+|i-j|.w}{4r}\right)\right) \tag{4}$$

$$\Pi(i,j,\theta_{ij\psi}) = |i-j|w + (\pi-2)\text{r} \tag{5}$$

$$\Pi\theta(i,j,\theta_{ij\psi}) = |i-j|w(1+\cot\theta) + r(\pi-2) \tag{6}$$

$$\Omega\theta(i,j,\theta_{ij\psi}) = \pi r + \frac{4r^2 - |i-j|w(4r + w\cot^2\theta_{ij\psi} + w)}{4r-2w} \times \sin^{-1}\frac{|i-j|w(4r\cot\theta_{ij\psi} - 2w\cot\theta_{ij\psi})}{4r^2 - |i-j|w(4r + w\,\text{c}\quad{}^2\theta_{ij\psi} + w)} \tag{7}$$

$$\sum_{j\in V} x_{f0,j}^{k} = 1 \;\; \forall f \in \text{F} \tag{8}$$

$$\sum_{i \in V} x_{fi,0}^{k} = 1 \; \forall f \in \mathrm{F} \tag{9}$$

$$\sum_{k \in K} \sum_{i \in V} x_{fij}^{k} = 1 \; i,j \neq 0, \forall j \in V: i \neq j, \forall f \in \mathrm{F} \tag{10}$$

$$\sum_{k \in K} \sum_{j \in V} x_{fij}^{k} = 1 \; i,j \neq 0, \forall i \in V: i \neq j \; , \forall f \in \mathrm{F} \tag{11}$$

$$\sum_{k \in K} \sum_{i \in V} x_{fij}^{k} = \sum_{k \in K} \sum_{j \in V} x_{fji}^{k} \; \forall f \in \mathrm{F} \tag{12}$$

$$\sum_{i \epsilon S} \sum_{j \in S} x_{fij}^{k} \leq \|S\| - 1, \forall S \subseteq V, \|S\| \geq 1 \; \forall k \in K \; , \forall f \in \mathrm{F} \tag{13}$$

$$\sum_{f \epsilon F} \sum_{i \in V} \sum_{j \in V} P_{ft} x_{fij}^{k} < Q_k \quad \forall k \in K, \forall t \in \mathrm{T} \tag{14}$$

$$\sum_{a \in F} y_{ab} = 1 \; \forall b \in F: a \neq b \tag{15}$$

$$\sum_{b \in F} y_{ab} = 1 \; \forall a \in F: a \neq b \tag{16}$$

5.3 Proposed Algorithm Description

Figure 24 lists the Evolutionary Surrounding Search (ESS) pseudocode. ESS has two main stages: the exploration stage, which searches for the solution among the individuals in a population; and the exploitation stage, which searches for the best solution found so far. The individuals are the candidate solution to the problem, while the population refers to the group of individuals in the algorithm. ESS is a hybrid of three kinds of heuristics, namely, distribution search, surrounding search and local search.

First, the ESS parameters are initialised (see Figure 24). The parameter settings are explained in more detail in Section 4.2. The details of the machines (number, capacity and turning radius) are also initialised at this point. The algorithm reads the field data, which contains the track coordinates, the distance between each entrance and the depot, and the length of each track. Next, the individuals are generated and the objective function values are calculated. Then, the main iterations start with the exploration stage of ESS. The exploration stage will perform the distribution search according to its rate, otherwise the surrounding search will be executed.

The distribution search (Figure 24, Lines 5–7) will build the new population according to the probability of the order of tracks for the selected individuals. The selected individuals are the group of individuals with a better objective function,

which are labelled h_1, h_2, h_3, …, h_n ($h \in H$) (Chen et al., 2014). Equation (17) calculates the sum of every track being placed in an order. In Equation (17), $\Gamma^{h}_{t[p]}$ is set to 1 if track t is visited at order p, otherwise, it is set to 0 ($t \epsilon T, p \epsilon$P). The size of T and P is equal to the number of tracks in the fields. The distribution search function will update each individual in the population by selecting the order of track proportionally according to $\Re$.

Where:

$$\Re_{tp} = \sum_{h\epsilon} \sum_{t\epsilon T} \sum_{p\epsilon} \Gamma^{h}_{tp} \quad (17)$$

$$fn = \frac{1}{z} \quad (18)$$

The surrounding search procedure is listed in Figure 24, Lines 9–19. The procedure is adapted from Hansen et al. (2010) and Utamima et al. (2019). The procedure runs for some iterations. At the beginning of every iteration, the roulette wheel selection is performed, and an individual is selected (S). The roulette wheel selection expects that the better fitted an individual, the higher the chance of its survival. The fitness function that is used in the roulette wheel selection is listed in Equation 18, which is one per the objective function (Equation 1). The chance of selection in this approach is proportional to the fitness of an individual.

After selection, the algorithm chooses two points randomly from S and performs flip, guided swap, or inversion operators. The details of the operators are listed in Table 17. Next, the new population is updated from the results of surrounding search.

```
procedure ENDA()
    Initialization_Read_Data()
    OF ← Calculate_ObjectiveFunction()
    for g=1 to max_generation do
        if in DistributionSearch_rate then
            NewPopulation ← DistributionSearch(Selected_Individuals)
        else
                switch c do
                    case 1:
                        Flip(S,P)
                    case 2:
                        GuidedSwap(S,P)
                    case 3:
                        Inversion(S,P)
            end for
            NewPopulation ← Update_Population()
        end if
        OF ← Calculate_ObjectiveFunction(NewPopulation)
        currentBest ← Min(OF)
        if currentBest < globalBest then
            globalBest ← currentBest
        else
                ndSol ← LocalSearch()
        end if
        if ndSol < globalBest then
            globalBest ← ndSol
        end if
        Elitism()
    end for
end procedure
```

Figure 24. The pseudocode of ESS

The objective function calculation evaluates the fitness of the new population and finds the current best solution (*cBest*) in the population. This solution is checked to determine whether it can replace the global best solution (the best solution found so far among generations). If *cBest* is not better than the global best solution (*globalBest*), then the second stage of ESS is implemented. In this stage, the surrounding search searches inside *globalBest* to find a better solution. The procedure is adapted from the Local Search procedure in Song et al. (2019). In this stage, the interchange and reverse operators are employed until no further improvement can be made. The operators are described in Table 17.

If the solution of the surrounding search (*ndSol*) is better than *globalBest*, then *ndSol* will become the *globalBest.* Next, the Elitism procedure is run to retain 10% of the best individuals through each generation to protect prospective solutions (Nouri & Ladhari, 2018). The recorded individuals are copied to the new population and are sent to the next generation.

Table 17. Description of operators used in ESS

No.	Operators	Description
1	Flip	Flips the order of tracks between two points
2	Guided Swap	Choose a track, then find its position in the global best solution. Place the selected track with the same global best position by swapping with the other track in that position
3	Inversion	Choose two points, then move the first point to the second point and slide forward the remaining points (starting from the first point + 1 until the second point)
4	Interchange	Interchange two elements randomly
5	Reverse	Choose two points, then reverse the tracks between the two points

5.4 Experimental Results and Analysis

5.4.1 Problem Sets Description

The experiments conducted in this study are comprised of two steps. The first step involves the validation of the proposed algorithm by using the data in the literature. The datasets are gathered from publications of studies and by contacting the corresponding authors. Table 18 shows a description of the data. In Table 18, the first column lists the instance code and the second column details the total number of tracks in that instance; the track widths and lengths (metres), turning radius (metres), shape and machine used are listed in columns 3 to 6; and the final column contains the references to the problem instances.

Table 18. Description of previous datasets

Problem Code	Total Tracks	Track Width & Length (metres)	Turning Radius (metres)	Shape	Machine	Reference
8rt	8	2.89 & 30	3.50	Rectangular	1 (single)	Bochtis & Vougioukas (2008)
12rta	12	2.5 & 40	3.50			Conesa-Muñoz et al. (2016a)
12rtb	12	2.5 & 70	3.50			
20rt	20	2.5 & 80	3.50			
37nc	37	9 & vary	6.00	Non-convex		Conesa-Muñoz et al. (2016a); Hameed, Bochtis, & Sørensen (2011)
74nc	74	19 & vary	9.77	Non-convex	3 (multiple and homogeneous)	Seyyedhasani & Dvorak (2018a)

In the second step, experiments are run using the extended ARP datasets. Table 19 gives the attributes of the datasets used in this study. There are seven datasets with multiple fields and different shapes. The problems are generated and vary in terms of the total number of tracks, width (metres) and turning radius (metres). The number of fields ranges from two to five, with total tracks ranging from 30 to 112. The machines used have different capacities (ranging from 700 to 3,000 metres), as detailed in Table 20. As indicated earlier, this study assumes a uniform measure of harvest per travelled distance. Hence, the machine capacity is depicted with the maximum distance it can travel on one trip.

Table 19. Description of the extended ARP problem set

No.	Problem Code	Number of Fields	Total Tracks	Track Width (metres)	Turning Radius (metres)	Shape
1	2A	2	30	8	7.5	Non-convex
2	2B	2	36	8	7.5	Non-convex
3	3A	3	56	7	5.5	Non-convex and Rectangular
4	3B	3	62	6	4.8	Non-convex
5	4A	4	78	6	4.8	Non-convex and Rectangular
6	4B	4	100	5	3.9	Non-convex and Rectangular
7	5A	5	112	5	3.9	Non-convex

Table 20. Machines provided

Machines	Maximum distance when harvesting	Availability
1	3,000	1
2	2,500	3
3	2,200	3
4	700	1

Figure 25 illustrates the fiveproblem instances. The number in the middle of each field is the field number, and the small number near the track is the track number. These numbers are used to give a better understanding of the output of the program, shown later in Table 25.

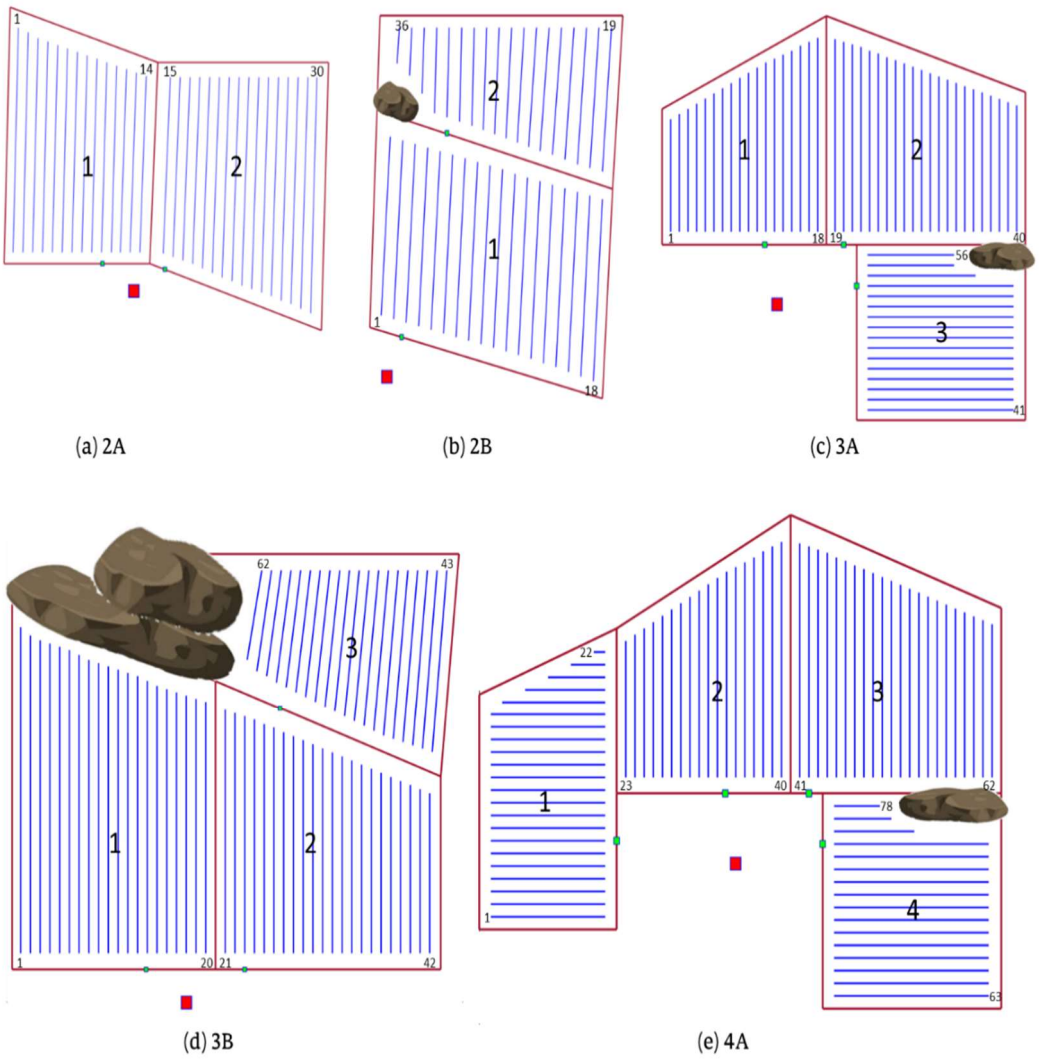


Figure 25. Illustration of extended ARP problem set

5.4.2 Parameter Settings

The parameter set-up is acquired by directing a two-level factorial plan with four elements. Each factor joins high and low dimensions (Montgomery, 2013). The algorithm is run ten times to make up for the non-deterministic nature of the heuristics (Guan & Lin, 2016). Table 21 lists the factors and the levels of each set-up. To adjust to the extent of the problem, the settings use *n*, which is the number of tracks in the fields. A larger problem requires a larger population and more generations. The value shown in bold indicates a better setting for ESS.

This study also builds on GA and TS to solve the extended ARP problem set. GA and TS are chosen because both algorithms have been used recently in

optimisation works in the agriculture domain (Florentino et al., 2018; Seyyedhasani & Dvorak, 2018). The selected settings for the number of generations and the size of the population are also applied to GA and TS.

Table 21. Factor settings for ESS

Factors	Levels		Range
	Low	High	
Generations	*40n*	***50n***	*30n-50n*
Population size	*4n*	***5n***	*3n-5n*
Surrounding search iteration	*2n*	***2.5n***	*2n-2.5n*
Distribution search rate	*4n*	***5n***	*4n-5n*

5.4.3 Numerical Results

Table 22 shows the details of the non-working distance of six published datasets. In Table 22, the first and second columns show the problem code and the associated references. The third column is the best-known solution to date obtained from the latest publication (Conesa-Muñoz et al., 2016a; Seyyedhasani & Dvorak, 2018), while the last column shows the results obtained by ESS. The bold values in the third and fourth columns refer to the minimum value in that row. As shown by the results, ESS can obtain the same solution for four instances (8rt, 12rta, 12rtb, 20rt) and achieves better solutions in two instances (37nc, 74nc). These results indicate that ESS can achieve the expected results for ARP instances.

Table 22. Test with published data (single field)

Problem	Best-Known Solution	Proposed Algorithm: ESS	References
8rt	**94.439**	**94.439**	Bochtis & Vougioukas (2008); Conesa-Muñoz et al. (2016a)
12rta	**146.027**	**146.027**	Bochtis & Vougioukas (2008); Conesa-Muñoz et al. (2016a)
12rtb	**145.602**	**145.602**	Bochtis & Vougioukas (2008); Conesa-Muñoz et al. (2016a)
20rt	**235.491**	**235.491**	Bochtis & Vougioukas (2008); Conesa-Muñoz et al. (2016a)
37nc	961.470	**958.930**	Conesa-Muñoz et al. (2016a); Hameed et al. (2011)
74nc	4416.300	**3869.590**	Seyyedhasani & Dvorak (2018a)

Table 23 presents the results of the experiments conducted to calculate the non-working distance of multiple field data with multiple machines of the extended ARP data. In Table 23, the first column lists the problem number and the second the problem code, the third column specifies the number of tours required to complete all tracks in the fields, and the fourth, fifth and sixth columns list the solutions obtained by ESS, GA and TS, respectively. The values in bold refer to the minimum value in that row. The last column is the average distance savings obtained by ESS compared to those of GA and TS for every problem, while the last row in bold is the average of the distance savings for all problems. ESS successfully achieves the shortest non-working distance for all problem sets compared to the performances of GA and TS. The GA solutions are better than TS in four instances (Problems 2, 4, 6 and 7), while the TS solutions for the remaining three cases (Problems 1, 3 and 4) are better than those obtained by GA. In terms of average distance savings, ESS successfully saves 11.72% of the non-working distance compared to other algorithms.

Table 23. Test with extended ARP problem set (multiple fields)

No.	Problem Code	Number of Tours	GA	TS	Proposed Algorithm: ESS	Distance Savings
1	2A	2	1051.118	1029.329	**955.869**	8.10%
2	2B	2	1101.292	1279.074	**1044.038**	11.79%
3	3A	3	1121.185	1013.578	**967.872**	9.09%
4	3B	3	2018.031	2333.044	**1909.22**	11.78%
5	4A	4	1776.730	1645.868	**1504.245**	11.97%
6	4B	6	2923.140	3063.433	**2605.433**	12.91%
7	5A	6	4145.357	4782.591	**3711.986**	16.42%
					Average:	**11.72%**

Table 24 shows the running time of all the algorithms for each problem set. The running time of ESS is the fastest compared to those of GA and TS. On the other hand, the runtimes of TS are faster than GA in five instances, while those for problems 2 and 4 are slower than GA.

Table 25 lists the details of the ESS solution for each problem set. In Table 25, the first column lists the problem number and the second the problem code, the third column shows the tours and the maximum in-track distance of the machine, and the last column is the order of tracks in the fields and lists the field number and the order of the tracks that correspond to Figure 26. For example, Problem 1 needs two machines (with maximum distance inside the tracks of 3,000m and 2,500m, respectively) to harvest the fields. The first machine goes to Field 1 and harvests 14 tracks in the order 10, 8, 6, 4, 2, 1, 3, 5, 7, 9, 13, 14, 12 and 11 and then goes to Field 2 and harvests four tracks in the order 29, 27, 28 and 30. The second machine then goes to Field 2 and harvests 12 tracks in the order 24, 26, 25, 23, 22, 20, 21, 19, 18, 16, 17 and 15. Note that every machine must start from the depot and return to the depot.

Table 24. The running times of ESS, GA and TS for all problem sets

No.	Problem Code	Number of Tours	Average Running Time (seconds)		
			GA	TS	ESS
1	2A	2	1.388	1.359	**0.807**
2	2B	2	6.842	9.171	**4.565**
3	3A	3	4.142	2.948	**2.179**
4	3B	3	12.264	16.986	**7.974**
5	4A	4	9.527	7.459	**5.283**
6	4B	6	25.608	23.814	**11.942**
7	5A	6	26.438	25.629	**13.663**

Table 25. The details of every tour in each problem set obtained by ESS

Problem	Tours (Max in-track distance)	Order of Tracks in Fields (Field Number [tracks' order])
2A	1 (3000)	Field 1 [10, 8, 6, 4, 2, 1, 3, 5, 7, 9, 13, 14, 12, 11]; Field 2 [29, 27, 28, 30]
	2 (2500)	Field 2 [24, 26, 25, 23, 22, 20, 21, 19, 18, 16, 17, 15]
2B	1 (3000)	Field 1 [2, 1, 3, 6, 10, 11, 8]; Field 2 [33, 35, 36, 34, 32, 30, 28, 26, 24, 22, 21, 19, 20, 23, 25, 27, 29, 31]; Field 1 [4]
	2 (2500)	Field 1 [5, 7, 9, 12, 13, 14, 15, 16, 17, 18]
3A	1 (3000)	Field 1 [13, 15, 17, 18, 16, 14, 12, 11, 8, 7, 4, 3, 1, 2, 5, 6, 9, 10]; Field 2 [40, 39, 36, 35, 37, 38]
	2 (2500)	Field 2 [29, 31, 33, 34, 32, 30, 28, 27, 25, 23, 20, 19, 21, 22, 24, 26]; Field 3 [54, 56, 55, 53, 50, 52]
	3 (2500)	Field 3 [51, 49, 47, 45, 48, 46, 44, 42, 41, 43]
3B	1 (3000)	Field 1 [13, 11, 9, 7, 5, 3, 1, 2, 4, 6, 8, 10, 12, 14, 16, 15]
	2 (2500)	Field 1 [17, 18, 20,19]; Field 2 [25]; Field 3 [52, 50, 51, 48, 49, 47, 45, 43, 44, 46, 42, 41, 39, 40]; Field 2 [23, 22, 21]
	3 (2500)	Field 2 [24]; Field 3 [57, 54, 53, 55, 56, 58, 60, 62, 61, 59]; Field 2 [26, 28, 27, 29, 30, 32, 31, 36, 35, 37, 38, 34, 33]

Problem	Tours (Max in-track distance)	Order of Tracks in Fields (Field Number [tracks' order])
4A	1 (3000)	Field 1 [5, 3, 1, 2, 4, 6, 8, 10, 12, 14, 16, 18, 21, 19, 22, 20, 17, 15, 13, 11, 9, 7]; Field 2 [33, 34, 36, 38, 40, 39, 37, 35]
	2 (2500)	Field 2 [32, 31, 29, 30, 28, 27, 24, 23, 25, 26]; Field 3 [61, 62, 59, 60, 58, 57, 55, 56, 54, 53]
	3 (2500)	Field 3 [51, 52, 50, 49, 47, 48, 46, 45, 43, 44, 42, 41]; Field 4 [74, 72, 69, 66, 64, 67, 70, 77, 78, 76]
	4 (700)	Field 4 [75, 73, 71, 68, 65, 63]
4B	1 (3000)	Field 1 [1, 5, 7, 12, 15, 19, 22, 25, 27, 29, 30, 28, 26, 24, 21, 17, 13, 11, 9, 3]
	2 (2500)	Field 1 [2, 4, 6, 8, 10, 14, 16, 18, 20, 23]; Field 2[37, 40, 42, 41, 38]; Field 3 [70, 72]; Field 2[32]
	3 (2500)	Field 2 [31, 33, 35]; Field 3[74, 76, 75, 73, 71, 68, 66, 64, 61, 59, 57, 55, 56, 58, 60, 62, 63, 65, 67, 69]; Field 2 [34]
	4 (2200)	Field 2 [36, 44, 46, 48, 50, 52, 54, 53, 51, 49, 47, 45, 43, 39]; Field 4[98, 100, 99, 97, 94, 96]
	5 (2200)	Field 4 [92, 90, 87, 85, 83, 81, 79, 77, 78, 80, 82, 84, 88, 95]
	6 (700)	Field 4 [93, 91, 89, 86]
5A	1 (3000)	Field 1 [28, 21, 19, 17, 15, 13, 11, 9, 7, 5, 3, 1, 2, 4, 6, 8, 10, 12, 14, 16, 18, 30]
	2 (2500)	Field 1 [31, 34, 32, 33, 29, 27, 25, 23, 20, 22, 24, 26]; Field 2 [47, 45, 43, 44, 46, 48]
	3 (2500)	Field 2[49, 51, 53, 54, 52, 50, 42, 41, 39, 37, 35, 36, 38, 40];
	4 (2200)	Field 3[59]; Field 4 [92, 94, 93, 96, 95, 90, 89, 87, 85, 82, 80, 78, 77, 79, 81, 83, 84, 86, 88, 91]; Field 3[58]
	5 (2200)	Field 3[62, 64, 66, 68, 70, 72, 74, 76, 75, 73, 71, 69, 67, 65, 60, 57]
	6 (2200)	Field 3[55, 56, 61, 63]; Field 5 [112, 110, 108, 106, 104, 102, 100, 98, 97, 99, 101, 103, 105, 107, 109, 111]

Figure 26 illustrates the manoeuvres of the machines in Problem 7. The field number is in the centre of each field, and the track numbers are indicated by the

small number in the corner of each field. Different coloured lines represent each of the six tours of the machines. In this problem, the turning radius (*r*) is 3.9 metres, and the track width (*w)* is 5 metres. Thus, according to Equation (3) in Section 3.4, the machine can perform a flat turn by skipping at least one track to go to the next track. As shown in Figure 26, the predominant type of manoeuvre is the flat turn.

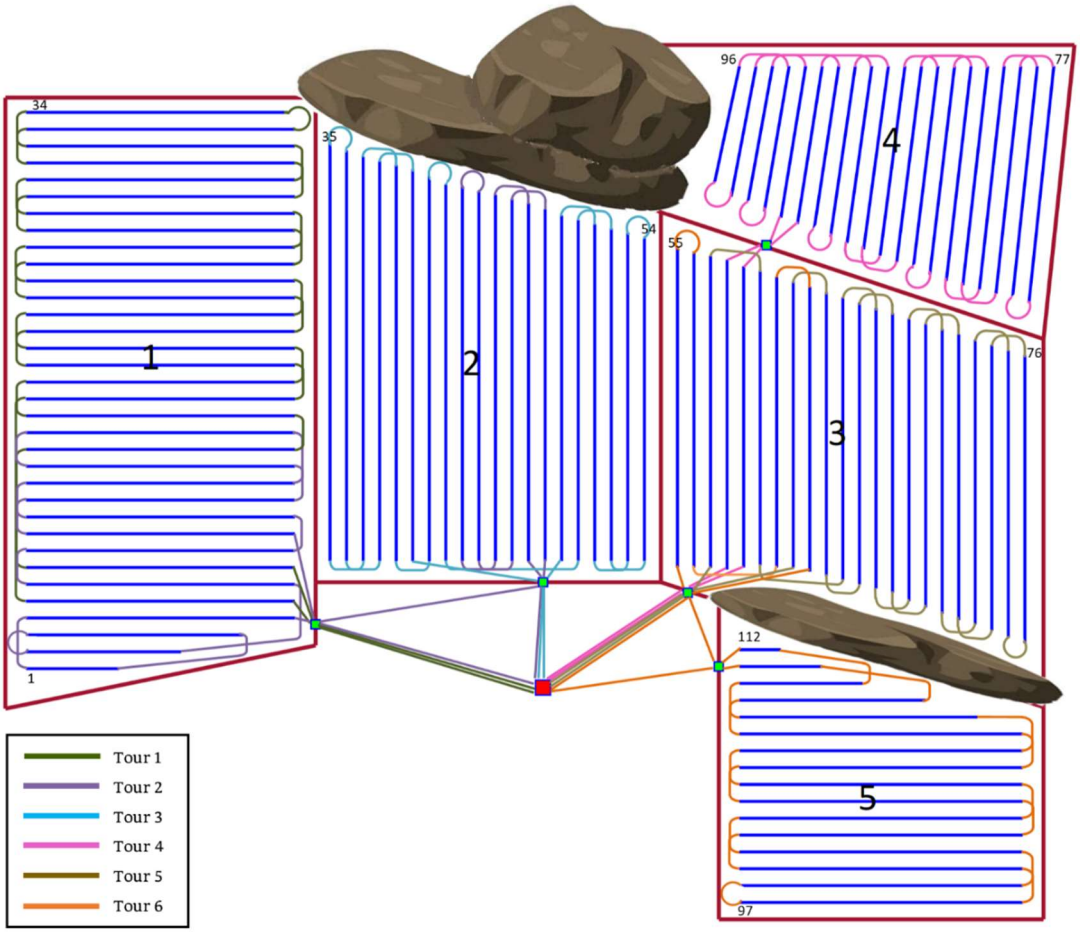


Figure 26. Illustration of the tours and the manoeuvres in five fields (Problem 7)

5.4.4 Validation of the Mathematical Model

To validate the mathematical model, it was implemented using the General Algebraic Modeling System (GAMS) software. Two small problems are tested, each consisting of multiple fields and multiple machines with different capacities. Each problem involves two fields with tracks measuring 4 metres in width, with two machines available with a turning radius of 3 metres.

Table 26 shows a comparison of the results of GAMS and ESS (coded with MATLAB). Both GAMS and ESS obtain the same solution for the two problems as well as the same tours of Machine 1 and Machine 2.

Table 26. Comparison of GAMS and ESS

Problem	GAMS			ESS		
	Solution (m)	Time (s)	Tour	Solution (m)	Time (s)	Tour
A	179.713	0.365	1: Field 1 [3, 1, 4, 2]	179.713	0.052	1: Field 1 [3, 1, 4, 2]
			2: Field 2 [6, 5]			2: Field 2 [6, 5]
B	225.413	0.449	1: Field 1 [5, 3, 1, 2, 4, 6]	225.413	0.105	1: Field 1 [5, 3, 1, 2, 4, 6]
			2: Field 2 [9, 7, 10, 8]			2: Field 2 [9, 7, 10, 8]

The experiments show that the running time of ESS, on average, is five times faster than that of GAMS. If the size of the problem is increased, the running time is expected to grow exponentially for the GAMS implementation. Therefore, the utilisation of ESS is required for both small and large problems because it can achieve a good solution with a faster running time.

5.5 Conclusion

The scope of this chapter comprises ARP for a harvesting problem with the tracks inside several fields on a flat surface. This chapter presents the extended ARP by considering multiple constraints simultaneously to generalise the harvesting problem effectively. Furthermore, it recommends the application of a new hybrid algorithm (ESS) that outperforms GA and TS. The experimental results demonstrate that during the validation process with the previous ARP dataset, ESS achieves the same optimal solution in four instances, and produces a better solution for the other two cases. ESS also achieves the fastest running time for all the given problems.

Chapter 6. Evolutionary Algorithms for ARP

Several variations of evolutionary algorithms have been published or built by the author during the PhD study. The main five evolutionary algorithms developed are:

1. Evolutionary Hybrid Neighbourhood Search (EHNS); published in *Biosystems Engineering* (Utamima et al., 2019c).
2. Evolutionary Estimation of Distribution Algorithm (EEDA); accepted for the Information Systems International Conference 2019 and published in *Procedia Computer Science* (Utamima et al., 2019b).
3. Lovebird Algorithm; accepted and published for the Australasian Conference in Information Systems 2019 (Utamima et al., 2019a).
4. Evolutionary Lovebird Algorithm (EvoLovebird); accepted for the 12th International Conference on Computer and Automation Engineering 2020 and indexed in ACM Digital Library (Utamima et al., 2020).
5. Evolutionary Surrounding Search (ESS); accepted for second round of review in *Annals of Operations Research*.

This chapter overviews the evolutionary algorithms and tests EHNS, EvoLovebird and ESS on ARP problem sets.

6.1 Input and Solution Representation of All Evolutionary Algorithms

The inputs for the evolutionary algorithms are the field details and machine information, specifically:

- number of tracks
- number of fields (if multiple fields)
- track width
- track length

- track coordinates
- depot coordinates (if available)
- turning radius of the machine
- number of machines (if using multiple machines)
- capacity of the machine (if considered in the problem).

Each candidate solution is represented as permutation numbers, as illustrated in Figure 27. Each track is labelled by a number, and the sequence indicates the order in which a machine will visit the tracks. For instance, in Figure 27, the machine will go to track 4 right after it visits track 1.

Sequence :	1	2	3	4	5	6	7	8
Track :	1	4	7	3	6	2	5	8

Figure 27. Representation of a candidate solution

6.2 Evolutionary Hybrid Neighbourhood Search

This section overviews the general stages of the Evolutionary Hybrid Neighbourhood Search (EHNS), as illustrated in Figure 28. A detailed explanation of EHNS is given in Section 4.4.

In the first process, the parameters and variables of EHNS are initialised. The parameters are the number of maximum generations, population size, neighbourhood search iteration and TS iteration. The variables are determined from reading the input of the problem set. Candidate solutions are generated and objective function values are calculated. The main loop is repeated until the *maxGen* iteration is reached (maximum number of generations). In the main iteration, EHNS will execute mutation-based neighbourhood search, replacement of candidate solution, TS, best solution updating and evolutionary strategy. TS will only be performed if *cBest* is not better than *prevBest* (the previous global best solution). The evolutionary strategy uses Elitism (to preserve the best solutions through generations) and Scramble (to mix several candidate solutions).

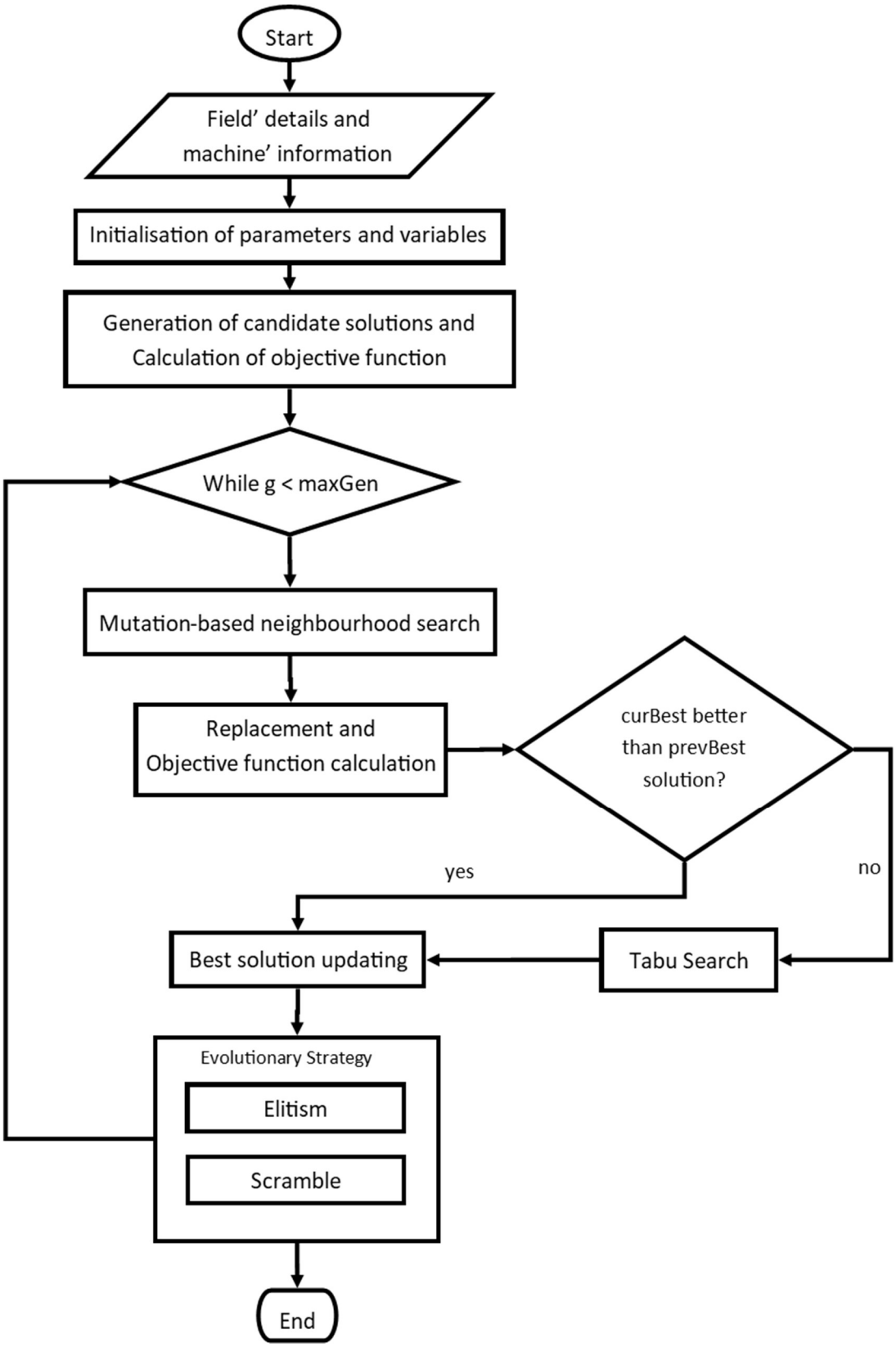


Figure 28. General flowchart of the stages of EHNS

6.3 Evolutionary Estimation of Distribution Algorithm

The Estimation of Distribution Algorithm (EDA) is a stochastic optimisation method that reproduces new candidate solutions from the evaluation of the probabilistic distribution of the population to achieve a more efficient search. EDA builds and samples the probabilistic models from the group of promising candidate solutions. It explores the space of potential solutions by exploiting the inter-variable dependency and sampling of promising candidate solutions. EDA constructs a probabilistic model to get the parental distribution and samples new solutions. The sampling from probabilistic models avoids the disruption of partial dominant solutions (Hauschild & Pelikan, 2011).

Most metaheuristic algorithms generate random candidate solutions to process into their new generation. In contrast, EDA uses the probabilistic models to create a group of new candidate solutions. This approach enhances the solution quality of EDA because the produced offspring are statistically built and sampled based on a probability model from a group of parents' best fitness. EDA, as a tool for evolutionary computation, then becomes a promising algorithm that can competitively solve the optimisation problem (Gao & de Silva, 2016, 2018).

Using EDA it is possible to construct a model of the promising areas of the search space and to use this model to direct the search for the optimum solution. Modelling is accomplished by constructing a probabilistic graphic model that depicts a simplified representation of the features shared by the selected solutions.

The Evolutionary Estimation of Distribution Algorithm (EEDA) proposed in this study is an extended version of EDA that hybridises the EDA with Permutation Neighborhood Search and Elitism as an evolutionary strategy. Each individual in the EEDA population is represented as a permutation of track numbers in the field. The sequence of track numbers indicates the order in which a machine traverses them. The general stages of EEDA are illustrated in Figure 29.

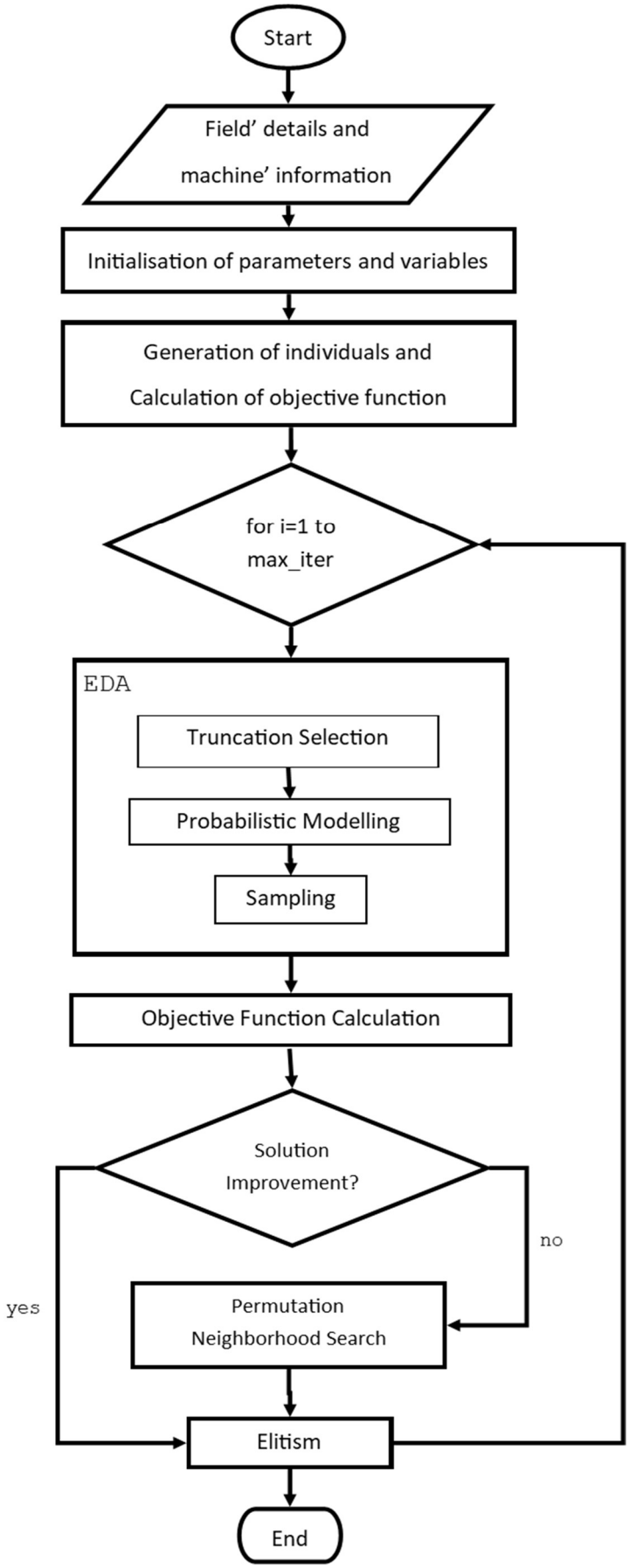


Figure 29. General flowchart of the stages of Evolutionary EDA

The initial population of EEDA contains randomly generated individuals. The algorithm loops for several iterations and starts with truncation selection, where the top 50% of the population are chosen (Chen et al., 2014) . The chosen individuals are identified as $h_1, h_2, h_3, \ldots, h_n$ ($h \in H$) with n being half the size of the population. Next, the probabilistic model is applied, reflecting the importance of each track in the sequence.

The final stage of EEDA is the sampling procedure, which builds the new population. The pseudocode of the sampling method is shown in Figure 30. The procedure assigns a track to be placed in the first position (Figure 30, Line 3). This assignment is chosen randomly from the first track of the category of chosen individuals. The process is repeated until the last track has been allocated. The roulette wheel selection is adapted (Figure 30, Lines 4–7) to place the remaining tracks in the order.

```
1  Procedure Sampling (Prob_Model, Selected_Individuals) {
2    for i=1 to Population_Size {
3     Population[i,1] = RandSelect(Selected_Individuals[1])
4     for j=2 to Tracks_Size {
5       P ← Cumulative_probability(Prob_Model)
6       Population [i,j] = RouletteWheel(P, Available_Tracks)
7     }
8    }
9  }
```

Figure 30. The sampling procedure of EDA

The objective function of each individual is then calculated. The EEDA algorithm then reviews *cBest* to see if it is better than *globalBest*. If no improvement is made, EEDA will perform the Permutation Neighborhood Search. Otherwise, it goes directly to the Elitism strategy. Permutation Neighborhood Search is a subset of neighbourhood search and mutation operators, which focuses on combination optimisation (Guan & Lin, 2016). The algorithm uses the swap, flip, slide and insertion operators used in permutation encoding. Elitism records the best individuals over generations to secure prospective individuals.

6.4 Lovebird Algorithm

The Lovebird Algorithm (Lovebird) adapts combinatorial operators to the generation of offspring (new candidate solution). Figure 31 illustrates the stages of the algorithm. The initialisation process of Lovebird defines the parameters (maximum iterations and population size) and the variables. Candidate solutions (birds) are generated and objective function values are calculated. The stop criterion is the *max_iter*, which is the maximum number of iterations in the algorithm. The maximum value is set to 50*n* (where *n* = number of fields).

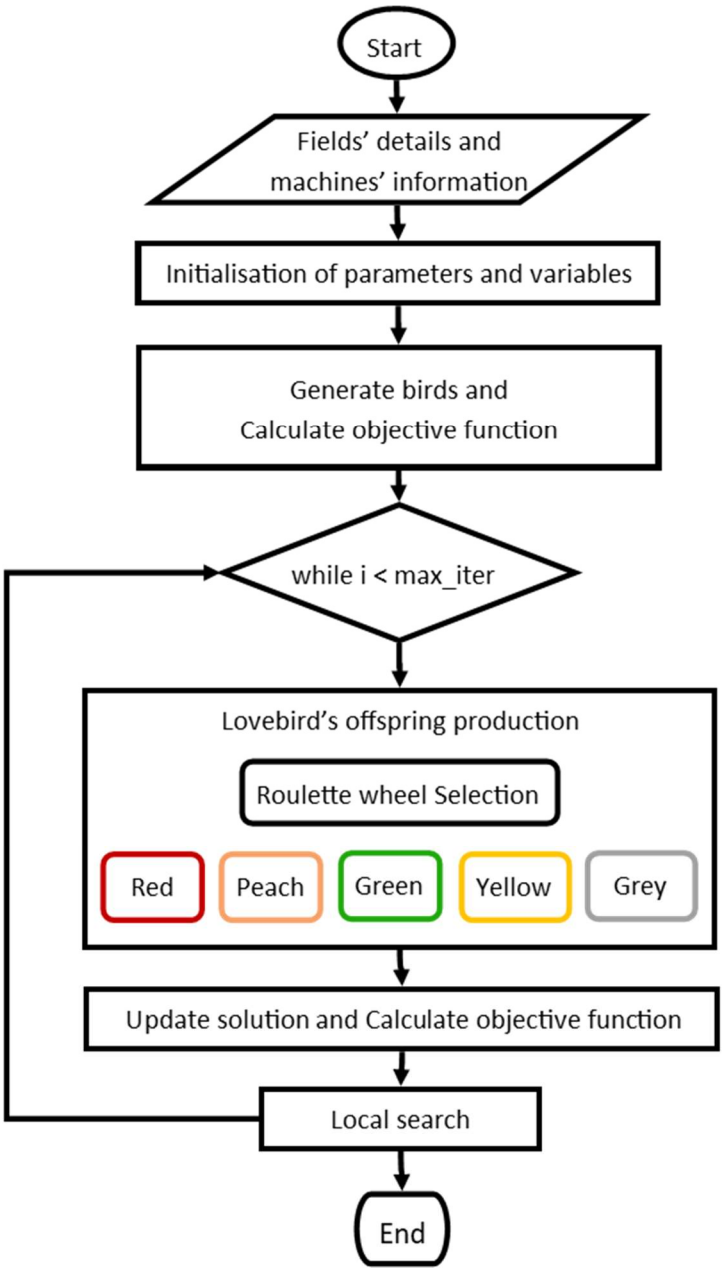


Figure 31. General flowchart of the stages of the Lovebird Algorithm

The key iterations begin with the production of offspring. Lovebird's offspring then carry out one of the following five options of combinatorics operators (illustrated in Figure 32):

- Red: Swap the sections
- Peach: Flip the sequence
- Green: Interchange two tracks
- Yellow: Move and push
- Grey: Mix the tracks.

Parent 1	1	2	3	4	5	6	7	8
Parent 2	2	1	4	5	6	7	8	3
Offspring 1	1	2	4	5	6	7	3	8
Offspring 2	2	1	3	4	5	6	7	8

(a) Red (Swap section)

Parent	2	1	4	5	6	7	8	3
Offspring	2	1	6	5	4	7	8	3

(b) Peach (Flip)

Parent	2	1	4	5	6	7	8	3
Offspring	2	8	6	5	4	7	1	3

(c) Green (Interchange)

Parent	2	1	4	5	6	7	8	3
Offspring	2	4	5	6	7	8	3	1

(d) Yellow (Move and push)

Parent	1	2	3	4	5	6	7	8
Offspring	2	8	6	5	4	7	1	3

(e) Grey (Mix)

Figure 32. Depiction of combinatorics operators in the Lovebird Algorithm

The offspring is the latest candidate solution after the mutation operator has been implemented, while the parent is the previous candidate solution. The swap section swaps the red section of the two parents. Offspring 1 holds the red segment of parent 2 and copies the rest of the tracks of parent 1, while offspring 2 does the

opposite. The flip operator changes the location of the track in the peach colour segment, while the interchange operator changes the location of the green portion. The move and push operator moves the position of the yellow front point to the yellow back point and pushes the remaining tracks forward. The mix operator combines the track sequences in a candidate solution.

Combinatorics operators are used as an experimental step in a metaheuristic algorithm (Soni & Kumar, 2014). The next step includes reviewing the latest candidate solutions and their objective values. The best solution among the iterations will be modified if a better solution is found in the current iteration. A Local Search explores the neighbourhood of the best solution found so far to consider whether further progress is possible. If *max_iter* is reached, then the algorithm will stop; otherwise, it will continue with the main iteration again.

6.5 Evolutionary Lovebird Algorithm (EvoLovebird)

The Evolutionary Lovebird Algorithm (EvoLovebird) is the enhanced version of the Lovebird Algorithm, as suggested in Utamima et al. (2019c). The improvements are a better choice of operator and an expansion of the Local Search technique. The goal of these changes is to improve the quality of the solutions and to achieve a faster running time.

Figure 33 represents the stages of EvoLovebird. The candidate solution in EvoLovebird is called a bird. The main iteration repeats for *max_iter* times and begins with the production of offspring (new birds). EvoLovebird adapts combinatoric mutation operators to produce the offspring (illustrated in Figure 34):

- Red: Swap tracks
- Peach: Flip the sequence
- Green: Interchange
- Yellow: Move and push
- Grey: Mix tracks.

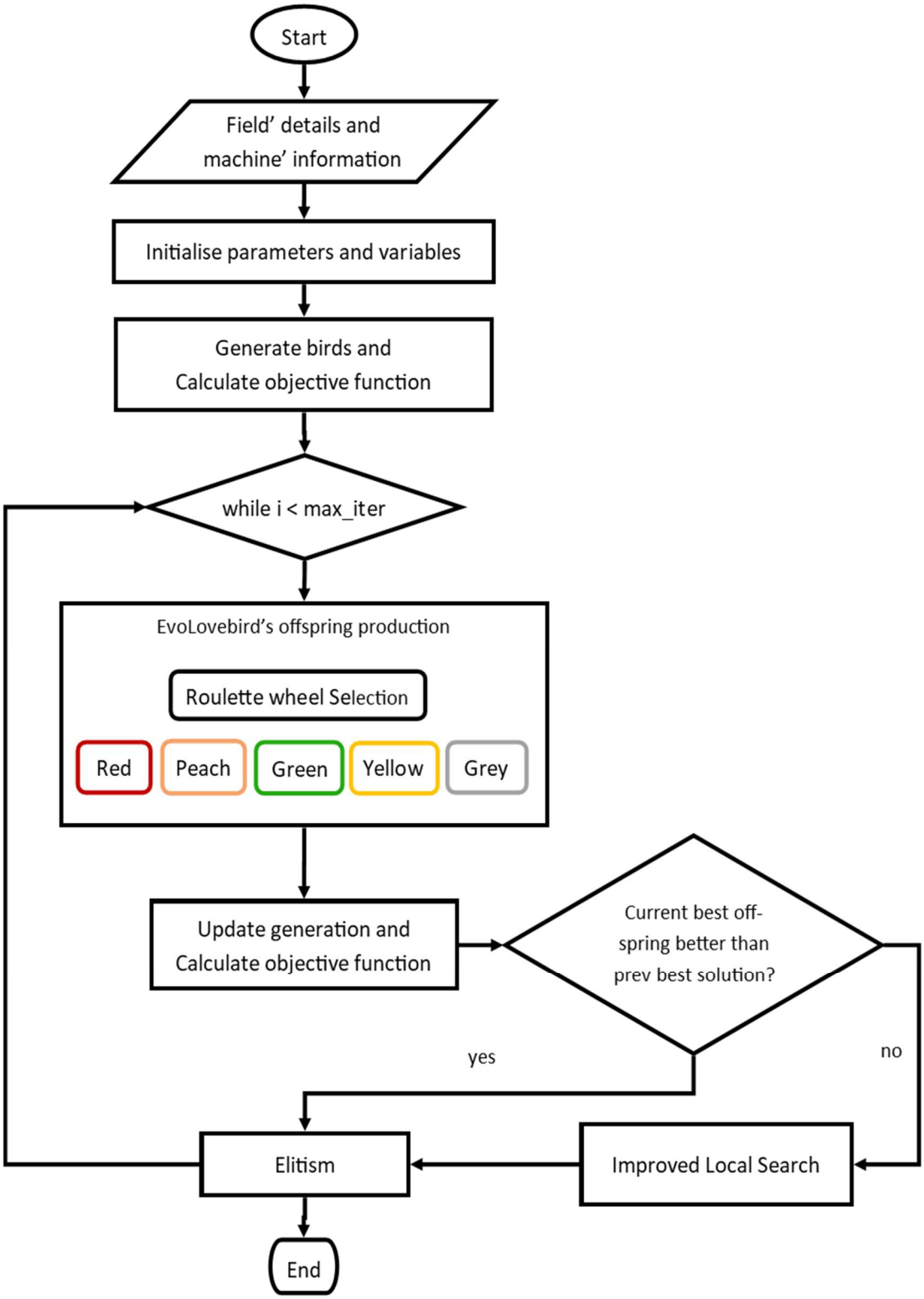


Figure 33. General flowchart of the stages of the Evolutionary Lovebird

Parent	1	2	3	6	1	7	4
Offspring	1	1	7	6	2	3	4

(a) Red (Swap tracks)

Parent	7	1	4	5	6	3	2
Offspring	7	1	6	5	4	3	2

(b) Peach (Flip)

Parent	6	1	4	5	2	3	7
Offspring	6	3	4	5	2	1	7

(c) Green (Interchange)

Parent	2	1	4	5	6	7	3
Offspring	2	4	5	6	7	3	1

(d) Yellow (Move and push)

Parent	1	2	3	4	5	6	7
Offspring	2	1	6	5	3	7	4

(e) Grey (Mix)

Figure 34. Depiction of combinatorics operators in EvoLovebird

The combinatorics operator is an evolutionary strategy implemented to chosen birds to build a better solution. The variation of operators is adapted from Soni and Kumar (2014) and Conesa-Muñoz et al. (2016b). The difference in the preference of operator between the Lovebird and EvoLovebird algorithms is the Red operator. The previous Red operator used a swap operator to exchange multiple tracks between two birds. Nonetheless, our study shows that the swap operator (from two birds) requires more running time than the swap elements within one bird. This enhanced Red operator, therefore, replaces the former Red operator.

Figure 34 displays the combinatorics operators utilised in EvoLovebird. The offspring is the new bird after the operator has been applied to the parent. The swap operator switches two red sections of the parent. The flip operator flips over the track position in the peach colour segment, while the interchange operator changes the place of the green part. The move and push operator moves the position of the yellow front to the yellow back and pushes the remaining tracks forward. The mix operator mixes track sequences in a bird.

The next step is to update the new birds and their objective values. If a better solution is found in the current iteration, the best solution among the iterations will be updated. Next, a Local Search checks the neighbourhood of the best solution found so far to discover the possible improvement of the solution. Local Search in EvoLovebird is also enhanced by the implementation of the mutation operators to the best bird found so far. Three types of mutation operators (swap, slide and flip) are applied to the best bird to find the best solution. The next step after a Local Search is to perform Elitism, adapted from (Utamima et al., 2019c). Elitism is at the end of the iteration and preserves the best birds with the best solutions over the generations.

6.6 Evolutionary Surrounding Search

This section overviews the general stages of the Evolutionary Surrounding Search (ESS), as illustrated in Figure 35. The detailed explanation of ESS has been listed in Section 5.3.

First, the ESS parameters are initialised (generation size, population size, distribution search rate and surrounding search iteration). The variables are also set at this stage. The candidate solutions (individuals) are generated and their objective function is calculated. The ESS exploration stage conducts a distribution search and surrounding search. The new group of individuals are then evaluated and used to update the previous generation. If there is no improvement in the best solution found, the surrounding search will be executed. At the end of every iteration, the Elitism process is performed.

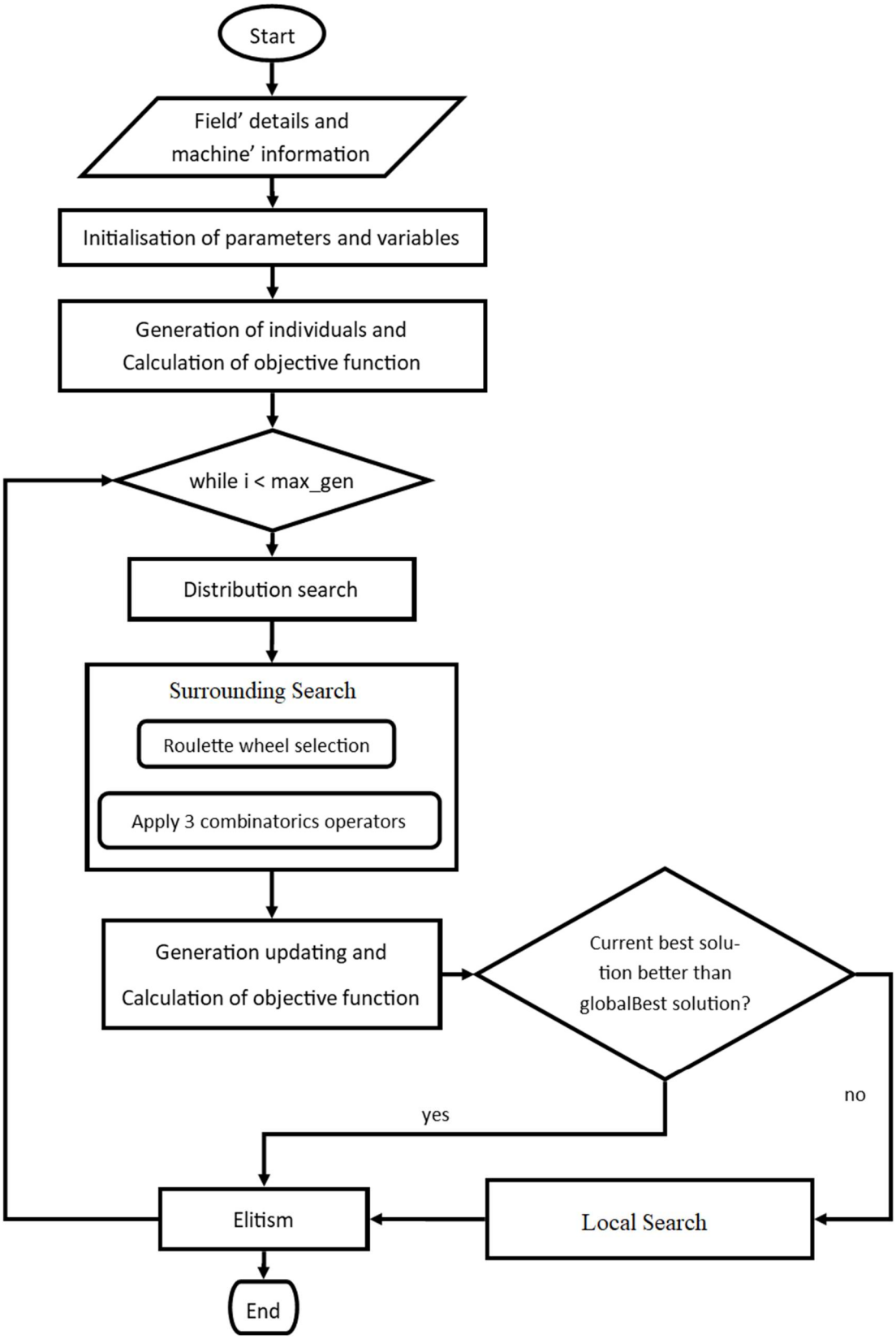


Figure 35. General flowchart of the stages of ESS

6.7 Characteristics of the Algorithms

The different characteristics of the algorithms are listed in Table 27. The EvoLovebird Algorithm is an improved version of the Lovebird Algorithm, while ESS is a combined version of EHNS and EEDA.

Table 27. The general characteristics of the algorithms

Algorithm	Characteristics
EHNS	♦ Produce a new solution with the neighbourhood search containing three kinds of mutation/combinatorics operators ♦ Local Search based on Tabu Search
EEDA	♦ Produce a new solution with the probabilistic model ♦ Local Search based on combinatorics operators
Lovebird	♦ Produce offspring with five kinds of combinatorics operators ♦ Local Search with combinatorics operators
EvoLovebird	♦ Improved version of the Lovebird Algorithm ♦ Produce offspring with five kinds of combinatorics operators ♦ Local Search with improved combinatorics operators
ESS	♦ Combined version of EHNS and EEDA ♦ Produce offspring with: - the probabilistic model - combinatorics operator ♦ Local Search with combinatorics operators

The author has demonstrated that EvoLovebird is able to outperform Lovebird (Utamima et al., 2020). Meanwhile, from the author's experiments, both the EHNS and ESS solutions for a dataset are better than EEDA. Hence, the following section will compare the performance of EHNS, EvoLovebird and ESS to solve the dataset in ARP.

6.8 A Comparison of EHNS, EvoLovebird and ESS in Solving the ARP Dataset

Table 28 lists the non-working distance comparison of all developed algorithms compared to GA and TS. The bold value represents the best (minimum) value of non-working distance in that row. For Problems 1–6, EHNS, ESS and EvoLovebird are able to get the same minimum value, which is better than those in GA (on all problems) and TS (Problems 4–6).

Table 28. The non-working distance comparison of all algorithms

Problem Code	Genetic Algorithm (GA)	Tabu Search (TS)	Evolutionary Hybrid Neighbourhood Search (EHNS)	Evolutionary Surrounding Search (ESS)	Evolutionary Lovebird Algorithm (EvoLovebird)
8rt	95.767	**94.439**	**94.439**	**94.439**	**94.439**
12rt_a	176.451	**146.027**	**146.027**	**146.027**	**146.027**
12rt_b	166.451	**145.602**	**145.602**	**145.602**	**145.602**
20rt	250.916	245.916	**235.491**	**235.491**	**235.491**
37nc	1142.474	1088.188	**958.930**	**958.930**	**958.930**
62Hs	562.414	559.914	**479.914**	**479.914**	**479.914**
90rt	2791.680	2870.172	2658.474	2661.871	**2651.871**
74nc	5212.590	*4416.30	3880.679	3869.590	**3832.206**
74nc2	6064.112	5856.542	5197.349	**5148.545**	5197.349

Note: The results of the specific problems in GA are obtained from Conesa-Muñoz et al. (2016a) and Hameed et al. (2011), and in TS from Seyyedhasani and Dvorak (2018a)

In Table 28, EvoLovebird achieves the best non-working distance compared to other algorithms in solving Problems 7–8, while ESS achieves the shortest non-working distance in Problem 9.

Table 29 lists the non-working distance comparison of GA, TS, EHNS, ESS and EvoLovebird in problems with multiple fields. The value in bold represents the

best (minimum) value of non-working distance in that row. In these problems, EHNS, ESS and EvoLovebird achieve better solutions than those in GA and TS.

EHNS, ESS and EvoLovebird achieve the same minimum distance on Problem 1. However, the EHNS solutions to the other problems are not better than those of ESS and EvoLovebird. Both ESS and EvoLovebird achieve the best solution for Problem 2. ESS provides the shortest distance in Problem 4, while EvoLovebird achieves the minimum distance in Problems 3, 5 and 6. In terms of the best solution, EvoLovebird achieves the shortest distance in five out of the six problems.

Table 29. The non-working distance comparison in multiple fields of all algorithms

No.	Problem Code	Genetic Algorithm (GA)	Tabu Search (TS)	Evolutionary Hybrid Neighbourhood Search (EHNS)	Evolutionary Surrounding Search (ESS)	Evolutionary Lovebird Algorithm (EvoLovebird)
1	P2	1051.12	1029.33	**953.44**	**953.44**	**953.44**
2	P3a	1293.81	1186.20	1119.93	**1109.37**	**1109.37**
3	P3b	2294.03	2469.78	2188.43	2164.27	**2091.97**
4	P4a	1968.95	1818.88	1767.65	**1700.11**	1703.41
5	P4b	2947.51	3063.43	2659.97	2615.47	**2517.59**
6	P5	4394.45	4677.44	4120.99	4074.422	**4034.04**

Table 30 provides the running time comparisons of GA, TS, EHNS, ESS and EvoLovebird in solving the problem with multiple fields. Generally, EvoLovebird is the fastest algorithm compared to others. The EvoLovebird running time is the fastest in five out of the six problems, while the ESS running time is the fastest in only one out of the six cases. EHNS gets the slowest running time in all instances. Both the GA and TS running time are generally similar; however, they are still behind those of ESS and EvoLovebird.

Table 30. The running time comparison in multiple fields of all algorithms

No.	Problem Code	Genetic Algorithm (GA)	Tabu Search (TS)	Evolutionary Hybrid Neighbourhood Search (EHNS)	Evolutionary Surrounding Search (ESS)	Evolutionary Lovebird Algorithm (EvoLovebird)
1	P2	1.388	1.359	2.828	0.852	**0.937**
2	P3a	4.375	3.234	8.691	2.315	**2.166**
3	P3b	12.994	17.988	48.823	10.544	**5.691**
4	P4a	9.633	7.032	28.182	3.622	**3.128**
5	P4b	25.608	23.814	71.693	12.969	**8.731**
6	P5	26.438	25.629	74.454	12.987	**8.960**

Figures 36 and 37 show the running time (x-axis) versus the non-working distance (y-axis; objective function) of EHNS, ESS and EvoLovebird when solving Problems 3 and 5, respectively. The maximum value of the running time of the algorithms is reduced, based on the time needed for the fastest algorithm (EvoLovebird). Both ESS and EvoLovebird start and end with a smaller non-working distance than EHNS. In other words, the convergence process of ESS and EvoLovebird is better than EHNS.

In Figure 36, EvoLovebird starts to converge in less than 2 seconds, while maintaining a shorter non-working distance than ESS and EHNS. In Figure 37, EvoLovebird starts better than ESS and EHNS. Then, in the 3.5th second until the 3.8th second, EvoLovebird's non-working distance is defeated by ESS. However, EvoLovebird's non-working distance outperforms ESS from the 3.9th second until the last second.

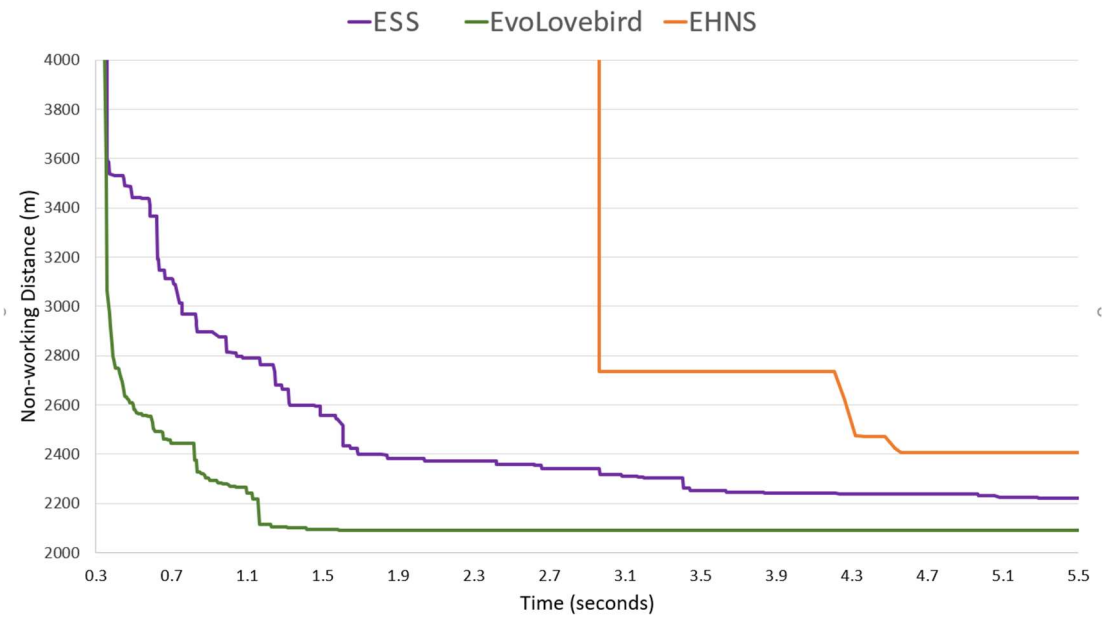


Figure 36. Running time versus distance of ESS, EvoLovebird, and EHNS for Problem 3

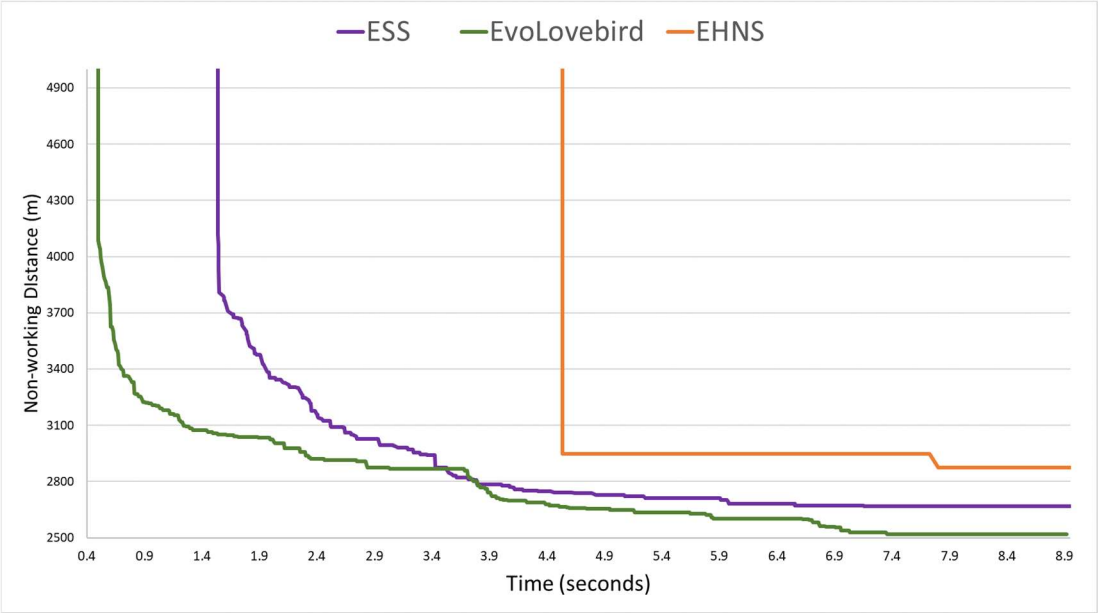


Figure 37. Running time versus distance of ESS, EvoLovebird, and EHNS for Problem 5

6.9 Discussion

Table 31 summaries the characteristics, strengths and weaknesses of EHNS, ESS and EvoLovebird. EHNS's strengths are the ease of coding the algorithm, and its solutions are better than the standard algorithms (GA and TS). However, it has the

slowest running time compared to other algorithms. The slow running time is the result of using TS as the Local Search.

Table 31. The strengths and weaknesses of EHNS, EvoLovebird and ESS

Algorithm	Characteristics	Strength	Weakness
EHNS	Neighbourhood search with three types of mutation or combinatorics operators Local Search with TS technique	Better solutions than GA and TS Easy to implement	Slow running time
ESS	Distribution search (Inter) surroundings search with three types of combinatorics operator Local Search with two types of combinatorics operators	Better solutions than GA, TS and EHNS Fast running time	Extra effort to implement
EvoLovebird	Neighbourhood search with five types of combinatorics operators Local Search with three types of combinatorics operators	Better solutions than GA, TS, EHNS 72% of solutions better than ESS Fast running time Strong Local Search	Extra effort to implement

The benefits of both ESS and EvoLovebird are the fast running time, and their solutions are better than GA, TS and EHNS. Their drawback is the extra effort required to implement and build the algorithm. EvoLovebird's combinatoric operators increase the ability to search for improvements in the group of candidate solutions, leading to a better solution compared to the other algorithms. The experiments also show that Local Search is essential in achieving an improvement in the global best solution. Therefore, a strong Local Search is necessary for building an evolutionary algorithm.

In this study, the reliability of the algorithm and the validity of the solutions were demonstrated and confirmed against previously conducted studies in academic publications by not only getting the same or better solutions but also finding the best solution in a significantly shorter time. As described in the limitations (Section 8.3), further experiments need to be conducted to evaluate the algorithms on other field constellations and extended parameter settings.

Chapter 7. Practical Applications

This chapter contains parts of two conference papers. One has been published in the Australasian Conference on Information Systems 2019 (Section 7.1), and the other accepted in the 12th International Conference on Computer and Automation Engineering 2020 and indexed in ACM Digital Library (Section 7.2). The details of the conference paper can be found in Appendix B and C. In this chapter, the possible future implementation of ARP is discussed as part of Decision Support Systems (DSS) and its application to different field operations.

7.1 Decision Support Systems

A recent trend in the agricultural sector is the integration of computers to support automation in the operation of small- and large-scale farms. The utilisation of computers for decision making is critical for farmers wanting to lower their operating costs and control their machines. Decision making supported by a useful DSS can improve the quality of the decision in the agricultural field.

The main objective of DSS is to help, improve and potentially automate the decision-making process (Turban et al., 2005). A decision-making process based on an optimisation technique is related to the recognition and solution of optimisation problems. Computer programs that solve optimisation problems are an essential element of several DSS (Bernus & Holsapple, 2008).

Recent studies have proposed several DSS for agriculture. For example, A DSS in fish farming is intended to optimise production strategies. The DSS contains an optimisation module that uses PSO to optimise seabream aquaculture production (Cobo et al., 2019). A fuzzy-based DSS is proposed to improve irrigation in agriculture by deciding whether irrigation is needed and determining the amount required according to a set of rules involving variations of several weather variables (Giusti & Marsili-Libelli, 2015). Navarro-Hellín et al. (2016) improved the DSS in irrigation by considering the soil measurement to precisely predict the irrigation needs. An agro-climate decision support tool is proposed to help users run crop simulation models for the targeted crops (Han et al., 2019).

Despite the variations of DSS in agriculture, no formal studies apply ARP in the form of decision making by farmers. Consequently, the author's work on this topic is the first to consider ARP for such decision-making processes (Utamima et al., 2019c). The study also contributes to the development of a new algorithm (the Lovebird Algorithm), and its application is represented in an optimisation module of DSS in ARP.

Figure 38 represents the proposed framework of an optimisation module in DSS for ARP. This framework is in line with Cobo et al. (2019) and Ben Jouida and Krichen (2018). The inputs of the module consist of the coordinates of every track, every entrance to each field and the depot, as well as the number and capacity of the machines. The process starts with the calculation of track distances inside the fields and to the entrances and the depot. Optimisation with the evolutionary algorithm is then executed. The outputs of the module are the optimised order of tracks that need to be traversed by the machines, the non-working distance and the length of harvested tracks (working distance).

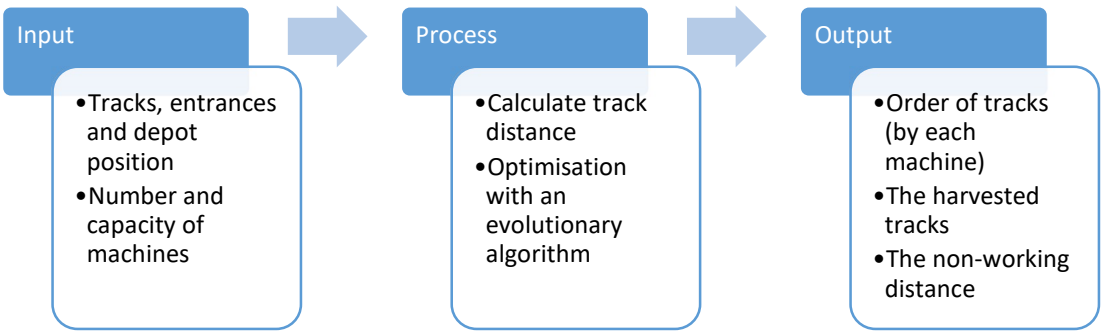


Figure 38. The framework of the optimisation module in DSS for ARP

The proposed module can be integrated into a fleet management system in agriculture (Sørensen & Bochtis, 2010). This system includes an online decision support system, which will help the farmer run the machines through online routing. The primary stakeholders on the DSS are the farm manager to manage and optimise the operations of their fleet. The DSS can be used to calculate the best solutions for different scenarios and thus support the efficient and effective fleet operation by providing the minimum distance and time to operate the fields. In addition, the algorithms provide route and path guidance for the driver of the machines. The DSS

can further improve the operations by receiving current positions of the machines to continuously recalculate the best solution.

Another alternative is to add the evolutionary algorithm to a comprehensive farm management system (Sørensen et al., 2010). This new design of a farm management system considers new situations from the perspective of both the farmers and the farm manager. Specifically, the routing algorithm can be included in one of the system modules, called 'plan generation'. After that, the farmers can execute the route provided by the management system.

The author introduced the Lovebird Algorithm as the routing algorithm for ARP to achieve the optimised solution (Utamima et al., 2019c). The Lovebird Algorithm has been described previously in Section 6.4. The Lovebird Algorithm acts as an optimisation tool to screen alternatives and focus only on efficient ones. The experimental results show that the proposed algorithm can outperform GA and TS.

7.2 Fertilising Operation

ARP can be applied in different field operations in agriculture. The field operation involves arable farming and orchard farming, which includes operations in tillage, mowing, conditioning, weeding, seeding, spraying and fertilising, etc. (Bochtis & Sørensen, 2009). Examples of tillage include seedbed preparation and cultivation, while spraying includes pest control and weed control.

Despite the variations of ARP in agriculture, the issue of seeding, spraying and fertiliser application has not been dealt with comprehensively, with studies tending to focus on only single-field operations. The author is the first to address ARP for fertiliser application in more than one field (Utamima et al., 2020). The ARP observed in this paper uses multiple machines with the same capacity. The complete paper can be found in Appendix E. The adapted mathematical model is similar to the one provided in Section 4.3.

In reference to maximum distance constraints, optimisation will assign machines to the routes provided based on the amount of fertiliser they hold. For example, if a machine is running out of fertiliser, it must return to the depot for a

refill before the next tour begins. The assumption is a uniform volume of fertiliser per travelled distance. Thus, in a machine, the volume of the fertiliser can be determined by the maximum distance it can travel on one tour. As the problem considers the machine's capacity, the mathematical model and the proposed method can also be applied to seeding and spraying in the field.

The author used EvoLovebird to determine the optimum path to take when fertilising fields (Utamima et al., 2020). EvoLovebird has been described previously in Section 6.5. The fertilising case study uses five generated instances with multiple fields and several homogeneous machines. Utamima et al. (2020) also provides a comparison of EvoLovebird with GA and TS. The experimental results showed the superiority of the proposed algorithm concerning the savings in the non-working distance and time. The running time of EvoLovebird was three times faster than GA and TS in all problem instances.

7.3 Discussion

The output of this thesis is expected to be able to reduce the resources required in ARP (i.e. machines, fuel and personnel), which may result in a competitive advantage and access to markets in a lower price segment. This will contribute to addressing the sustainability issue of agricultural fields, by considering its impact on the improvement of economic and environmental factors. In economic terms, ARP can reduce the general cost of operating the machines, including the fuel cost, by minimising the travelled distance. By lowering the travelled distance, this can help to reduce fuel consumption (energy saving) and consequently decrease the environmental impact of CO_2 emissions.

Chapter 8. Conclusion

8.1 Contribution

The aim of ARP optimisation is to minimise the total distance driven to manage a field. By solving the proposed ARP with the suggested algorithm, this research is significant as it is expected to improve the fleet management and operations of the machinery in agriculture by generating effective routing plans. The better the routing plan, the shorter the distance travelled by the machines and the less time needed for doing field operations. This will benefits farmer and farm manager as well. Quantitative improvements such as reduced resources (e.g. fuel, machines and workers) will result in a competitive advantage and market access to the lower price segment. This would then address the sustainability problem in agriculture by taking into account the impact on the development of economic (cost reduction) and environmental factors and energy savings (reduced fuel consumption) and, as a result, lowering greenhouse gas emissions.

As well as the significant benefits indicated earlier, this study also presents several contributions:

1. Defining an extended ARP that deals with multiple fields, tracks and machines with different capacities and turning manoeuvres.
2. Developing the mathematical formulations to represent the general problem sets (from previous research) and the extended ARP.
3. Building and exploring new evolutionary algorithms to address the ARP.
4. Benchmarking the new evolutionary algorithms against existing algorithms in the literature.

8.2 Summary and Concluding Remarks

A variety of examples of ARP optimisation have been implemented in the agriculture literature, and solutions have been presented using different algorithms. To collect benchmark data in ARP, this research gathered various datasets reflecting

the past studies. The thesis then introduced the extended ARP by considering several constraints at the same time to generalise the problem of harvesting more effectively. The work also generates seven problem sets for the extended ARP.

This study introduces several new evolutionary and hybrid algorithms, including the Evolutionary Hybrid Neighbourhood Search (EHNS), Evolutionary Surrounding Search (ESS) and Evolutionary Lovebird Algorithm (EvoLovebird). The experimental results demonstrate that during the validation process with the previous ARP dataset, the application of these algorithms can achieve either the same or better optimal solutions in all instances.

The experimental results also show that the proposed evolutionary algorithms (EvoLovebird, EHNS and ESS) made improvement in different problem sets in ARP compared to the previous algorithms utilised in the literature (GA and TS). From a total of nine problem sets obtained from the literature, EvoLovebird obtains the best results in eight of the problems and EHNS in six of the problems, while ESS achieves the minimum solution in seven of the problems. On the other hand, TS provides the best solution in three of the problems, while GA provides the worst solutions in all problems compared to the other algorithms.

The proposed evolutionary algorithms also perform well in extended ARP problem sets. From a total of six instances, EvoLovebird achieves the best solution in five of the problems, EHNS gets the minimum solution in one problem, while ESS obtains the best solution in three of the problems. On the other hand, both GA and TS fail to get a single best solution. EHNS, ESS and EvoLovebird maintain the best objective function and the fastest convergence speed in all general and extended ARP problems when compared to GA and TS. Furthermore, the running time of both ESS and EvoLovebird are faster than GA and TS.

8.3 Limitations and Future Research

This scope of this study is limited to the use of data provided from the literature or generated (hypothetical) data. The data comprises two-dimensional fields with straight and parallel tracks. The headland areas are assumed to be wide enough for the machines to make the required manoeuvres.

Future research may focus on building an entire DSS to help decision making in ARP. Various applications of the DSS in ARP, including optimised machine routing, may also be considered as a future direction, such as for orchard operations and herbicide applications.

Future studies may also focus on extending the experiments utilising the evolutionary algorithms. Additional ARP problem sets could be generated to test the algorithms and their reliability of providing sophisticated results. The algorithms could then be further tuned by applying different parameter settings. The different combinatorics operators could be embedded in the algorithm to maintain the diversity of the algorithm, which may lead to improvements in the quality of the solution.

More prospective research could concentrate on combining multiple systems, enhancing field coordination, or including DSS and other information systems relevant to field operations and harvested crop management. The information on minimised machine routes is essential for both current and future agricultural field management. In the future, optimised routes may become input factors for field robots or autonomous vehicles used for field harvesting.

References

Adrian, A. M., Utamima, A., & Wang, K. J. (2015). A comparative study of GA, PSO and ACO for solving construction site layout optimization. *KSCE Journal of Civil Engineering*, *19*(3). https://doi.org/10.1007/s12205-013-1467-6

Akhtari, S., Sowlati, T., & Day, K. (2014). Optimal flow of regional forest biomass to a district heating system. *International Journal of Energy Research*, *38*(7), 954–964. https://doi.org/10.1002/er.3099

Al-Zamil, A., & Saudagar, A. K. J. (2018). Drivers and challenges of applying green computing for sustainable agriculture: A case study. *Sustainable Computing: Informatics and Systems*. https://doi.org/10.1016/J.SUSCOM.2018.07.008

Amiama, C., Cascudo, N., Carpente, L., & Cerdeira-Pena, A. (2015). A decision tool for maize silage harvest operations. *Biosystems Engineering*, *134*, 94–104. https://doi.org/10.1016/j.biosystemseng.2015.04.004

Archetti, C., & Speranza, M. G. (2014). A survey on matheuristics for routing problems. *EURO Journal on Computational Optimization*, *2*(4), 223–246. https://doi.org/10.1007/s13675-014-0030-7

Bakhtiari, A., Navid, H., Mehri, J., Berruto, R., & Bochtis, D. D. (2013). Operations planning for agricultural harvesters using ant colony optimization. *Spanish Journal of Agricultural Research*, *11*(3), 652–660. https://doi.org/10.5424/sjar/2013113-3865

Barbosa, H. (2013). *Ant Colony Optimization: Techniques and Applications*. IntechOpen. https://books.google.com.au/books?id=tPyPDwAAQBAJ

Barrientos, A., Colorado, J., Cerro, J. Del, Martinez, A., Rossi, C., Sanz, D., & Valente, J. (2011). Aerial remote sensing in agriculture: A practical approach to area coverage and path planning for fleets of mini aerial robots. *Journal of Field Robotics*, *28*(5), 667–689. https://doi.org/10.1002/rob.20403

Ben Jouida, S., & Krichen, S. (2018). A DSS based on optimizer tools and MTS meta-heuristic for the Warehousing Problem with Conflicts. *Information Processing Letters*, *135*, 14–21. https://doi.org/10.1016/j.ipl.2018.02.001

Bernus, F., & Holsapple, C. W. (2008). Handbook on Information Systems 1: Basis Themes.

In *Decision Support Systems*. Springer-Verlag. https://doi.org/10.1007/978-3-540-48713-5

Bochtis, D. D., & Sørensen, C. G. (2009). The vehicle routing problem in field logistics part I. *Biosystems Engineering*, *104*(4), 447–457. https://doi.org/10.1016/j.biosystemseng.2009.09.003

Bochtis, D. D., & Sørensen, C. G. (2010). The vehicle routing problem in field logistics: Part II. *Biosystems Engineering*, *105*(2), 180–188. https://doi.org/10.1016/j.biosystemseng.2009.10.006

Bochtis, D. D., Sørensen, C. G., Busato, P., & Berruto, R. (2013). Benefits from optimal route planning based on B-patterns. *Biosystems Engineering*, *115*(4), 389–395. https://doi.org/10.1016/j.biosystemseng.2013.04.006

Bochtis, D. D., Sørensen, C. G. C., & Busato, P. (2014). Advances in agricultural machinery management: A review. *Biosystems Engineering*, *126*, 69–81. https://doi.org/10.1016/j.biosystemseng.2014.07.012

Bochtis, D. D., & Vougioukas, S. G. (2008). Minimising the non-working distance travelled by machines operating in a headland field pattern. *Biosystems Engineering*, *101*(1), 1–12. https://doi.org/10.1016/j.biosystemseng.2008.06.008

Câmara, D., & Câmara, D. (2015). Evolution and Evolutionary Algorithms. In *Bio-inspired Networking* (pp. 1–30). Elsevier. https://doi.org/10.1016/B978-1-78548-021-8.50001-6

Chen, S.-H., Chen, M.-C., & Liou, Y.-C. (2014). Artificial chromosomes with genetic algorithm 2 (ACGA2) for single machine scheduling problems with sequence-dependent setup times. *Applied Soft Computing*, *17*, 167–175. https://doi.org/10.1016/J.ASOC.2013.12.019

Clerc, M. (2010). Particle Swarm Optimization. In *Particle Swarm Optimization*. Wiley-ISTE. https://doi.org/10.1002/9780470612163

Cobo, Á., Llorente, I., Luna, L., & Luna, M. (2019). A decision support system for fish farming using particle swarm optimization. *Computers and Electronics in Agriculture*, *161*(February 2018), 121–130. https://doi.org/10.1016/j.compag.2018.03.036

Conesa-Muñoz, J., Bengochea-Guevara, J. M., Andujar, D., & Ribeiro, A. (2016a). Route planning for agricultural tasks: A general approach for fleets of autonomous vehicles in

site-specific herbicide applications. *Computers and Electronics in Agriculture*, *127*, 204–220. https://doi.org/10.1016/j.compag.2016.06.012

Conesa-Muñoz, J., Pajares, G., & Ribeiro, A. (2016b). Mix-opt: A new route operator for optimal coverage path planning for a fleet in an agricultural environment. *Expert Systems with Applications*, *54*, 364–378. https://doi.org/10.1016/j.eswa.2015.12.047

Cordeau, J.-F., Gendreau, M., Laporte, G., Potvin, J.-Y., & Semet, F. (2002). A guide to vehicle routing heuristics. *Journal of the Operational Research Society*, *53*(5), 512–522. https://doi.org/10.1057/palgrave/jors/2601319

Dorigo, M., & Stützle, T. (2004). *Ant Colony Optimization*. MIT Press.

Edwards, G. T. C., Hinge, J., Skou-Nielsen, N., Villa-Henriksen, A., Sørensen, C. A. G., & Green, O. (2017). Route planning evaluation of a prototype optimised infield route planner for neutral material flow agricultural operations. *Biosystems Engineering*, *153*, 149–157. https://doi.org/10.1016/j.biosystemseng.2016.10.007

Fisher, M. (1995). Vehicle routing. In *Handbooks in Operations Research and Management Science* (Vol. 8, pp. 1–33). Elsevier. https://doi.org/https://doi.org/10.1016/S0927-0507(05)80105-7

Florentino, H. de O., Irawan, C., Aliano, A. F., Jones, D. F., Cantane, D. R., & Nervis, J. J. (2018). A multiple objective methodology for sugarcane harvest management with varying maturation periods. *Annals of Operations Research*, *267*(1–2), 153–177. https://doi.org/10.1007/s10479-017-2568-2

Galceran, E., & Carreras, M. (2013). A survey on coverage path planning for robotics. *Robotics and Autonomous Systems*, *61*(12), 1258–1276. https://doi.org/10.1016/j.robot.2013.09.004

Galvan, B., D., G., J., P., Sefrioui, M., & Winter, G. (2003). Parallel Evolutionary Computation for Solving Complex CFD Optimization Problems : A Review and Some Nozzle Applications. In *Parallel Computational Fluid Dynamics 2002* (pp. 573–604). North-Holland. https://doi.org/10.1016/B978-044450680-1/50072-3

Gen, M., & Cheng, R. (2000). *Genetic algorithms and engineering optimization*. Wiley. https://books.google.com.au/books?hl=en&lr=&id=U7MuV1q6P1oC&oi=fnd&pg=PR13&dq=algorithm+optimization+validation&ots=50yjFoxnlp&sig=zVhU0-WA5Kfh5cID-0dTEUwYV74#v=onepage&q=algorithm optimization

validation&f=false

Giusti, E., & Marsili-Libelli, S. (2015). A Fuzzy Decision Support System for irrigation and water conservation in agriculture. *Environmental Modelling & Software*, *63*, 73–86. https://doi.org/10.1016/J.ENVSOFT.2014.09.020

Glover, F., & Laguna, M. (1997). Tabu Search. In *INFORM Journal on Computing* (1st ed., Vol. 1). Springer US. https://doi.org/10.1007/978-1-4615-6089-0

Gonzalez-de-Soto, M., Emmi, L., Garcia, I., & Gonzalez-de-Santos, P. (2015). Reducing fuel consumption in weed and pest control using robotic tractors. *Computers and Electronics in Agriculture*, *114*, 96–113. https://doi.org/10.1016/j.compag.2015.04.003

Gracia, C., Velázquez-Martí, B., & Estornell, J. (2014). An application of the vehicle routing problem to biomass transportation. *Biosystems Engineering*, *124*, 40–52. https://doi.org/10.1016/J.BIOSYSTEMSENG.2014.06.009

Graf Plessen, M. (2019). Optimal in-field routing for full and partial field coverage with arbitrary non-convex fields and multiple obstacle areas. *Biosystems Engineering*, *186*, 234–245. https://doi.org/10.1016/j.biosystemseng.2019.08.001

Graf Plessen, M. M., & Bemporad, A. (2017). Reference trajectory planning under constraints and path tracking using linear time-varying model predictive control for agricultural machines. *Biosystems Engineering*, *153*(2013), 28–41. https://doi.org/10.1016/j.biosystemseng.2016.10.019

Guan, J., & Lin, G. (2016). Hybridizing variable neighborhood search with ant colony optimization for solving the single row facility layout problem. *European Journal of Operational Research*, *248*(3), 899–909. https://doi.org/10.1016/j.ejor.2015.08.014

Hameed, I. A., Bochtis, D. D., & Sorensen, C. G. (2011). Driving angle and track sequence optimization for operational path planning using genetic algorithms. *Applied Engineering in Agriculture*, *27*(6), 1077–1086. https://doi.org/10.13031/2013.40615

Hameed, I. A., Bochtis, D., & Sørensen, C. A. (2013). An optimized field coverage planning approach for navigation of agricultural robots in fields involving obstacle areas. *International Journal of Advanced Robotic Systems*, *10*. https://doi.org/10.5772/56248

Han, E., Baethgen, W. E., Ines, A. V. M., Mer, F., Souza, J. S., Berterretche, M., Atunez, G., & Barreira, C. (2019). SIMAGRI: An agro-climate decision support tool. *Computers*

and Electronics in Agriculture, *161*, 241–251. https://doi.org/10.1016/J.COMPAG.2018.06.034

Hansen, P., Mladenović, N., & Moreno Pérez, J. A. (2010). Variable neighbourhood search: Methods and applications. *Annals of Operations Research*, *175*(1), 367–407. https://doi.org/10.1007/s10479-009-0657-6

Hansen, P., Mladenović, N., Todosijević, R., & Hanafi, S. (2017). Variable neighborhood search: basics and variants. *EURO Journal on Computational Optimization*, *5*(3), 423–454. https://doi.org/10.1007/s13675-016-0075-x

Haupt, R. L., & Haupt, S. E. (2004). *PRACTICAL GENETIC ALGORITHMS* (2nd ed.). Wiley-Interscience.

Hauschild, M., & Pelikan, M. (2011). An introduction and survey of estimation of distribution algorithms. *Swarm and Evolutionary Computation*, *1*(3), 111–128. https://doi.org/10.1016/j.swevo.2011.08.003

He, P., & Li, J. (2019). The two-echelon multi-trip vehicle routing problem with dynamic satellites for crop harvesting and transportation. *Applied Soft Computing Journal*, *77*, 387–398. https://doi.org/10.1016/j.asoc.2019.01.040

He, P., Li, J., Zhang, D., & Wan, S. (2018). Optimisation of the harvesting time of rice in moist and non-moist dispersed fields. *Biosystems Engineering*, *170*, 12–23. https://doi.org/10.1016/J.BIOSYSTEMSENG.2018.03.008

Jensen, M. A. F., Bochtis, D., Sørensen, C. G., Blas, M. R., & Lykkegaard, K. L. (2012). In-field and inter-field path planning for agricultural transport units. *Computers and Industrial Engineering*, *63*(4), 1054–1061. https://doi.org/10.1016/j.cie.2012.07.004

Jensen, M. F., Bochtis, D., & Sørensen, C. G. (2015). Coverage planning for capacitated field operations, part II: Optimisation. *Biosystems Engineering*, *139*(2009), 149–164. https://doi.org/10.1016/j.biosystemseng.2015.07.002

Ji, M., & Tang, H. (2004). Global optimizations and tabu search based on memory. *Applied Mathematics and Computation*, *159*(2), 449–457. https://doi.org/10.1016/j.amc.2003.10.028

Jin, J., & Tang, L. (2010). Optimal Coverage Path Planning for Arable Farming on 2D Surfaces. *Transactions of the ASABE*, *53*(1), 283–295.

https://doi.org/10.13031/2013.29488

Jones, J. W., Antle, J. M., Basso, B., Boote, K. J., Conant, R. T., Foster, I., Godfray, H. C. J., Herrero, M., Howitt, R. E., Janssen, S., Keating, B. A., Munoz-Carpena, R., Porter, C. H., Rosenzweig, C., & Wheeler, T. R. (2017). Brief history of agricultural systems modeling. *Agricultural Systems*, *155*, 240–254. https://doi.org/10.1016/j.agsy.2016.05.014

Jung, S., & Moon, B. R. (2002). Toward minimal restriction of genetic encoding and crossovers for the two-dimensional Euclidean TSP. *IEEE Transactions on Evolutionary Computation*, *6*(6), 557–565. https://doi.org/10.1109/TEVC.2002.804321

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by Simulated Annealing. *Science*, *220*(4598), 671–680. https://doi.org/10.1126/science.220.4598.671

Kittilertpaisan, K., & Pathumnakul, S. (2017). Integrating a multiple crop year routing design for sugarcane harvesters to plant a new crop. *Computers and Electronics in Agriculture*, *136*, 58–70. https://doi.org/10.1016/j.compag.2017.03.001

Knuth, D. (2007). MOP Evolutionary Algorithm Approaches. In *Evolutionary Algorithms for Solving Multi-Objective Problems: Second Edition* (pp. 61–130). Springer US. https://doi.org/10.1007/978-0-387-36797-2_2

Koç, Ç., Bektaş, T., Jabali, O., & Laporte, G. (2016). Thirty years of heterogeneous vehicle routing. *European Journal of Operational Research*, *249*(1), 1–21. https://doi.org/10.1016/j.ejor.2015.07.020

Laarhoven, P. J. van, & Aarts, E. H. (2013). *Simulated Annealing: Theory and Applications*. Springer Netherlands. https://books.google.com.au/books?id=GPPqCAAAQBAJ

Lipowski, A., & Lipowska, D. (2012). Roulette-wheel selection via stochastic acceptance. *Physica A: Statistical Mechanics and Its Applications*, *391*(6), 2193–2196. https://doi.org/10.1016/J.PHYSA.2011.12.004

Marinho Diana, R. O., & de Souza, S. R. (2020). Analysis of variable neighborhood descent as a local search operator for total weighted tardiness problem on unrelated parallel machines. *Computers and Operations Research*, *117*, 104886. https://doi.org/10.1016/j.cor.2020.104886

Mladenović, N., & Hansen, P. (1997). Variable neighborhood search. *Handbook of*

Heuristics, *1–2*(1), 759–787. https://doi.org/10.1007/978-3-319-07124-4_19

Montgomery, D. C. (2013). *Design and Analysis of Experiments* (Vol. 8). John Wiley & Sons, New York.

Navarro-Hellín, H., Martínez-del-Rincon, J., Domingo-Miguel, R., Soto-Valles, F., & Torres-Sánchez, R. (2016). A decision support system for managing irrigation in agriculture. *Computers and Electronics in Agriculture*, *124*, 121–131. https://doi.org/10.1016/J.COMPAG.2016.04.003

Nayeem, M. A., Rahman, M. K., & Rahman, M. S. (2014). Transit network design by genetic algorithm with elitism. *Transportation Research Part C: Emerging Technologies*, *46*, 30–45. https://doi.org/10.1016/J.TRC.2014.05.002

Nazarahari, M., Khanmirza, E., & Doostie, S. (2019). Multi-objective multi-robot path planning in continuous environment using an enhanced genetic algorithm. *Expert Systems with Applications*, *115*, 106–120. https://doi.org/10.1016/j.eswa.2018.08.008

Nouri, N., & Ladhari, T. (2018). Evolutionary multiobjective optimization for the multi-machine flow shop scheduling problem under blocking. *Annals of Operations Research*, *267*(1–2), 413–430. https://doi.org/10.1007/s10479-017-2465-8

Oksanen, T., & Visala, A. (2009). Coverage path planning algorithms for agricultural field machines. *Journal of Field Robotics*, *26*(8), 651–668. https://doi.org/10.1002/rob.20300

Ou-Yang, C., & Utamima, A. (2013). Hybrid Estimation of Distribution Algorithm for solving Single Row Facility Layout Problem. *Computers and Industrial Engineering*, *66*(1), 95–103. https://doi.org/10.1016/j.cie.2013.05.018

Panigrahi, B. K., Shi, Y., & Lim, M. H. (2011). *Handbook of Swarm Intelligence: Concepts, Principles and Applications*. Springer Berlin Heidelberg. https://books.google.com.au/books?id=LYsbMvMRvckC

Pereira, F. B., & Tavares, J. (2009). *Bio-inspired Algorithms for the Vehicle Routing Problem* (J. Tavares & S. (Online Service) (eds.)). Berlin, Heidelberg : Springer Berlin Heidelberg.

Sethanan, K., & Neungmatcha, W. (2016). Multi-objective particle swarm optimization for mechanical harvester route planning of sugarcane field operations. *European Journal*

of Operational Research, *252*(3), 969–984. https://doi.org/10.1016/J.EJOR.2016.01.043

Seyyedhasani, H., & Dvorak, J. S. (2017). Using the Vehicle Routing Problem to reduce field completion times with multiple machines. *Computers and Electronics in Agriculture*, *134*, 142–150. https://doi.org/10.1016/j.compag.2016.11.010

Seyyedhasani, H., & Dvorak, J. S. (2018a). Reducing field work time using fleet routing optimization. *Biosystems Engineering*, *169*, 1–10. https://doi.org/10.1016/j.biosystemseng.2018.01.006

Seyyedhasani, H., & Dvorak, J. S. (2018b). Dynamic rerouting of a fleet of vehicles in agricultural operations through a Dynamic Multiple Depot Vehicle Routing Problem representation. *Biosystems Engineering*, *171*, 63–77. https://doi.org/10.1016/J.BIOSYSTEMSENG.2018.04.003

Seyyedhasani, H., Dvorak, J. S., & Roemmele, E. (2019). Routing algorithm selection for field coverage planning based on field shape and fleet size. *Computers and Electronics in Agriculture*, *156*(March 2018), 523–529. https://doi.org/10.1016/j.compag.2018.12.002

Sivanandam, S. N., & Deepa, S. N. (2008). *Introduction to genetic algorithms*. Springer.

Song, X., Jones, D., Asgari, N., & Pigden, T. (2019). Multi-objective vehicle routing and loading with time window constraints: a real-life application. *Annals of Operations Research*. https://doi.org/10.1007/s10479-019-03205-2

Soni, N., & Kumar, T. (2014). Study of Various Mutation Operators in Genetic Algorithms. *International Journal of Computer Science and Information Technologies*, *5*(3), 7235–7238.

Sørensen, C. G., & Bochtis, D. D. (2010). Conceptual model of fleet management in agriculture. *Biosystems Engineering*, *105*(1), 41–50. https://doi.org/10.1016/J.BIOSYSTEMSENG.2009.09.009

Sørensen, C. G., Fountas, S., Nash, E., Pesonen, L., Bochtis, D., Pedersen, S. M., Basso, B., & Blackmore, S. B. (2010). Conceptual model of a future farm management information system. *Computers and Electronics in Agriculture*, *72*(1), 37–47. https://doi.org/10.1016/j.compag.2010.02.003

Spekken, M., & de Bruin, S. (2013). Optimized routing on agricultural fields by minimizing maneuvering and servicing time. *Precision Agriculture*, *14*(2), 224–244. https://doi.org/10.1007/s11119-012-9290-5

Toth, P., & Vigo, D. (2002). The Vehicle Routing Problem. In *Monographs on Discrete Mathematics and Applications*. Society for Industrial and Applied Mathematics. https://books.google.com.au/books?id=TeMgA5S74skC

Toth, P., & Vigo, D. (2014). *Vehicle Routing, Problems, Methods and Applications* (2nd ed.). Society for Industrial and Applied Mathematics.

Turban, E., Aronson, J. E., & Liang, T.-P. (2005). *Decision support systems and intelligent systems*. Pearson/Prentice Hall. https://dl.acm.org/citation.cfm?id=994103

Utamima, A., Pradina, K. R., Dini, N. S., & Studiawan, H. (2015). Distribution Route Optimization of Gallon Water Using Genetic Algorithm and Tabu Search. *Procedia Computer Science*, *72*, 503–510. https://doi.org/10.1016/J.PROCS.2015.12.132

Utamima, A., Reiners, T., & Ansaripoor, A. H. (2019a). Decision making for Farmers : A case study of Agricultural Routing Planning. *Australasian Conference on Information Systems 2019*.

Utamima, A., Reiners, T., & Ansaripoor, A. H. (2019b). Evolutionary Estimation of Distribution Algorithm for Agricultural Routing Planning in Field Logistics. *Procedia Computer Science*, *161*, 560–567. https://doi.org/10.1016/j.procs.2019.11.156

Utamima, A., Reiners, T., & Ansaripoor, A. H. (2019c). Optimisation of agricultural routing planning in field logistics with Evolutionary Hybrid Neighbourhood Search. *Biosystems Engineering*, *184*, 166–180. https://doi.org/10.1016/j.biosystemseng.2019.06.001

Utamima, A., Reiners, T., & Ansaripoor, A. H. (2020). Automation in Agriculture: A Case Study of Route Planning Using an Evolutionary Lovebird Algorithm. *Proceedings of the 2020 12th International Conference on Computer and Automation Engineering*, 13–17. https://doi.org/10.1145/3384613.3384621

Vaessens, R. J. M., Aarts, E. H. L., & Lenstra, J. K. (1998). A local search template. *Computers & Operations Research*, *25*(11), 969–979. https://doi.org/https://doi.org/10.1016/S0305-0548(97)00093-2

Valente, J., Cerro, J. Del, Barrientos, A., & Sanz, D. (2013). Aerial coverage optimization in precision agriculture management: A musical harmony inspired approach. *Computers and Electronics in Agriculture*, *99*, 153–159. https://doi.org/10.1016/j.compag.2013.09.008

Vidal, T., Laporte, G., & Matl, P. (2019). A concise guide to existing and emerging vehicle routing problem variants. *European Journal of Operational Research*, *xxxx*, 1–16. https://doi.org/10.1016/j.ejor.2019.10.010

Voratas Kachitvichyanukul. (2012). Comparison of Three Evolutionary Algorithms : GA, PSO, and DE. *Industrial Engineering & Management Systems*, *11*(3), 215–223.

Wang, Z., Ren, X., Ji, Z., Huang, W., & Wu, T. (2019). A novel bio-heuristic computing algorithm to solve the capacitated vehicle routing problem based on Adleman–Lipton model. *Biosystems*. https://doi.org/10.1016/j.biosystems.2019.103997

Yu, X., & Gen, M. (2010). *Introduction to Evolutionary Algorithms*. Springer.

Zhou, K., Leck Jensen, A., Sørensen, C. G., Busato, P., & Bothtis, D. D. (2014). Agricultural operations planning in fields with multiple obstacle areas. *Computers and Electronics in Agriculture*, *109*, 12–22. https://doi.org/10.1016/j.compag.2014.08.013

APPENDICES

Appendix A Journal Publication (Biosystems Engineering)

This part contains details of the author's journal article in Biosystems Engineering, vol 184, 2019, Elsevier. The full article can be found in this link: https://www.sciencedirect.com/science/article/abs/pii/S153751101831198X

A.1 Attribution Statement

Title : Optimisation of agricultural routing planning in field logistics with Evolutionary Hybrid Neighbourhood Search

	Conception and Design	Collect Data	Identify Math Model	Build the Algorithm	Experi ments	Analyse Result	Conclu sion	Final Approval
Co-Author 1 (Amalia Utamima)	✔	✔	✔	✔	✔	✔	✔	✔
Co Author 1 Acknowledgment: I acknowledge that these represent my contribution to the above research output Signed: [redacted]								
Co-Author 2 (Torsten Reiners)	✔		✔				✔	✔
Co Author 2 Acknowledgment: I acknowledge that these represent my contribution to the above research output Signed: [redacted]								
Co-Author 3 (Amir H. Ansaripoor)			✔				✔	✔
Co Author 3 Acknowledgment: I ackn[redacted]se represent my contribution to the above research output Signed [redacted]								

A.2 Copyright Permission

Copyright Clearance Center RightsLink®

Home Create Account Help LIVE CHAT

Title: Optimisation of agricultural routing planning in field logistics with Evolutionary Hybrid Neighbourhood Search
Author: Amalia Utamima,Torsten Reiners,Amir H. Ansaripoor
Publication: Biosystems Engineering
Publisher: Elsevier
Date: August 2019
© 2019 IAgrE. Published by Elsevier Ltd. All rights reserved.

LOGIN
If you're a copyright.com user, you can login to RightsLink using your copyright.com credentials.
Already a RightsLink user or want to learn more?

Please note that, as the author of this Elsevier article, you retain the right to include it in a thesis or dissertation, provided it is not published commercially. Permission is not required, but please ensure that you reference the journal as the original source. For more information on this and on your other retained rights, please visit: https://www.elsevier.com/about/our-business/policies/copyright#Author-rights

BACK CLOSE WINDOW

A.3 Capture of Article

BIOSYSTEMS ENGINEERING 184 (2019) 166–180

Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/issn/15375110

Research Paper

Optimisation of agricultural routing planning in field logistics with Evolutionary Hybrid Neighbourhood Search

Amalia Utamima [a,b], *Torsten Reiners* [a,*], *Amir H. Ansaripoor* [a]

[a] *Curtin University, Australia*
[b] *Institut Teknologi Sepuluh Nopember, Indonesia*

ARTICLE INFO

Article history:
Received 6 December 2018
Received in revised form
7 May 2019
Accepted 2 June 2019
Published online 29 June 2019

Keywords:
Agriculture
Routing planning
Metaheuristic algorithm
Evolutionary Hybrid Neighbourhood Search (EHNS)

The optimisation of the agricultural process has gained importance over the past years as a means of increasing harvest yield, reducing cost and time required to maintain and harvest the fields, and maintaining economic and environmental sustainability. This research focuses on agricultural routing planning (ARP) for farmers' fields. The objective is to minimise the intra-field distance of the agricultural machine(s) when traversing all tracks, using an Evolutionary Hybrid Neighbourhood Search (EHNS) to solve different scenario problems. To obtain datasets for the agricultural routing problem, we gathered data from previous publications describing different fields. A mathematical model representing the optimisation of these datasets is also provided. The experimental results conclude that EHNS can either out-perform or obtain the same best solution as other algorithms in the literature. Among 9 problem sets, this study could find for 56% of the cases an improved combination of tracks saving an average of 10.68% non-working distance compared to other algorithms. The results also show that EHNS successfully gets the best objective function and the fastest convergence speed compared with the published algorithms.

1. Introduction

The agricultural sector is facing an increasing number of challenges regarding logistics. Driven by climate change and expectations for sustainable operations, optimisation of resource utilisation (i.e., water, fertiliser and fuel) and of the time needed to manage and harvest a growing size of land has become an integral part of farm operations. Environmental changes affect the timeframe to apply water and fertiliser or harvest a crop, while cost and reduction in CO_2 emissions require the optimisation of fleet size and corresponding utilisation by minimising the driven distance (Al-Zamil & Saudagar, 2018).

Optimisation of logistics in the agriculture supply chain has been widely studied (Borodin, Bourtembourg, Hnaien, & Labadie, 2016); yet the field of logistics seems to be limited to isolated case studies that explore optimisation of small and real-world examples. Studies on in-field logistics mostly do not provide sufficient in-depth analysis to generalise the research and develop algorithms across multiple cases. In particular, the lack of a benchmarking compilation of datasets limits the

* *Corresponding author.*
E-mail address: t.reiners@curtin.edu.au (T. Reiners).
https://doi.org/10.1016/j.biosystemseng.2019.06.001

Nomenclature

G	Graph $G = \{N, A\}$ representing the field's layout
N	Set of nodes in Graph G
A	Set of arcs in Graph G
S	Subgraph of Graph G, $\forall S \subseteq N$
n	Number of tracks in the field
i, j	Nodes indices ($i, j = (0), 1, 2, \ldots, n$), 0 is the Depot
T	Set of tracks ($t = 1, \ldots, n$)
l_t	Length of track t
M	m Set of machines available at the Depot Machine m, $m \in M$
r	The minimum turning radius of a machine
ω	The effective operating width of a machine
d_{ij}	The manoeuvre's degree between track i and j
θ	The degree of tilt between the previous and next track, note that $0 < \theta \leq 90$
Q	The maximum distance that can be travelled by a machine(s) (the machines are assumed homogeneous)
$\Omega(d_{ij})$	The bulb type of manoeuvre turn executed by a machine in the headland area
$\Pi(d_{ij})$	The flat type of manoeuvre turn executed by a machine in the headland field area
$\Pi\Theta(d_{ij})$	The flat type of manoeuvre turns with θ degree, executed by a machine in the headland field area if the previous track and the next track are aligned with an angle θ
$\Omega\Theta(d_{ij})$	The bulb type of manoeuvre turns with θ degree, executed by a machine in the headland field area if the previous track and the next track are aligned with an angle θ
x_{ij}^{m}	The decision variable, equal to 1 if machine m moves from node i to node j ($i, j \in A$), otherwise it is equal to 0
ARP	Agricultural routing planning
EHNS	Evolutionary Hybrid Neighbourhood Search
TS	Tabu Search
GA	Genetic Algorithm
ACO	Ant Colony Optimisation

comparability of algorithms and the promotion of the transfer of findings to other cases. Examples of cases in the literature are, among others, forage harvester routes (Cerdeira-Pena, Carpente, & Amiama, 2017), path planning for agricultural transport units (Jensen et al., 2012) and optimisation of the rice harvesting operation (He, Li, Zhang, & Wan, 2018).

Route planning in field operations is one of the core tasks in agricultural machinery management (Bochtis, Sørensen, & Busato, 2014). This can include the composition and size of the fleet, that is, vehicle types and mode of operation and consumption (Jensen, Bochtis, & Sørensen, 2015); the managed tasks, that is, seeding, applying fertilisers, watering, soil treatment and harvesting (Bochtis, Sørensen, & Vougioukas, 2010); and optimised operations with respect to time and cost (Bochtis et al., 2015; Seyyedhasani & Dvorak, 2018a). The optimised operation includes agricultural routing planning (ARP), which is employed to optimise the logistics of the primary agricultural machines (hereinafter 'machines') concerning their movements in the field and other farm locations, such as machine shed, silos or maintenance areas.

Agricultural machines can be classified into primary and supporting units. According to the recognised definitions (Bochtis & Sørensen, 2009; Bochtis et al., 2014), a 'primary unit' refers to a machine that performs the main task, while a 'supporting unit' refers to a machine supporting one or more primary units. The machines are restricted by, among other things, their capacity to hold harvests or products to apply on the field, size, turning radius, fuel consumption, CO_2 emissions, operator assignment, other machines' dependencies to operate together, time windows of operation and cost structure. Several researchers have developed initial models to describe ARP, yet the models are generally limited to a basic consideration of constraints (Bochtis & Vougioukas, 2008; Seyyedhasani & Dvorak, 2017).

This paper focuses on the in-field routing optimisation of machines for rectangular or non-convex field contexts. An illustration of the problem is shown in Fig. 1. The parallel tracks of a given, possibly different, length must be traversed by a given number of machines with a defined capacity. The capacity refers to the maximum volume of a machine to be filled with harvested crops. The tracks have a so-called headland area at each end to allow machines to perform manoeuvres to change direction and shift to another track. The objective of the ARP is the determination of a sequence of tracks to minimise the cost of operations, which are here equal to the travelled distance. Note that the ARP can be considered an NP-hard problem, requiring the application of heuristics to find near-optimal solutions (Marinakis, Migdalas, & Sifaleras, 2017).

Thus far, the research focus of several studies has been on real-world scenarios and the application of standard algorithms rather than considering specific ARP configurations in algorithm development (Jensen, Bochtis, et al., 2015; Seyyedhasani & Dvorak, 2018b). This paper offers two major contributions. First, this research brings together several datasets from previous independent real-world studies and provides a general mathematical model to represent them. Second, this study builds a new hybrid algorithm, Evolutionary Hybrid Neighbourhood Search (EHNS), as a competitive alternative to previous algorithms from the ARP literature.

The paper continues with an overview of the literature and algorithms relevant to ARP in Section 2. Based on this, we define the problem, mathematical model and the developed algorithm, EHNS, in Section 3. In Section 4, we list the experimental results, including datasets descriptions, parameter settings and numerical results. Section 4 also presents the proposed algorithm's performance compared to other algorithms used by various authors. Finally, the paper concludes in Section 5 by providing an outlook on future research to develop ARP further.

2. Literature review

2.1. Agricultural routing planning

Coverage path planning (CPP) is a problem that decides a route that passes over all points of a region while avoiding barriers

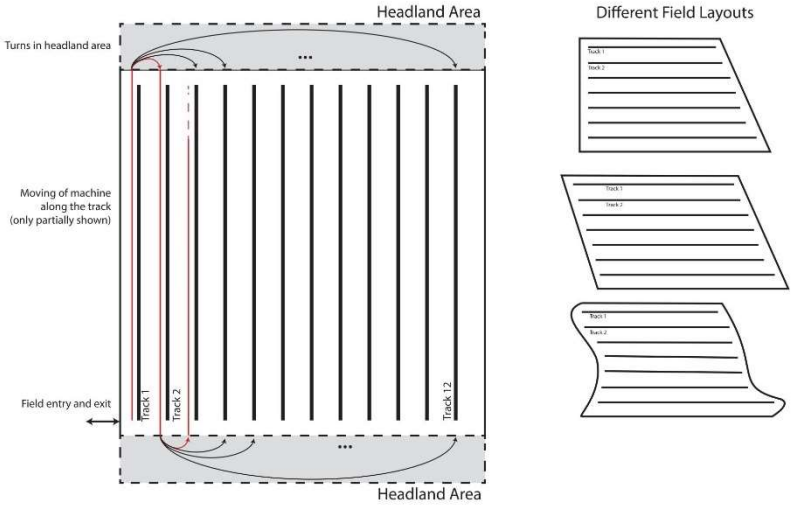


Fig. 1 – Illustration of an agricultural field with headland area. The arrow shows the possible movement of machines from one track to another.

(Galceran & Carreras, 2013). The form of CPP used in field logistics can also be considered alongside the concept of the travelling salesman problem (Bochtis & Sørensen, 2009), the multi-vehicle routing problem (Burger, Huiskamp, & Keviczky, 2013), capacitated field operation (Jensen and Bochtis, et al., 2015; Jensen, Nørremark, Busato, Sørensen, & Bochtis et al., 2015) and aerial coverage optimisation (Valente, Cerro, Barrientos, & Sanz, 2013). CPP has been explored in a wide range of fields in agriculture (Oksanen, 2007) and turning path generation (Backman, Piirainen, & Oksanen, 2015; Jin & Tang, 2010).

In the agriculture area, CPP can be understood as ARP. The ARP problem involves the minimisation of the distance travelled by machines employed to cover all tracks (Utamima, Reiners, Ansaripoor, & Seyyedhasani, 2018, pp. 261–283). This fundamental problem has been modified and extended regarding the objectives (e.g., optimisation of time, multiple machines, minimisation of the input cost) and constraints (e.g., limited machine capacity, multiple fields or obstacles).

Examples for the modification of the objective function can be found in Bochtis and Vougioukas (2008), Backman et al. (2015) and Seyyedhasani and Dvorak (2018b). Seyyedhasani and Dvorak (2018b) used multiple machines and modified the objective function by minimising the total finish time of every machine. Backman et al. (2015) generated smooth turning in the agricultural field, while Bochtis and Vougioukas (2008) minimised the non-working distance of machines in an agricultural field.

Variations have been introduced by several authors, among them, Seyyedhasani and Dvorak (2018a); Conesa-Muñoz, Bengochea-Guevara, Andujar, and Ribeiroet al. (2016), Conesa-Muñoz, Pajares, and Ribeiro et al. (2016); Gracia, Velázquez-Martí, and Estornell (2014); Zhou, Leck Jensen, Sørensen, Busato, and Bothtis (2014) and Bakhtiari, Navid, Mehri, Berruto, and Bochtis (2013). Seyyedhasani and Dvorak (2018a) worked on the routing of machines in agriculture as a dynamic multiple depot vehicle routing problem. Meanwhile, Bakhtiari et al. (2013) introduced a capacity constraint that refers to the capacity limit of each machine to be filled with harvested crops. Bakhtiari et al. (2013) examined an extension of the problem whereby the harvester unloads at a stationary facility located outside the field area, while Conesa-Muñoz and Bengochea-Guevara, et al. (2016) and Conesa-Muñoz and Pajares, et al. (2016) defined a new problem that considers input cost, the machine's features and the possibility of tank refilling for every machine. In contrast, Zhou et al. (2014) focused on a field containing multiple obstacles, while Gracia et al. (2014) applied the routing problem concept to biomass transportation.

Optimisation of the headland distance was first introduced by Bochtis and Vougioukas (2008). Here, the objective is the minimisation of the distance travelled in the headland area. This distance is labelled 'non-working distance' since the machine is not performing an agricultural operation when making the turning manoeuvres (Bochtis & Vougioukas, 2008). Hameed, Bochtis, and Sorensen (2011) continued the research by optimising the driving angle. Valente et al. (2013) and Barrientos et al. (2011) also used the concept of coverage planning for agricultural machines in the field. Their coverage planning was then extended using capacity constraints (Jensen and Bochtis, et al., 2015) and the navigation of service units (Jensen et al., 2012). The model was adapted by Conesa-

Muñoz and Bengochea-Guevara, et al. (2016) and Conesa-Muñoz and Pajares, et al. (2016) for a weed-killing problem and by Seyyedhasani and Dvorak (2018b) for reducing the field work time. It was also applied to orchard operations (Bochtis et al., 2015) and a potato cultivation system (Zhou, Jensen, Bochtis, & Sørensen, 2015).

2.2. *Metaheuristics algorithms for agricultural routing planning*

Several metaheuristic methods have been proposed to address the CPP and agricultural field logistics problem. The most common algorithms applied to the problem are Genetic Algorithm (GA; Gracia et al., 2014; Hameed et al., 2011), Ant Colony Algorithm (ACO; Bakhtiari et al., 2013; Zhou et al., 2014), Simulated Annealing (SA; Conesa-Muñoz and Pajares et al., 2016), Harmony Search (HS; Valente et al., 2013), Particle Swarm Optimisation (PSO; Sethanan & Neungmatcha, 2016) and Tabu Search (TS; Seyyedhasani & Dvorak, 2017; 2018a).

Algorithm selection is made by considering the objective of the problem and configuration of the field. Gracia et al. (2014) aimed for a decrease in the total distance travelled in biomass transportation by applying a method utilising GA and local search to adjust exploration and exploitation of the method. Meanwhile, Zhou et al. (2014) utilised ACO to create good solutions proficiently in a field with obstacles contrasted with manual operations by farmers. Another effective approach was shown by Sethanan and Neungmatcha (2016) who employed PSO with a different structure for route planning in sugarcane field operation; it provided outcomes competitive with basic PSO and GA.

Seyyedhasani and Dvorak (2017, 2018a) used TS to decrease the field work time and dynamic routing of multiple machines in agricultural operations. TS was selected for its capacity to compel the improvement technique to widen the search and prevent local optima. The outcomes demonstrate that in non-curved fields, TS can enhance the effective field capacity more successfully than the Clarke-Wright method (Seyyedhasani & Dvorak, 2017).

SA hybridised with Mix-opt (mutation operators) has been used to plan the route of autonomous vehicles in herbicide application (Conesa-Muñoz and Bengochea-Guevara, et al., 2016 and Conesa-Muñoz and Pajares, et al,. 2016). Valente et al. (2013) used the HS method to diminish the time consumed by coverage optimisation in vineyard parcels. The HS method improved the results of the previous technique by Barrientos et al. (2011).

To summarise, we can state the following research gaps, as already indicated in the introduction. First, the ARP literature covers individual cases with no comparative benchmarks across multiple datasets and algorithms for ARP. Second, most previous studies used established metaheuristic methods for ARP. Our experiments show that the solutions of several datasets in previous studies were not optimally solved. Thus, there is a need to develop and apply an improved algorithm, not only to enhance the solution quality but also to suggest a general approach to ARP. Hence, the spread of algorithms used across various research groups could be lowered by developing an advanced method to solve ARP.

To compare the previous methods, we gathered datasets from previous publications describing different fields' layouts. The layouts were derived either from those given in the literature or by requesting them from previous authors. The results were compared using the same objective function. The collected datasets are presented in Section 4.1.

3. Agricultural routing problem

3.1. *Problem description*

Figure 2(a) shows a field with tracks in crops that need to be harvested. In ARP, the tracks represent vertices in a graph, which must be visited by a machine (Utamima et al., 2018, pp. 261–283). The arcs connecting two vertices represent ways for the machine to move from one track to another track. In some cases, the machines start and end at a specific location outside of the tracks, such as the depot of the farm. Our objective is to find the minimal length of all trips for a set of machines to cover all tracks.

Regarding capacity constraints, optimisation will assign machines to the given routes based on their capacities. For instance, if a machine cannot hold any more crop, it must return to the depot to drop off the harvested crop before starting the next tour. In this research, we assume a uniform amount of harvest per travelled distance. Thus the machine capacity can be defined by the maximum distance it can travel on one tour.

Figure 2(b) describes an example of two possible solutions to the problem. The first option (dotted directed edges) shows a tour with one machine being able to harvest the whole field. The second option uses two machines, each of which covers a part of the graph. One machine covers the green part (Tracks 1–7) while the other covers the remaining tracks (Tracks 8–12).

Figure 3 illustrates the four kinds of machine manoeuvres based on Jin and Tang (2010) that are considered in this study: bulb (a), flat (b), flatΘ (c) and bulbΘ (d). The condition of the flat and bulb manoeuvres (Fig. 3(a, b)) are based on Bochtis and Vougioukas (2008), as illustrated in Fig. 4. The left side of Fig. 4 shows the connection of operating width (ω) and the minimum turning radius of a machine (r), while on the right side is a corresponding graph that represents each track as a node on the graph. Since $\omega > r$ and $r > \omega/2$, the flat (Π) turn can be made when a machine is skipping a minimum of one track. Otherwise, the bulb (Ω) turn will be performed. The details of the manoeuvres' condition are provided in Constraint (3) in the mathematical model in Section 3.2. Figure 3(c, d) shows the flatΘ and bulbΘ turns that occur when the previous track and the next track are aligned with an acute angle $\theta < 90$.

3.2. *Mathematical model*

The mathematical model to solve the collected datasets is formulated as a binary integer programing model. The group of tracks in a field are assumed to be a set of nodes and arcs in Graph G. A detailed description of parameters is given in the Nomenclature. The binary decision variables (x_{ij}^{m}) indicate the movement of the machine(s) between two nodes.

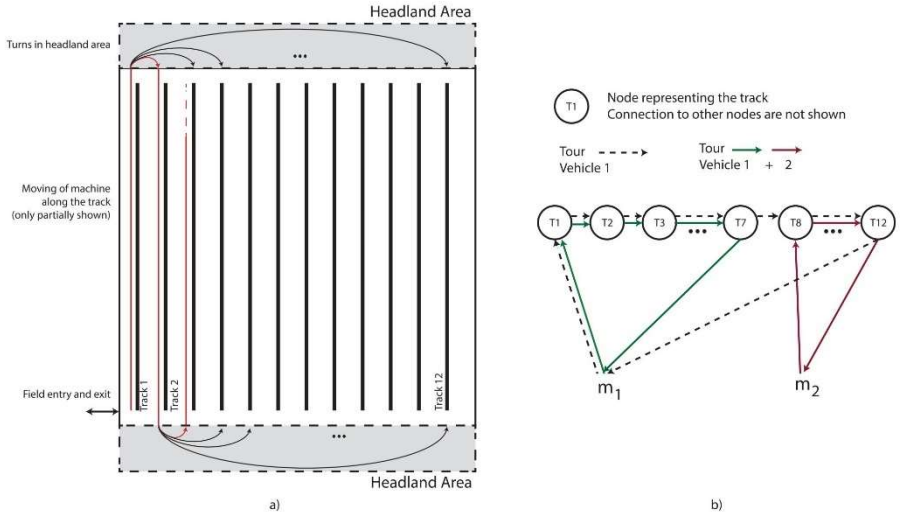


Fig. 2 – Illustration of ARP: (a) field with 12 tracks; (b) possible solutions to the problem.

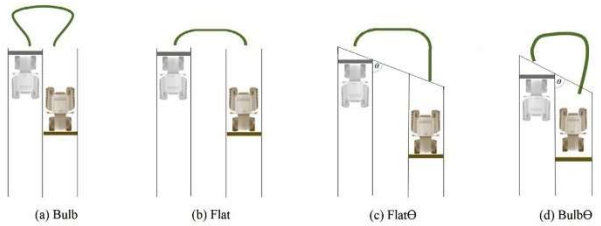


Fig. 3 – Manoeuvre types considered in this study. The brown machine represents the current location of the machine while the grey machine represents the previous location of the machine. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

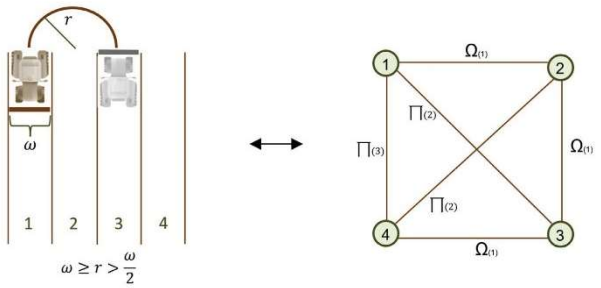


Fig. 4 – Illustration of the combination of operating width (ω) and minimum turning radius (r) (left side) and the associated turning types of the illustrated problem (right side).

The objective function (Equation (1)) minimises the non-working distance of the machines. The total distance (Equation (2)) is a sum of the total tracks' length and the minimisation of the non-working distance of the machines. The non-working distance refers to the distance that is traversed by a machine when moving to the headland areas or specific locations outside the field, for example, the warehouse or garage. Constraints (3)–(7) specify the four different manoeuvres: flat (Π), bulb (Ω), flatΘ (ΠΘ) and bulbΘ (ΩΘ). Constraints (8)–(9) ensure that every node is visited only once by the machine. Constraint (10) guarantees that if a machine enters a node, it will also leave that node again (Flow constraint). Constraint (11) excludes disjoint sub-tours from a solution. Constraint (12) restricts the maximum distance for a machine. The last constraint (13) specifies that the decision variable is a binary number.

$$z = \min\left(\Sigma_{i\in N}\Sigma_{j\in N}\Sigma_{m\in M} d_{ij}.x_{ij}^{m}\right) \tag{1}$$

$$Total\ Distance = \Sigma_{t\in T} l_t + \min\left(\Sigma_{i\in N}\Sigma_{j\in N}\Sigma_{m\in M} d_{ij}.x_{ij}^{m}\right) \tag{2}$$

s.t.

$$d_{ij} = \begin{cases} \Pi(i,j), & if\,|i-j| \le \dfrac{2r}{\omega} \ \wedge\ \theta = 90 \\ \Omega(i,j), & if\,|i-j| > \dfrac{2r}{\omega} \ \wedge\ \theta = 90 \\ \Pi\Theta(i,j), & if\,|i-j| \le \dfrac{2r}{\omega} \ \wedge\ \theta < 90 \\ \Omega\Theta(i,j), & if\,|i-j| > \dfrac{2r}{\omega} \ \wedge\ \theta < 90 \end{cases} \tag{3}$$

$$\Pi(i,j) = |i-j|.\omega + (\pi - 2)r \tag{4}$$

$$\Omega(i,j) = r\left(3\pi - 4\sin^{-1}\left(\frac{2r + |i-j|.\omega}{4r}\right)\right) \tag{5}$$

$$\Pi\Theta(i,j) = |i-j|.\omega(1 + \cot\theta) + r(\pi - 2) \tag{6}$$

$$\Omega\Theta(i,j) = \pi r + \frac{4r^2 - 4\omega r + \omega^2\cot^2\theta + \omega^2}{4r - 2\omega} \times \sin^{-1}\frac{4r\omega\cot\theta - 2\omega^2\cot\theta}{4r^2 - 4\omega r + \omega^2\cot^2\theta + \omega^2} \tag{7}$$

$$\Sigma_{m\in M}\Sigma_{i\in N} x_{ij}^{m} = 1, \quad i,j \ne 0,\ j\in N : i \ne j \tag{8}$$

$$\Sigma_{m\in M}\Sigma_{j\in N} x_{ij}^{m} = 1, \quad i,j \ne 0,\ i\in N : i \ne j \tag{9}$$

$$\Sigma_{m\in M}\Sigma_{i\in N} x_{ij}^{m} = \Sigma_{m\in M}\Sigma_{j\in N} x_{ji}^{m}, \quad i,j\in N \tag{10}$$

$$\Sigma_{i\in S}\Sigma_{j\in S} x_{ij}^{m} \le \left\|S\right\| - 1,\ \forall S\subseteq N,\ \left\|S\right\| \ge 1, \quad m\in M \tag{11}$$

$$\Sigma_{i\in N}\Sigma_{j\in N} l_i x_{ij}^{m} < Q, \quad m\in M,\ i,j\in N \tag{12}$$

$$x_{ij}^{m} \epsilon \{0,1\} \tag{13}$$

3.3. Algorithm description

The proposed algorithm EHNS adapts an evolutionary technique and combines it with Mutation-based Neighbourhood Search and TS. In general, a starting population of the solution is modified over multiple generations by selecting individual solutions, which are mutated and improved by a hybridised algorithm to form the next generation. The mutation operator that acted as a neighbourhood search is a strong technique to obtain an effective solution (Conesa-Muñoz and Pajares, et al., 2016).

Figure 5 illustrates the general stages of EHNS. First, the parameters of the algorithm are initialised. The parameters are number of generations, population size and the dataset details (number of tracks, turning radius, tracks' width, and each tracks' length and coordinates). The iterations begin with the calculation of the objective function of every candidate solution. Then, the roulette wheel selection is performed. The roulette wheel selection assumes that the better fitted a candidate solution, the bigger the probability of its survival. The probability of selection in this method is proportional to the fitness of a candidate solution (Lipowski & Lipowska, 2012; Utamima, Pradina, Dini, & Studiawan, 2015).

The Mutation-based Neighbourhood Search follows the previous stage and runs for some iterations. Figure 6 shows the pseudocode for this Mutation-based Neighbourhood Search. The procedure is repeated for *max_iteration* times. In each iteration, the roulette wheel selection will choose a candidate to be applied with the mutation operator. Three kinds of mutation operators are used as illustrated in Fig. 7. For each operator, two points (*p1* and *p2*) are randomly picked between 1 and *n* (candidate's size) and one of the following operators is applied:

- *Flap:* flips the sequence between two points (Fig. 7(a))
- *Interchange:* swaps two elements (Fig. 7(b))
- *Slide:* moves the first point to the second point and slides forward the remaining points (starting from the first point + 1 until the second point) (Fig. 7(c)).

The new candidates from the neighbourhood search will replace the previous candidates in the replacement step. Next, the current best solution (*curBest*) from the Mutation-based Neighbourhood Search is checked to determine whether it is better than the previous best solution (*prevBest*) thus far. If so, then *prevBest* will be replaced by *curBest*. Otherwise, Tabu Search will explore the local neighbourhood around the solution *curBest*. The pseudocode for Tabu Search is listed in Fig. 8. First, the parameters, which are the tenure and the number of iterations, are initialised. Then the list *swap_list* is

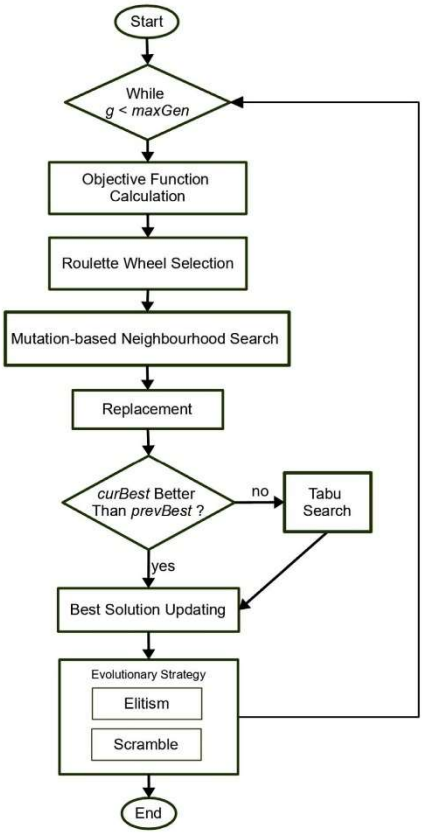


Fig. 5 – General flowchart of the Evolutionary Hybrid Neighbourhood Search.

constructed in every iteration. It contains a set of *moves* (the swap of two tracks) and the respective distances. Next, Tabu Search checks whether a *move* is tabu and whether the distance of every *move* (*tabu_dist*) is worse than *tabu_Sol*. A penalty is imposed if a *move* matches these conditions. Subsequently, Tabu Search updates the *tabu_List* and *tabu_Sol* as well. At the end of Tabu Search's iteration, a *tabu_Sol* will contain the best solution for Tabu Search. Then, *tabu_Sol* is compared with the *gBest* (the global best solution found so far in the EHNS's generations). If *tabu_Sol* obtains a better solution, then it will replace the value of *gBest*. Otherwise the *gBest* will remain the same.

Several evolutionary strategies have been proposed by previous researchers to retain the best solution while maintaining diversity in the population. This study applies evolutionary strategies called elitism (Nayeem, Rahman, & Rahman, 2014; Sivanandam & Deepa, 2007), which retain a group of the best solutions in the generations. Elitism records ten per cent of the best solutions in each generation to preserve them from extinction and copy this group to the new generation. Then, a randomisation technique (Haupt & Haupt, 2004) is also employed to diversify the population. The randomisation technique, called scramble, will bring a small group of solutions into a random order. This scrambled group is also copied to ten per cent of the new generation.

Figure 9 shows the last step of EHNS, which is the modification of the new generation that contains the evolutionary strategy. The number of solutions selected for both elitism and scramble is initially set to each ten per cent of the population (*ElitismSize* and *ScrambleSize*). Lines 2 to 5 in Fig. 9 list the elitism process. The process starts with obtaining the best candidate of the current generation (*best*). The process continues to check whether *best* is better than the current candidate with the highest fitness (*max(EliteGroup)*) in the *EliteGroup* (the group of best candidate solutions) and replaces it if all conditions are met. Eventually, the solutions in the *EliteGroup* replace the originally selected solutions. Next, the scramble process is applied, as shown in lines 9 to 11. The result is again copied back into the population, replacing the selected solutions.

4. Experimental results

4.1. Dataset description

Table 1 presents an overview of the nine datasets used in this study. The information about the datasets was either acquired through accessible public sources or by contacting the respective authors. Generally, the data differ regarding the number of tracks and the shape of the field.

Seven of these datasets are from four previous publications, while the last is an adapted version of the data of Problem 74nc. The second and third columns list the problem code and the number of tracks (problem size) in the field. The 'Tracks' Width & Length' column lists the width and the length of the tracks. The 'Shape' column indicates the form of the field: rectangular means that the field's shape is a rectangle, while non-convex refers to the irregular shape of the field. The 'Type of Manoeuvres' column refers to the type of manoeuvres that can be used in this particular dataset, while the 'Machines' column shows the number of machines that are used for the dataset. The 'Turning Radius' column refers to the machine's turning radius, and the last column lists the references to the data.

4.2. Parameter settings

The parameter configuration is obtained by conducting a two-level factorial design with four factors. Every factor incorporates high and low levels (Montgomery, 2013). The algorithm is then run ten times with varying random settings. Each factor is in the given range to calculate an average to

```
Procedure Mutation_based_Neighbourhood_Search {
   for 1 until max_iteration {
      Old ← Roulette_Wheel_Selection)
      k = rand*(3)
      switch k {
           [p1, p2] ← Select_2_different_points()
           case 1: New ← Flap(p1,p2)
           case 2: New ← Interchange(p1,p2)
           case 3: New ← Slide(p1,p2)
      }
   }
}
```

Fig. 6 – Pseudocode for the mutation-based neighbourhood search.

Old	1	2	3	4	5	6	7	8	9	10
			*				*			
New	1	2	4	5	6	7	3	8	9	10

(a)

Old	1	2	3	4	5	6	7	8	9	10
			*				*			
New	1	2	7	4	5	6	3	8	9	10

(b)

Old	1	2	3	4	5	6	7	8	9	10
			*			*				
New	1	2	6	5	4	3	7	8	9	10

(c)

Fig. 7 – Illustration of Flap (a), Interchange (b), and Slide (c) operators.

compensate for the non-deterministic nature of the algorithm (Guan & Lin, 2016). Table 2 provides the details of all factors and the range of each setting. The settings use n, which is the number of tracks in the field, to adapt to the size of the problem. The larger the size of the problem, the greater the number of iterations and the population size. The values in bold indicate the better setting for EHNS.

For several datasets, this study re-coded GA, TS, and ACO as the comparer algorithms. The number of generations or iterations of GA, TS and ACO are set to match those of EHNS. In GA, the crossover rate is set to 0.7 and the mutation rate is 0.3, while TS's tenure is set to $0.5n$ (Ou-Yang & Utamima, 2013). Meanwhile, the number of ants, the weight of pheromones and the evaporation rate in ACO are set to n, 1, and 0.01, respectively (Bakhtiari et al., 2013; Zhou et al., 2014). In this study, the local search is also embedded in ACO.

4.3. Results

Table 3 represents the experimental results of the EHNS algorithm in comparison with several state-of-the-art algorithms found in the agricultural routing optimisation literature. The algorithms used in previous studies are GA, Mix-opt + Simulated Annealing (MS), TS, and ACO in the second, third, fourth and fifth columns of Table 3. The 'Problem Code' column refers to the datasets that are explained in

```
Procedure Tabu_Search (Candidate_Solution) {
     Initialization of Parameters
     tabu_Sol ← Candidate_Solution
     for 1 until tabu_iteration {
         Construct swap_list
         Calculate tabu_dist for every move in swap_list
         if move is taboo and move's tabu_dist > tabu_Sol {
              give move penalty
         }
         Update tabu_List(tenure)
         Update tabu_Sol
     }
}
```

Fig. 8 – Pseudocode for tabu search.

```
Procedure Modify_New_Generation {
    best ← get_BestCandidate()
    if best < max(EliteGroup){
        EliteGroup[max(EliteGroup)] ← best
    }
    for i=1 until ElitismSize {
        New_Population[i] ← EliteGroup[i]
    }
    for j=i+1 until i+ScrambleSize {
        New_Population[j] ← rand(n)
    }
}
```

Fig. 9 – Pseudocode for the modification of the new generation in EHNS.

Section 4.1. The values shown in bold in Table 3 indicate the best (and the lowest values) in that row; '-' indicates that there is no reported solution; '*' explains that the results of the specific problems in GA, MS and TS are obtained from the references; and the remainder valued without '*' indicate that we re-coded the algorithms described in the corresponding reference.

Generally, EHNS successfully achieves either the lowest or the same objective function compared with other algorithms in the literature. As listed in Table 3, EHNS outperforms TS in six problems (20rt, 37nc, 90rt, 74nc, 74nc2 and H-shaped). EHNS obtains the same best solution as MS for the first four problems, while for Problem 37nc, EHNS introduces a new best solution. EHNS achieves better objective function in seven problem instances compared with ACO. Moreover, EHNS outperforms GA in all problems. In short, EHNS presents the new best solution in six of the nine datasets. Hence, this study could find for 56% of the cases an improved combination of tracks saving an average of 10.68% non-working distance compared to other algorithms.

In Table 3, TS is shown to reach the same objective function as EHNS in three problems; moreover, its solutions are better than GA and ACO in seven problems. ACO performs well in small-sized problems; however, its solution quality decreases in larger problems. GA can obtain better solutions than ACO in six problems; however, GA's solutions are not as good as TS's.

The results of Problems 37nc, 90rt, and 74nc2 are presented in detail in Figs. 10–12, respectively. Figure 10 shows the optimised path chosen by EHNS when the bulb turn is chosen for all the non-convex area of the field, while the flat turn dominates the rectangular section of the field. The bulbΘ turn requires less turning length than the bulb turn (Jin & Tang, 2010). In contrast, the flatΘ turn needs to skip one or more tracks, and its length is similar to the flat turn with the addition of the different alignment between two tracks. Therefore, the algorithm chooses the bulbΘ turn in the non-convex area because it tends to cover a shorter distance than the flatΘ.

Figure 11 shows the path planning for three machines in a rectangular field of 90rt. Most of the manoeuvres are flat turns. The colour differentiates the route for each machine. Figure 12 shows that 74nc2 path planning is the most complicated problem compared with others. This is because the machine needs to start and end at the depot and it considers the bulbΘ turn. The bulbΘ turn is predominant in this non-convex field.

Table 4 shows the comparative running times of GA, TS, ACO and EHNS in four problem instances (20rt, 37nc, 90rt and 74nc2). Generally, the running time of ACO is the fastest among the algorithms. As shown in Table 4, the running time

Table 2 – Factor settings for EHNS.

Factors	Levels	
	Low	High
Number of generations	30n	**40n**
Population size	3n	**5n**
Mutation-based Neighbourhood Search iteration	1.5n	**2.5n**
Tabu Search iteration	0.5n	**n**

The bold values indicate the chosen factors.

Table 1 – Description of the datasets.

No	Problem Code	Size	Tracks' Width & Length (metres)	Shape	Type of Manoeuvres	Machines	Turning Radius (metres)	Ref[a]
1	8rt	8	2.89 & 30	Rectangular	Flat, bulb	1 (single)	3.50	[1]
2	12rt_a	12	2.5 & 40					[2]
3	12rt_b	12	2.5 & 70					
4	20rt	20	2.5 & 80					
5	37nc	37	9 & vary	Non-convex	Flat, bulb, flatΘ		6.00	[2], [3]
6	90rt	90	10 & 100	Rectangular	Flat, bulb	3 (multiple and	7.00	[4]
7	74nc	74	19 & vary	Non-convex	Flat	homogeneous)	9.77	[5]
8	74nc2				Flat, bulb, flatΘ, bulbΘ			–
9	62Hs	62	2.5 & vary	Rectangular, H-shaped	Flat, bulb,	1 (single)	2.30	[6]

[a] [1] Bochtis and Vougioukas (2008); [2] Conesa-Muñoz and Bengochea-Guevara, et al. (2016) and Conesa-Muñoz and Pajares, et al. (2016); [3] Hameed et al. (2011); [4] Seyyedhasani and Dvorak (2017); [5] Seyyedhasani and Dvorak (2018a); [6] Oksanen (2007).

Table 3 – The non-working distance comparison.

No	Problem Code	Genetic Algorithm (GA)[1, 2]	Mix-opt + SA (MS) [1, 3]	Tabu Search (TS) [4]	Ant Colony Optimisation (ACO)	Proposed Algorithm: Evolutionary Hybrid Neighbourhood Search (EHNS)
1	8rt	95.767	[a]**94.439**	**94.439**	**94.439**	**94.439**
2	12rt_a	176.451	[a]**146.027**	**146.027**	**146.027**	**146.027**
3	12rt_b	166.451	[a]**145.602**	**145.602**	147.076	**145.602**
4	20rt	250.916	[a]**235.491**	245.916	269.642	**235.491**
5	37nc	[a]1142.474	[a]961.470	1088.188	1228.129	**958.930**
6	90rt	2791.680	–	2870.172	3987.434	**2658.474**
7	74nc	5212.590	–	[a]4416.300	5506.67	**3880.679**
8	74nc2	6064.112	–	5856.542	6911.151	**5197.349**
9	62Hs	562.414	–	559.914	579.914	**479.914**

–The results are not available in the references.
The bold values indicate the best solution in that row.
[a] The results of the specific problems in GA, MS and TS are obtained from [1] Conesa-Muñoz and Pajares et al. (2016), [2] Hameed et al. (2011), [3] Conesa-Muñoz and Bengochea-Guevara, et al. (2016) and Conesa-Muñoz and Pajares, et al. (2016) and [4] Seyyedhasani and Dvorak (2018a).

of GA depends on the size of the problem: the greater the number of tracks, the longer the time needed. In contrast, the running times of TS, ACO and EHNS in Problem 74nc2 are longer than in 90rt. This is because the neighbourhood search aspect checks every move's objective function, and the objective function of Problem 74nc2 contains the flatΘ and bulbΘ manoeuvres, which are more complex than the manoeuvre in Problem 90rt.

Figure 13 presents the graphs of the convergence process of EHNS, GA, TS and ACO when solving Problems 20rt (a), 37nc (b), 90rt (c), and 74nc2 (d). The X-axis of Fig. 13 shows the number of generations of the algorithms, while the Y-axis represents the objective function reached. In the four problems presented, EHNS always successfully achieves the best objective function compared with GA, TS and ACO from the beginning of the generation. TS is placed second in terms of solution quality while GA's solution is placed third. Although ACO's solutions are better than GA's at the beginning of generations (in Problems 20rt, 90rt and 74nc2), its solutions are defeated by GA at the end of the generations.

Figure 14 describes the graphs of the objective function (non-working distance) versus the time needed to achieve the objectives of EHNS, GA, TS and ACO in four problem instances (20rt, 37nc, 90rt and 74nc2). The X-axis of Fig. 14 shows the time, while the Y-axis represents the objective function reached. The time is shortened to ACO's running

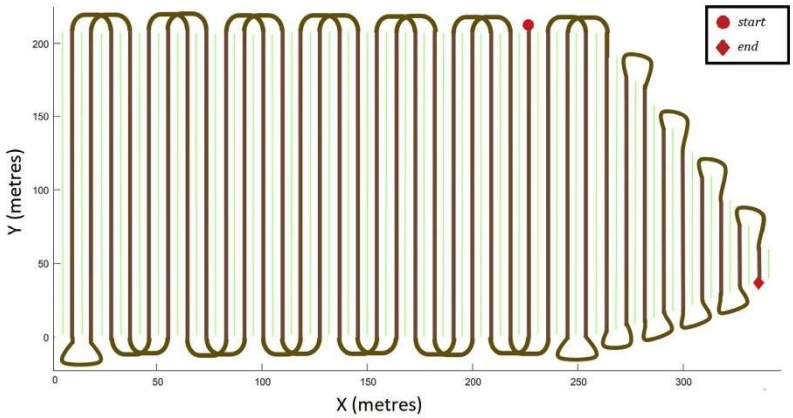


Fig. 10 – EHNS's manoeuvre results for the machine when solving Problem 37nc.

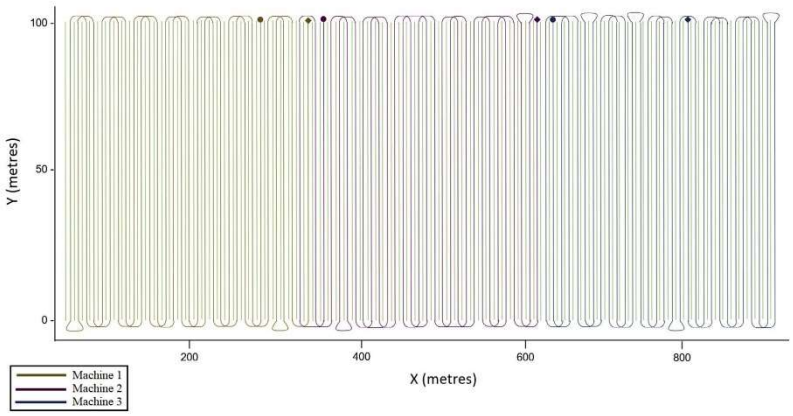


Fig. 11 – EHNS's manoeuvre results for the machine when solving Problem 90rt.

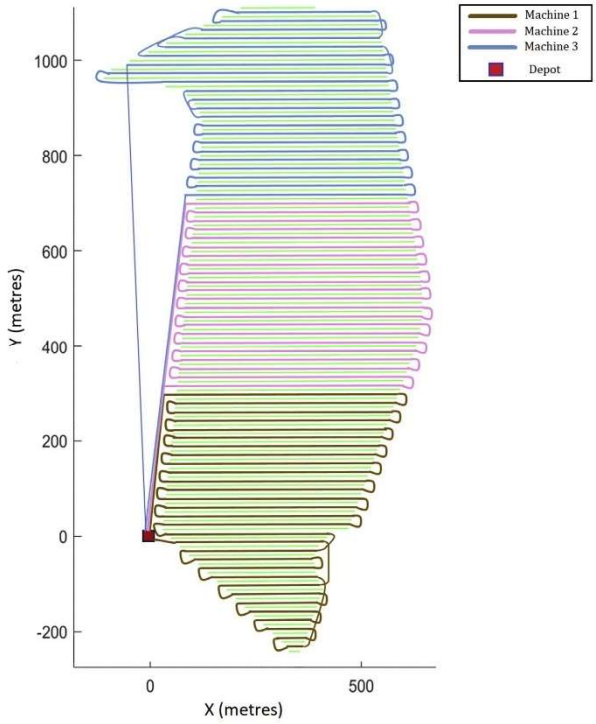


Fig. 12 – EHNS's manoeuvre results for the machine when solving Problem 74nc2.

Table 4 – Runtime comparison of GA, TS, ACO and EHNS.

Algorithm	Running Time (Seconds)			
	20rt	37nc	90rt	74nc2
Genetic Algorithm (GA)	3.551	8.273	23.064	20.463
Tabu Search (TS)	1.768	5.102	14.616	21.188
Ant Colony Optimisation (ACO)	1.649	4.995	13.238	16.308
Evolutionary Hybrid Neighbourhood Search (EHNS)	2.059	5.695	15.633	23.470

time, which has the shortest running time. For all cases, EHNS achieves the best objective function in the fastest time compared with other algorithms, while TS is placed second.

As shown in Fig. 13(a, c, d) and Fig. 14(a, c, d), TS (the blue dashed line) is often trapped in local optima and produces the same solution when other algorithms are progressing to reach better solutions. TS does sometimes manage to achieve a better solution in the middle of its running (Figs. 13(b) and 14(b)). In Figs. 13 and 14, ACO (the yellow striped and dotted line) can produce good solutions until the middle of its running; however, its quality decreases and it becomes trapped in the local optima after the middle of its iteration. GA (the orange dotted line) slowly progresses to obtain better solutions from the beginning to the end of its generation. However, GA's objective function still cannot defeat TS's (Problems 20rt, 37nc, and 74nc2) and EHNS's object function (all problems).

The given running time of TS shown in Fig. 14 acts similarly to that shown in Fig. 13, while ACO's objective function is better than GA's in Problems 20rt and 37nc. In less than 150 generations (Fig. 13(a)) and less than 0.4 s (Fig. 14(a)), EHNS starts to converge while obtaining the best solution to Problem 20rt. EHNS always reaches a better solution than the others in Problem 37nc, and EHNS reaches its best solution before reaching 1200 generations (Fig. 13(b)) in less than 2.5 s (Fig. 14(b)). Solving Problem 90rt, EHNS starts to converge in about 500 generations (Fig. 13(c)) and in less than 3 s (Fig. 14(c)) while maintaining the best solutions compared with other algorithms. Finally, EHNS always achieves a superior solution and converges in about 700 generations (Fig. 13(d)) and less than 6 s (Fig. 14(d)).

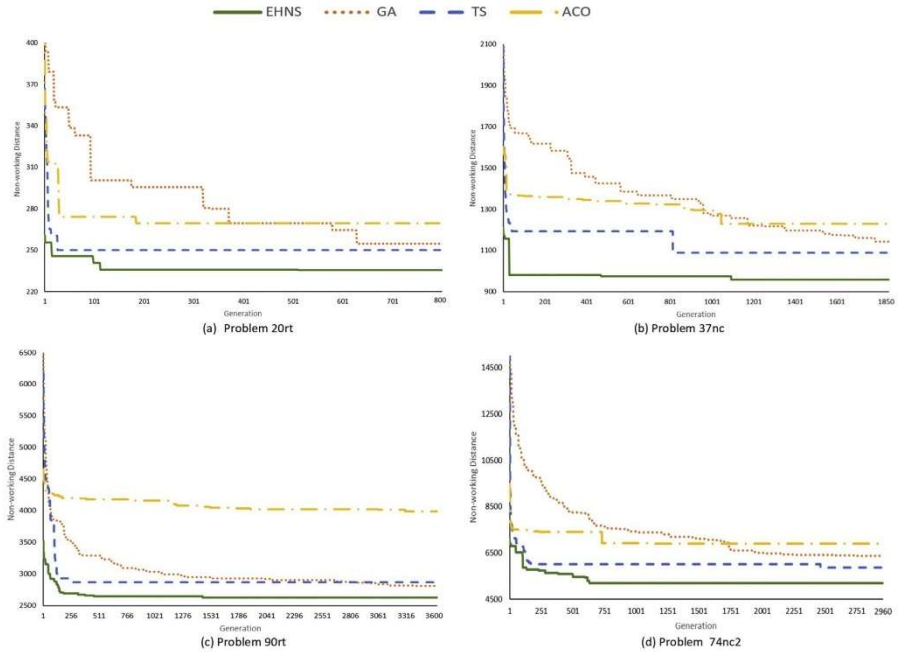


Fig. 13 – Graphs of the convergence process of GA, TS, ACO and EHNS solving Problems 20rt (a), 37nc (b), 90rt (c) and 74nc2 (d).

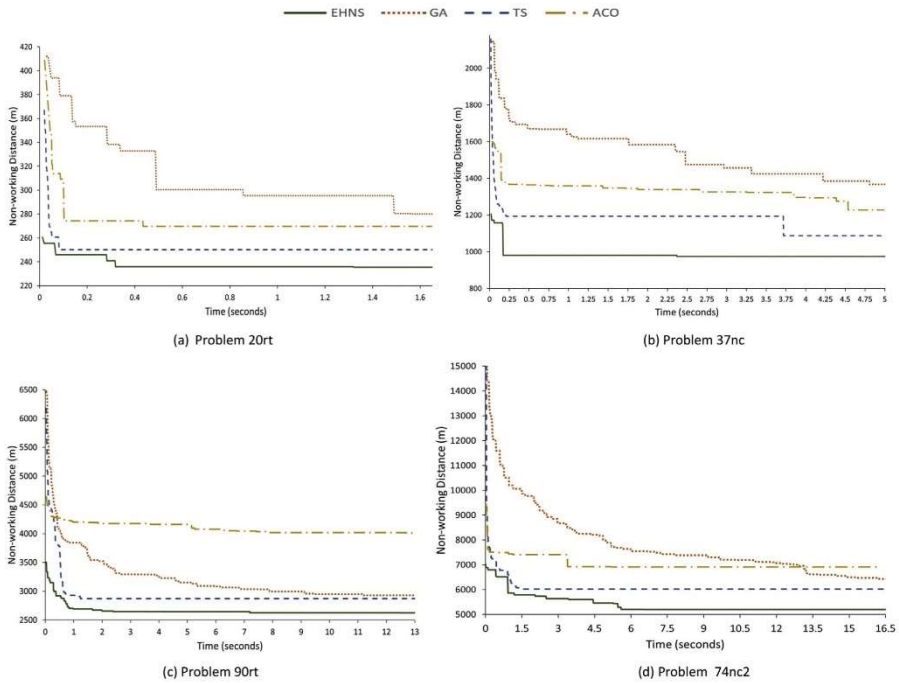


Fig. 14 – Graphs of the objective function (non-working distance) versus the running time of GA, TS, ACO and EHNS when solving Problems 20rt (a), 37nc (b), 90rt (c) and 74nc2 (d).

5. Conclusion and future research

This study is expected to improve the management of conventional machinery systems by producing better routing plans. The quantitative improvement with the reduction of resources such as machines, fuel and personnel can result in a competitive advantage and access to markets in a lower price segment. As a result, this contributes to addressing the sustainability issue in agriculture, taking into account the effect on the improvement of economic (reduction of cost) and environmental factors (reduced fuel consumption [energy saving] and consequently a decrease in CO_2 emissions).

The agriculture literature has introduced several cases of agricultural routing optimisations and provided solutions using different algorithms. Here, we have collected different datasets representing past studies and suggest the application of a new algorithm (EHNS) that outperforms the others used thus far. The algorithms used in studies from the literature are Genetic Algorithm, Mix-opt with Simulated Annealing, Tabu Search, and Ant Colony Optimisation. The experimental results show that EHNS can obtain a better solution for 5 of 9 problems set while achieving the same best-known optimal solutions for the rest of problem sets compared with the algorithms used in the previous literature. EHNS also successfully save the non-working distance by 10.68% in the improved cases. The results also illustrate that EHNS can maintain the best objective function and the fastest convergence speed compare with GA, TS and ACO.

The scope of this paper is limited to the application of the EHNS and several state-of-the-art algorithms in rectangular and non-convex fields in flat land. Future research needs to focus on (generated) datasets of different sizes, structures, and complexity to replicate particular field configurations. Another research path is the improvement of the EHNS algorithm for larger and more complex cases.

Acknowledgement

This works is sponsored by Indonesia Endowment Fund for Education (LPDP - Lembaga Pengelola Dana Pendidikan) of the Ministry of Finance of Indonesia. We thank the editor of *Biosystems Engineering* and the two anonymous reviewers for their valuable feedback to improve the quality of the paper.

REFERENCES

Al-Zamil, A., & Saudagar, A. K. J. (2018). Drivers and challenges of applying green computing for sustainable agriculture: A case study. *Sustainable Computing: Informatics and Systems*. https://doi.org/10.1016/J.SUSCOM.2018.07.008.

Backman, J., Piirainen, P., & Oksanen, T. (2015). Smooth turning path generation for agricultural vehicles in headlands. *Biosystems Engineering*, *139*, 76–86. https://doi.org/10.1016/J.BIOSYSTEMSENG.2015.08.005.

Bakhtiari, A., Navid, H., Mehri, J., Berruto, R., & Bochtis, D. D. (2013). Operations planning for agricultural harvesters using ant colony optimization. *Spanish Journal of Agricultural Research*, *11*(3), 652–660. https://doi.org/10.5424/sjar/2013113-3865.

Barrientos, A., Colorado, J., Cerro, J. Del, Martinez, A., Rossi, C., Sanz, D., et al. (2011). Aerial remote sensing in agriculture: A practical approach to area coverage and path planning for fleets of mini aerial robots. *Journal of Field Robotics*, *28*(5), 667–689. https://doi.org/10.1002/rob.20403.

Bochtis, D., Griepentrog, H. W., Vougioukas, S., Busato, P., Berruto, R., & Zhou, K. (2015). Route planning for orchard operations. *Computers and Electronics in Agriculture*, *113*, 51–60. https://doi.org/10.1016/j.compag.2014.12.024.

Bochtis, D. D., & Sørensen, C. G. (2009). The vehicle routing problem in field logistics part I. *Biosystems Engineering*, *104*(4), 447–457. https://doi.org/10.1016/j.biosystemseng.2009.09.003.

Bochtis, D. D., Sørensen, C. G. C., & Busato, P. (2014). Advances in agricultural machinery management: A review. *Biosystems Engineering*, *126*, 69–81. https://doi.org/10.1016/j.biosystemseng.2014.07.012.

Bochtis, D. D., Sørensen, C. G., & Vougioukas, S. G. (2010). Path planning for in-field navigation-aiding of service units. *Computers and Electronics in Agriculture*, *74*(1), 80–90. https://doi.org/10.1016/j.compag.2010.06.008.

Bochtis, D. D., & Vougioukas, S. G. (2008). Minimising the non-working distance travelled by machines operating in a headland field pattern. *Biosystems Engineering*, *101*(1), 1–12. https://doi.org/10.1016/j.biosystemseng.2008.06.008.

Borodin, V., Bourtembourg, J., Hnaien, F., & Labadie, N. (2016). Handling uncertainty in agricultural supply chain management: A state of the art. *European Journal of Operational Research*, *254*(2), 348–359. https://doi.org/10.1016/J.EJOR.2016.03.057.

Burger, M., Huiskamp, M., & Keviczky, T. (2013). Complete field coverage as a multi-vehicle routing problem. In *IFAC proceedings volumes (IFAC-PapersOnline)* (Vol. 4, pp. 97–102). https://doi.org/10.3182/20130828-2-SF-3019.00050.

Cerdeira-Pena, A., Carpente, L., & Amiama, C. (2017). Optimised forage harvester routes as solutions to a traveling salesman problem with clusters and time windows. *Biosystems Engineering*, *164*, 110–123. https://doi.org/10.1016/J.BIOSYSTEMSENG.2017.10.002.

Conesa-Muñoz, J., Bengochea-Guevara, J. M., Andujar, D., & Ribeiro, A. (2016a). Route planning for agricultural tasks: A general approach for fleets of autonomous vehicles in site-specific herbicide applications. *Computers and Electronics in Agriculture*, *127*, 204–220. https://doi.org/10.1016/j.compag.2016.06.012.

Conesa-Muñoz, J., Pajares, G., & Ribeiro, A. (2016b). Mix-opt: A new route operator for optimal coverage path planning for a fleet in an agricultural environment. *Expert Systems with Applications*, *54*, 364–378. https://doi.org/10.1016/j.eswa.2015.12.047.

Galceran, E., & Carreras, M. (2013). A survey on coverage path planning for robotics. *Robotics and Autonomous Systems*, *61*(12), 1258–1276. https://doi.org/10.1016/j.robot.2013.09.004.

Gracia, C., Velázquez-Martí, B., & Estornell, J. (2014). An application of the vehicle routing problem to biomass transportation. *Biosystems Engineering*, *124*, 40–52. https://doi.org/10.1016/J.BIOSYSTEMSENG.2014.06.009.

Guan, J., & Lin, G. (2016). Hybridizing variable neighborhood search with ant colony optimization for solving the single row facility layout problem. *European Journal of Operational Research*, *248*(3), 899–909. https://doi.org/10.1016/j.ejor.2015.08.014.

Hameed, I. A., Bochtis, D. D., & Sorensen, C. G. (2011). Driving angle and track sequence optimization for operational path planning using genetic algorithms. *Applied Engineering in Agriculture*, *27*(6), 1077–1086. https://doi.org/10.13031/2013.40615.

Haupt, R. L., & Haupt, S. E. (2004). *Practical genetic algorithms* (2nd ed.). Wiley-Interscience.

He, P., Li, J., Zhang, D., & Wan, S. (2018). Optimisation of the harvesting time of rice in moist and non-moist dispersed fields. *Biosystems Engineering*, *170*, 12–23. https://doi.org/10.1016/J.BIOSYSTEMSENG.2018.03.008.

Jensen, M. F., Bochtis, D., & Sørensen, C. G. (2015). Coverage planning for capacitated field operations, part II Optimisation.pdf. *Biosystems Engineering*, *139*, 149–164.

Jensen, M. A. F., Bochtis, D., Sørensen, C. G., Blas, M. R., & Lykkegaard, K. L. (2012). In-field and inter-field path planning for agricultural transport units. *Computers and Industrial Engineering*, *63*(4), 1054–1061. https://doi.org/10.1016/j.cie.2012.07.004.

Jensen, M. F., Nørremark, M., Busato, P., Sørensen, C. G., & Bochtis, D. (2015). Coverage planning for capacitated field operations, Part I: Task decomposition. *Biosystems Engineering*, *139*, 136–148. https://doi.org/10.1016/J.BIOSYSTEMSENG.2015.07.003.

Jin, J., & Tang, L. (2010). Optimal coverage path planning for arable farming on 2D surfaces. *Transactions of the ASABE*, *53*(1), 283–295. https://doi.org/10.13031/2013.29488.

Lipowski, A., & Lipowska, D. (2012). Roulette-wheel selection via stochastic acceptance. *Physica A: Statistical Mechanics and its Applications*, *391*(6), 2193–2196. https://doi.org/10.1016/J.PHYSA.2011.12.004.

Marinakis, Y., Migdalas, A., & Sifaleras, A. (2017). A hybrid particle swarm optimization – Variable neighborhood search algorithm for constrained shortest path problems. *European Journal of Operational Research*, *261*(3), 819–834. https://doi.org/10.1016/j.ejor.2017.03.031.

Montgomery, D. C. (2013). *Design and analysis of experiments* (Vol. 8). New York: John Wiley & Sons.

Nayeem, M. A., Rahman, M. K., & Rahman, M. S. (2014). Transit network design by genetic algorithm with elitism. *Transportation Research Part C: Emerging Technologies*, *46*, 30–45. https://doi.org/10.1016/J.TRC.2014.05.002.

Oksanen, T. (2007). *Path planning algorithms for agricultural machines* (Vol. IX). Helsinki University of Technology Automation Technology Laboratory Series A: Research Reports. Retrieved from https://ecommons.cornell.edu/handle/1813/10604.

Ou-Yang, C., & Utamima, A. (2013). Hybrid estimation of distribution algorithm for solving single row facility layout problem. *Computers & Industrial Engineering*, *66*(1), 95–103. https://doi.org/10.1016/j.cie.2013.05.018.

Sethanan, K., & Neungmatcha, W. (2016). Multi-objective particle swarm optimization for mechanical harvester route planning of sugarcane field operations. *European Journal of Operational Research*, *252*(3), 969–984. https://doi.org/10.1016/J.EJOR.2016.01.043.

Seyyedhasani, H., & Dvorak, J. S. (2017). Using the Vehicle Routing Problem to reduce field completion times with multiple machines. *Computers and Electronics in Agriculture*, *134*, 142–150. https://doi.org/10.1016/j.compag.2016.11.010.

Seyyedhasani, H., & Dvorak, J. S. (2018a). Dynamic rerouting of a fleet of vehicles in agricultural operations through a dynamic multiple depot vehicle routing problem representation. *Biosystems Engineering*, *171*, 63–77. https://doi.org/10.1016/J.BIOSYSTEMSENG.2018.04.003.

Seyyedhasani, H., & Dvorak, J. S. (2018b). Reducing field work time using fleet routing optimization. *Biosystems Engineering*, *169*, 1–10. https://doi.org/10.1016/j.biosystemseng.2018.01.006.

Sivanandam, S. N., & Deepa, S. N. (2007). *Introduction to genetic algorithms*. Springer.

Utamima, A., Pradina, K. R., Dini, N. S., & Studiawan, H. (2015). Distribution route optimization of Gallon water using genetic algorithm and tabu search. *Procedia Computer Science*, *72*, 503–510. https://doi.org/10.1016/J.PROCS.2015.12.132.

Utamima, A., Reiners, T., Ansaripoor, A., & Seyyedhasani, H. (2018). The agricultural routing planning in field logistics. In *Contemporary approaches and strategies for applied logistics*. IGI Global https://doi.org/10.4018/978-1-5225-5273-4.ch010.

Valente, J., Cerro, J. D.el, Barrientos, A., & Sanz, D. (2013). Aerial coverage optimization in precision agriculture management: A musical harmony inspired approach. *Computers and Electronics in Agriculture*, *99*, 153–159. https://doi.org/10.1016/j.compag.2013.09.008.

Zhou, K., Jensen, A. L., Bochtis, D. D., & Sørensen, C. G. (2015). Quantifying the benefits of alternative fieldwork patterns in a potato cultivation system. *Computers and Electronics in Agriculture*, *119*, 228–240. https://doi.org/10.1016/j.compag.2015.10.012.

Zhou, K., Leck Jensen, A., Sørensen, C. G., Busato, P., & Bothtis, D. D. (2014). Agricultural operations planning in fields with multiple obstacle areas. *Computers and Electronics in Agriculture*, *109*, 12–22. https://doi.org/10.1016/j.compag.2014.08.013.

Appendix B Conference Paper (ACIS 2019)

This part contains details of the author's conference article in Australasian Conference in Information System 2019, Perth, Australia. The full article can be found in this link: https://acis2019.io/pdfs/ACIS2019_PaperFIN_159.pdf

B.1 Attribution Statement

Title: Decision making for Farmers: A case study of Agricultural Routing Planning

	Conception and Design	Identification and Interpretation	Building the Algorithm	Experi ments	Analyse Result	Conclu sion	Final Approval
Co-Author 1 (Amalia Utamima)	✔	✔	✔	✔	✔	✔	✔
Co Author 1 Acknowledgment: I acknowledge that these represent my contribution to the above research output Signed: [redacted]							
Co-Author 2 (Torsten Reiners)	✔					✔	✔
Co Author 2 Acknowledgment: I acknowledge that these represent my contribution to the above research output Signed: [redacted]							
Co-Author 3 (Amir H. Ansaripoor)						✔	✔
Co Author 3 Acknowledgment: I acknow[redacted]ese represent my contribution to the above research output Signed: [redacted]							

B.2 Copyright Permission

B.3 Capture of Article

Decision making for Farmers: A Case Study of Agricultural Routing Planning

Full Paper

Amalia Utamima
Information Systems, School of Management
Curtin University
Perth, Australia
Email: Amalia.Utamima@postgrad.curtin.edu.au

Torsten Reiners
Information Systems, School of Management
Curtin University
Perth, Australia
Email: T.Reiners@curtin.edu.au

Amir H. Ansaripoor
Information Systems, School of Management
Curtin University
Perth, Australia
Email: Amir.Ansaripoor@curtin.edu.au

Abstract

Agricultural business is shifting to a stronger integration of information technology and data analysis to optimise the management and operations of small- and large-scale farms. In particular, computer support for decision-making is critical for farmers who want to decrease the cost of operations and control their (semi-)automated fleet of agricultural machines. This paper develops an optimisation module for decision support in Agricultural Routing Planning (ARP). The output is expected to help farmers to decide on the most efficient route for their harvesting machines. Specifically, the aim of this study is to contribute to optimisation solutions by introducing a new methodology called a Lovebird Algorithm, to address the routing problem. The Lovebird Algorithm acts as an optimisation tool to screen alternatives and focus only on efficient ones. The experimental results show that the proposed algorithm can save 8% of the non-working distance compared to the Genetic Algorithm and Tabu Search.

Keywords decision making, agriculture, routing planning, Lovebird Algorithm

1 Introduction

The main objective of Decision Support Systems (DSS) is to help, improve, and potentially automate the decision-making process (Turban et al. 2005). A decision-making process based on an optimisation technique is related to the recognition and solution of optimisation problems. Computer programs that solve optimisation problems are an essential element of several DSS (Bernus and Holsapple 2008).

Agricultural Routing Planning (ARP) is intended to optimise the design of machines' movements for agricultural field operations inside the farmer's field. The optimised design can minimise the length of routes travelled by machines, thereby saving costs and time associated with agricultural field operations (Utamima et al. 2018, 2019a). Utamima et al. (2019b) formalise the published ARP case with a mathematical model and optimise the published dataset of ARP. Seyyedhasani and Dvorak (2018) implemented multiple machines and minimised the total travel duration of every machine. Backman et al. (2015) used fluid turning in a manoeuvre, while Bochtis and Vougioukas (2008) minimised the non-working distance of machines used in a field.

To date, no DSS studies have addressed ARP. Recent studies on agricultural DSS focus on a different application. A DSS based on the optimisation model in fish farming is used to maximise the operators' profits (Cobo et al. 2019). The cultivation process of DSS is simulated through a bioeconomic model to obtain the optimal solution under certain conditions. Hafezalkotob et al. (2018) used a DSS to select the best olive harvesting machine among several alternatives. The output is expected to develop and improve the economic conditions in the agricultural field to meet food demand. A DSS based on the prediction model is proposed for the improvement of irrigation in agriculture (Giusti and Marsili-Libelli 2015; Navarro-Hellín et al. 2016).

The focus of this study is on building an optimisation module as an element of DSS. The ARP optimisation concentrates on the planning of routes for machines inside several agricultural fields for harvesting operations. In this research, each agricultural field has several established tracks with symmetrically-planted crops. These tracks can be traversed by both agricultural machines and harvesters. The decision-maker needs to determine which sequence of tracks will cover the shortest distance. ARP belongs to the class of NP-complete problems that makes an exact optimisation impossible as it is too time-consuming and complex to be applied (Marinakis et al. 2017). Therefore, this research develops a variation of an evolutionary algorithm called the Lovebird Algorithm.

This research contributes the development of a new algorithm (Lovebird Algorithm), and its application is represented in an optimisation module of DSS in ARP. The rest of this paper is organised as follows. Section 2 presents a literature review of current studies in DSS and ARP. Section 3 formalises the decisional problem with a mathematical formula of ARP and describes the proposed method. Section 4 presents the experimental results and analysis, while Section 5 suggests avenues for future research and concludes the paper.

2 Literature Review

Recent studies have proposed several decision support systems for agriculture. A DSS in fish farming is intended to optimise production strategies. The DSS contains an optimisation module that uses Particle Swarm Optimisation to optimise seabream aquaculture production (Cobo et al. 2019). A fuzzy-based DSS is proposed to improve irrigation in agriculture by deciding whether irrigation is needed and determining the amount required according to a set of rules involving variations of several weather variables (Giusti and Marsili-Libelli 2015). Navarro-Hellín et al. (2016) improved the DSS in irrigation by considering the soil measurement to precisely predict the irrigation needs. An agro-climate decision support tool is proposed to helps users to run crop simulation models for the targeted crops (Han et al. 2019).

The ARP problem involves minimising the distance travelled by machines when performing field operations inside an agricultural field (Utamima et al. 2019b). This problem has been altered and extended regarding the targets [e.g., improvement of time (Seyyedhasani and Dvorak 2018), minimisation of the headland distance (Backman et al. 2015)], specific field operations [e.g., herbicide application (Conesa-Muñoz, Bengochea-Guevara, et al. 2016), potato cultivation (Zhou et al., 2015), or orchard operation (Bochtis et al., 2015)] and limitations [e.g., restricted machine limit (Bakhtiari, et al., 2013), and obstacles (Zhou, et al., 2014)].

Previous studies on ARP focused mostly on real-case problems and solved these by means of several established algorithms. GA has been adapted for machine routing to decrease the total distance travelled in biomass transportation (Gracia et al. 2014). Sethanan and Neungmatcha (2016) used Particle Swarm

Optimisation (PSO) for route planning in sugarcane field operations, while Valente et al. (2013) employed Harmony Search to optimise coverage path planning in vineyard parcels. A hybrid Simulated Annealing was used for route planning of autonomous vehicles in herbicide application (Conesa-Muñoz, Bengochea-Guevara, et al. 2016).

Based on the previous research, two research gaps can be stated. First, despite the variations of DSS in agriculture, no formal studies apply the ARP in the context of farmers' decision-making. Therefore, this study is the first to consider ARP for such decision-making. Second, most studies use the currently-established algorithms rather than improving an algorithm for better results. Hence, the need to develop a better algorithm to improve the quality of ARP solutions.

3 Materials and Method

3.1 Problem Formulation

In ARP, a field has several established tracks with symmetrically-planted crops. These tracks can be traversed by both agricultural machines and harvesters. Each field has a headland area which is the crop-free area where machines perform manoeuvres to go to the next track. In the problem of interest, the machines need to start and end at the Depot. The farmer needs to determine which sequence of tracks in all fields will cover the shortest distance.

The tracks in ARP represent nodes in a graph, which must be visited by a machine. The arcs interfacing two nodes represent paths for the machines to move from one node to its neighbours. The machines can move to another track with a specific type of manoeuvre in the headland area of the field. Four manoeuvres are considered like shown in Figure 1: flat(Π), bulb(Ω), Flatθ ($\Pi\theta$), and Bulbθ ($\Omega\theta$). Note that $0 < \theta \leq 90$. Fig. 1(a-d) show an illustration of the four manoeuvres. If $\omega \geq r > \omega/2$ (ω = width of the track, r = turning radius of machines), the flat turn can occur only when the machine skips one or more tracks; otherwise, the bulb turn will be performed (Bochtis and Vougioukas 2008). A similar condition is also applied to Flatθ and Bulbθ with $\theta < 90$.

Suppose graph G contains a set of nodes N ($i,j \in N$) representing tracks in the fields. The set of homogeneous machines is represented as M ($m \in M$) and the set of tracks is T ($t \in T$). There are two decision variables: x_{ij}^m and x_i^m. The x_{ij}^m is equal to 1 if machine m moves from node i to node j; otherwise, it is equal to 0. The x_i^m is equal to 1 if machine m visits node i; otherwise, it is equal to 0. Equation (1) lists the objective function of the ARP in this study which is the minimisation of the non-working distance in the field. This distance is labelled 'non-working distance' since the machine is not performing an agricultural operation when making the turning manoeuvres.

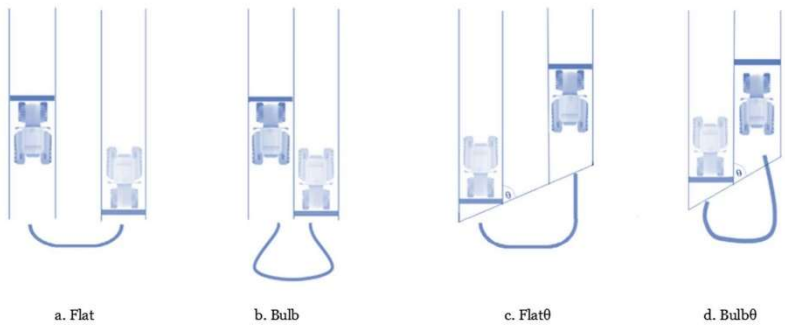


Figure 1: The four manoeuvres that are considered in this study (Utamima et al. 2019b)

The total working distance of every machine m is calculated with Eq. (2). The constraints of this model (Eq. 3-Eq 10) are adapted from the work of Utamima et al. (2019). The d_{ij} represents the types of manoeuvres or turns in the headland area that specifies in Eq. (3). Constraints (4)-(6) ensure that every node is visited only once by the machine. Constraint (7) guarantees that if a machine enters a node, it will also leave that same node. Constraint (8) excludes disjoint sub-tours (S) from a solution. Constraint

(9) restricts the maximum distance (B) for every machine. The last constraint (10) specifies that the decision variables are binary numbers.

$$z = \min\left(\sum_{i\in N}\sum_{j\in N}\sum_{m\in M} d_{ij}.x_{ij}^{m}\right) \tag{1}$$

$$Working\_Distance_m = \sum_{i\epsilon N}\sum_{m\in M} x_i^m\, l_t \tag{2}$$

s.t

$$d_{ij} = \begin{cases} \Pi(i,j) = w|i-j| + (\pi-2)\mathrm{r}, & if|i-j| \le \frac{2r}{\omega} \wedge \theta = 90 \\ \Omega(i,j) = \mathrm{r}\left(3\pi - 4\sin^{-1}\left(\frac{2r+w|i-j|}{4r}\right)\right), & if|i-j| > \frac{2r}{\omega} \wedge \theta = 90 \\ \Pi\theta(i,j) = w|i-j|(1+\cot\theta) + r(\pi-2), & if|i-j| \le \frac{2r}{\omega} \wedge \theta < 90 \\ \Omega\theta(i,j) = \pi r + \frac{4r^2 - w|i-j|(4r + \mathrm{w\,c}\quad{}^2\theta + \mathrm{w})}{4r - 2w|i-j|} \times \sin^{-1}\frac{w|i-j|(4r\cot\theta - 2\mathrm{w}\cot\quad)}{4r^2 - \mathrm{w}|i-j|(4r + \mathrm{w\,c}\quad{}^2\theta + \mathrm{w})}, & if|i-j| > \frac{2r}{\omega} \wedge \theta < 90 \end{cases} \tag{3}$$

$$\sum_{m\in M}\sum_{i\in N} x_{ij}^m = 1, \quad i,j \ne 0,\ j \in N: i \ne j \tag{4}$$

$$\sum_{m\in M}\sum_{i\in N} x_i^m = 1, \quad i \ne 0 \tag{5}$$

$$\sum_{m\in M}\sum_{j\in N} x_{ij}^m = 1, \quad i,j \ne 0,\ i \in N: i \ne j \tag{6}$$

$$\sum_{m\in M}\sum_{i\in N} x_{ij}^m = \sum_{m\in M}\sum_{j\in N} x_{ji}^m, \quad i,j \in N \tag{7}$$

$$\sum_{i\epsilon S} x_i^m \le \|S\| - 1, \forall S \subseteq N, \|S\| \ge 1, m \in M \tag{8}$$

$$\sum_{i\in N} l_i x_i^m < B, m \in M,\ i \in N \tag{9}$$

$$x_{ij}^m, x_i^m \epsilon \{0,1\} \tag{10}$$

3.2 Lovebird Algorithm

The representation of a candidate solution uses a permutation number as shown in Figure 2. Each track is allocated a number, and it has the sequence that will be visited by a machine. For instance, in Figure 2, the machine will visit track no. 1.

Sequence:	1	2	3	4	5	6	7	8
Track:	1	4	7	3	6	2	5	8

Figure 2: A candidate solution representation

This research proposes a new algorithm called the Lovebird Algorithm to solve the ARP. The Lovebird Algorithm adapts combinatorics operators to produce the offspring (new candidate solution). Figure 3 shows the flowchart of the Lovebird Algorithm. In the beginning, the fields' details (containing every track, entrance, and Depot coordinates) and machine information (the number of available machines and the capacity) become the input of the algorithm. The initialisation phase of Lovebird Algorithms sets parameters (maximum of iterations and the size of the population) and the variables (every tracks distance to the entrances and Depot). The stopping criteria is the *max_iter,* which is the maximum number of iterations in the algorithm. The *max_iter* is set to *50n* (n = number of fields). The main iterations start with the calculation of the objective function based on the mathematical model that is shown in Section 3.1. Then, the Lovebird's offspring production executes one of the five choices of combinatorics operators:

a) Red: Swap the sections (Figure 4 (a))
b) Peach: Flip the sequence (Figure 4 (b))
c) Green: Interchange two tracks (Figure 4 (c))
d) Yellow: Move and push (Figure 4 (d))
e) Grey: Mix the tracks (Figure 4 (e))

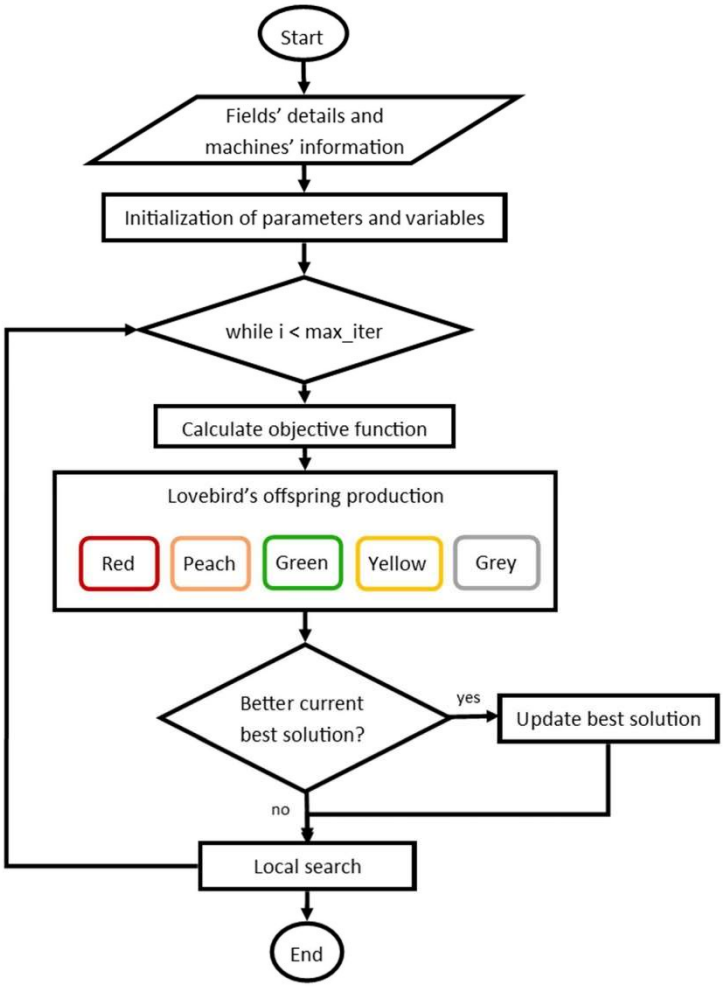


Figure 3: Flowchart for Lovebird Algorithm

Figure 4 illustrates the combinatorics operators that are used in the Lovebird Algorithm as listed previously in a-e. The offspring is the new candidate solution after the combinatorics operator has been applied, while the parent is the previous candidate solutions. The swap section (Figure 4 (a)) swaps the red section of two parents. Offspring 1 keeps the red section from Parent 2 and copies the rest of the tracks from Parent 1, while Offspring 2 does the opposite. The flip operator (Figure 4 (b)) flips over the tracks' position in the peach colour section, while the interchange operator (Figure 4 (c)) changes the position of the green section. Figure 4 (d)) shows the move and push operators that move the location of a front yellow point to a back yellow point and push forward the remaining tracks. The last operator is the grey operator that mixes the sequences of the tracks in a candidate solution.

The combinatorics operators are used as the exploration stage in a metaheuristic algorithm (Soni and Kumar 2014). The next phase involves the updating of the new candidate solutions and their objective

values. The best solution among the iterations is updated if a better solution is found in the current iteration. Next, a local search scans the neighbourhood of the best solution found so far to determine whether further improvement is possible.

Parent 1	1	2	3	4	5	6	7	8
Parent 2	2	1	4	5	6	7	8	3
Offspring 1	1	2	4	5	6	7	3	8
Offspring 2	2	1	3	4	5	6	7	8

(a) Red (Swap section)

Parent	2	1	4	5	6	7	8	3
Offspring	2	1	6	5	4	7	8	3

(b) Peach (Flip)

Parent	2	1	4	5	6	7	8	3
Offspring	2	8	6	5	4	7	1	3

(c) Green (Interchange)

Parent	2	1	4	5	6	7	8	3
Offspring	2	4	5	6	7	8	3	1

(d) Yellow (Move and push)

Parent	1	2	3	4	5	6	7	8
Offspring	2	8	6	5	4	7	1	3

(e) Grey (Mix)

Figure 4: Depiction of combinatorics operators in Lovebird

3.3 The Optimisation Module in DSS

Figure 5 presents the proposed framework of an optimisation module in DSS for ARP. This framework is in line with what is stated in Cobo et al. (2019) and Ben Jouida and Krichen (2018). The input of the module consists of the coordinates of every track, entrances to each field, and the Depot. The machines' capacity and the number of machines also become input. The process starts with the calculation of track distances inside the fields and to the entrances and the Depot. Then, the Lovebird Algorithm is executed, as explained in Section 3.2. The outputs of the module are the optimised order of tracks that need to be traversed by the machines, the non-working distance, and the length of harvested tracks (working distance).

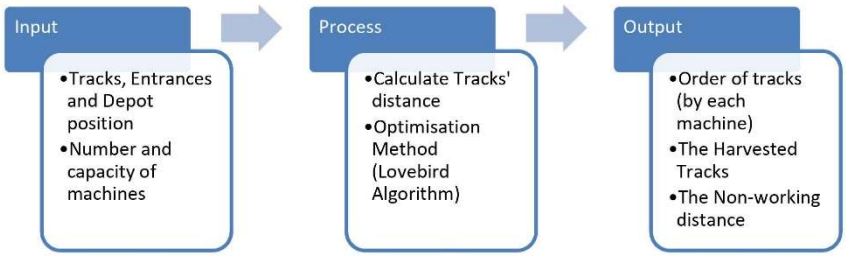


Figure 5: Framework of the optimisation module in DSS for ARP

We can integrate the proposed module in a fleet management system in agriculture (Sørensen and Bochtis 2010). This system includes an online decision support system, and the online routing will assist the farmer in running the machines. Another alternative is to add the Lovebird algorithm to a comprehensive farm management system (Sørensen et al. 2010). The new design of a farm management system considers new situations from the perspectives of both farmers and managers. Specifically, the routing algorithm can be included in one of the system modules called 'plan generation'. After that, the farmers can execute the route provided by the management system.

Based on Sørensen et al. (2010), we can derive the CATWOE (Customers, Actors, Transformation Process, World-view, and Ownership) elements of a DSS in ARP as listed below:

a. Customers: the primary customer of the DSS is the farm manager.

b. Actors: operates the DSS, in this case, is the farm manager or other farm staff.

c. Transformation process: related to the transformation of operational field data into manageable information for decision making.

d. World-view: the operational data is easily acquired and can be used to improve decision making.

e. Ownership: the farm manager as responsible to the everyday decision-maker, and decides whether the framework is of use.

f. Environmental constraints: includes the reliability and structure of information technology.

4 Experimental Results and Analysis

The experiments record the output of our optimisation module of DSS. At first, the Lovebird Algorithm is applied to the ARP dataset derived from previous research. Then, the Lovebird Algorithm is applied to solve the harvesting problem. Besides the Lovebird Algorithm, this research also applies GA and Tabu Search (TS) to compare the results.

The first column in Table 1 listed the dataset of ARP (based on the real field) that are taken from Bochtis and Vougioukas (2008) and Conesa-Muñoz et al. (2016). As shown in Table 1 columns 2-4, the Lovebird Algorithm can achieve the smallest non-working distance compared to those of GA and TS.

Problem Code	Non-working distance (meters)		
	Lovebird Algorithm	Genetic Algorithm	Tabu Search
A12	**146.027**	150.602	146.027
B12	**145.602**	160.602	146.027
C20	**235.491**	250.915	240.915

Table 1. The non-working distance of ARP dataset from previous research

For the harvesting problem, this research uses two kinds of fields as instances of ARP. The layout of the fields is shown in Figure 6. Each field has an entrance point, and the machines can enter the field only at that point. Also, every machine needs to start and end at the Depot. Every track is labelled with a number. The small number near the blue tracks in Figure 6 refers to the track number. For instance, in Figure 6, the first field has 18 tracks starting from the left to right, and the second field has 22 tracks (track no. 19-40). We use three machines with the same capacity (homogeneous machines). Another assumption is that each machine can harvest a maximum of 3000 meters of track.

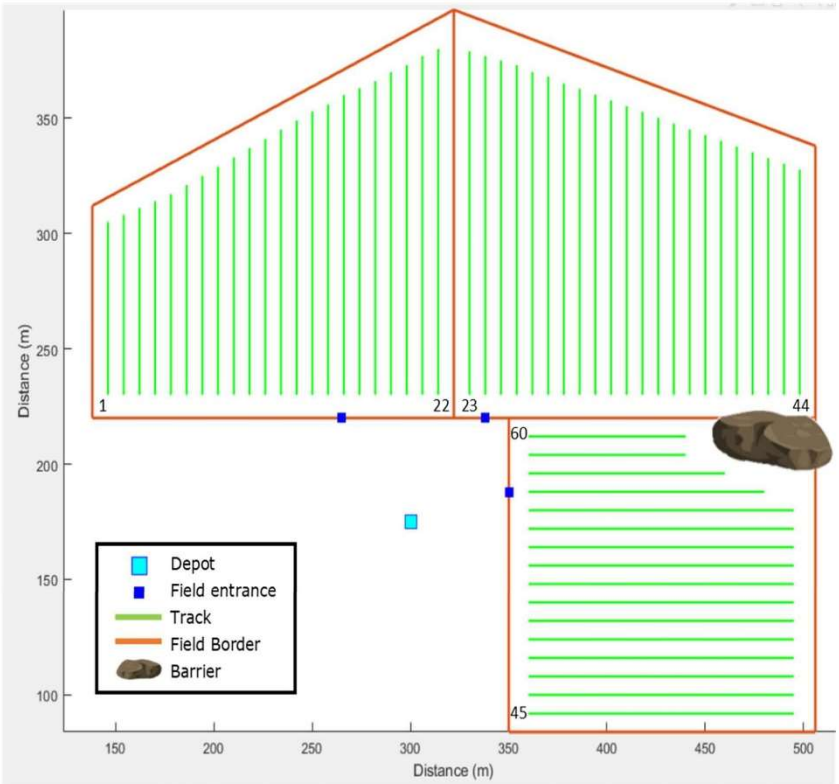


Figure 6: Diagram of three fields with 60 tracks.

Table 2 presents the results achieved with the Lovebird Algorithm compared to GA and TS. The first column in Table 2 refers to the number of fields and the total tracks in that field while the second, third, and fourth columns show the non-working distance of Lovebird Algorithm, GA, TS. The last column lists the distance reductions achieved by the Lovebird Algorithm compared to GA and TS. As shown in Table 2, the Lovebird Algorithm's solution always has the smallest non-working distance compared to those of other algorithms. The Lovebird Algorithm performs better than other algorithms because it applies several combinatorial operators that produce better solutions in ARP (Conesa-Muñoz, Pajares, et al. 2016). The Lovebird Algorithm successfully achieves an average of 8% distance reduction compared to that of GA and TS.

#Fields (#total tracks)	Non-working distance (meters)			Distance Reduction
	Lovebird Algorithm	Genetic Algorithm	Tabu Search	
2 (40)	**614.625**	673.634	634.718	6%
3 (60)	**1106.034**	1246.720	1189.704	10%

Table 2. Non-working distance comparison of Lovebird Algorithm, GA, and TS

Table 3 listed the optimised order of tracks of the Lovebird Algorithm. The first column refers to the fields and the machines used. The second column refers to the order of tracks, while the last column lists the length of the harvested tracks. The problem with two fields (Table 3 row 2-4) needs two machines to harvest the fields while the problem with three fields (Table 3 row 5-8) needs three machines. For example, in the problem with two fields, Machine 1 will go to Field 1 and harvest the tracks in the order 13, 15, 17, 18, 16, 14, 12, 11, 9, 8, 6, 5, 2, 1, 3, 4, 7, 10 and then it will go to Field 2 and harvest track no. 38, 36, 39, and 40. The rest of the tracks in Field 2 (track no. 37, 35, 33, 34, 32, 31, 29, 30, 28, 27, 25, 26, 24, 23, 20, 19, 21, 22) will be harvested by Machine 2.

Table 4 shows the comparison of the running time of the Lovebird Algorithm and other algorithms. The first column listed the problem, while the second column refers to the running time of the algorithms in seconds. The Lovebird Algorithm is able to get the faster running time in all problems compared to GA and TS.

Fields & Machine	Optimised Tracks-Order	Harvested Tracks (meters)
2 fields:		
Machine 1	Field 1 [13, 15, 17, 18, 16, 14, 12, 11, 9, 8, 6, 5, 2, 1, 3, 4, 7, 10]; Field 2 [38, 36, 39, 40]	2745
Machine 2	Field 2 [37, 35, 33, 34, 32, 31, 29, 30, 28, 27, 25, 26, 24, 23, 20, 19, 21, 22]	2159
3 fields:		
Machine 1	Field 1 [16, 14, 12, 11, 8, 7, 5, 3, 1, 2, 4, 6, 9, 10, 13, 15, 19, 20, 22, 21, 18, 17] Field 2 [34, 33]	2729
Machine 2	Field 2 [40, 42, 44, 43, 41, 39, 36, 35, 33, 34, 32, 31, 29, 30, 28, 27, 25, 26, 24, 23] Field 3 [58, 60, 59, 57, 54, 56]	2782
Machine 3	Field 3 [55, 52, 50, 48, 46, 45, 47, 49, 53, 51]	1350

Table 3. The optimised tracks'-order and the harvested tracks of Lovebird Algorithm

#Fields (#total tracks)	Running Time (Seconds)		
	Lovebird Algorithm	Genetic Algorithm	Tabu Search
2 (40)	**3.786**	4.850	4.859
3 (60)	**5.712**	6.819	7.693

Table 4. The running time comparison of Lovebird Algorithm, GA, and TS

5 Conclusion and Future Works

In regard to an agricultural field, the decision-making that is supported by a useful DSS can improve the quality of the decision. This study presents an optimisation module of a DSS in ARP that aims to decrease costs and to maintain sustainability. The Lovebird Algorithm is proposed as the optimisation method in the module to indicate the routes of machines in respect to the shortest non-working distance.

The comparison of the proposed algorithm with the Genetic Algorithm and Tabu Search shows that the Lovebird Algorithm successfully saves 8% travel distance and achieves the fastest running time in all problem instances. This study is limited to the development of the optimisation module of DSS in ARP.

Future research can focus on building the whole DSS to support decision-making in ARP. The various applications of the DSS in ARP, which include the optimised routing of the machines, can also be considered as a future direction, such as for herbicide applications, orchard operation, or fertilising operation. Another future research can focus on combining multiple systems, on improving the organisation of the field, which includes the DSS and several information systems related to fieldwork and the harvested crop management. Information about minimised routes for the machines is essential for both current and future agriculture field management. In the future, minimised routes can become the input for autonomous vehicles (without farmer onboard) that are used to harvest the fields.

6 References

Backman, J., Piirainen, P., and Oksanen, T. 2015. "Smooth Turning Path Generation for Agricultural Vehicles in Headlands", *Biosystems Engineering* (139), Academic Press, pp. 76–86. (https://doi.org/10.1016/J.BIOSYSTEMSENG.2015.08.005).

Bakhtiari, A., Navid, H., Mehri, J., Berruto, R., and Bochtis, D. D. 2013. "Operations Planning for Agricultural Harvesters Using Ant Colony Optimization", *Spanish Journal of Agricultural Research* (11:3), pp. 652–660. (https://doi.org/10.5424/sjar/2013113-3865).

Bochtis, D. D., and Vougioukas, S. G. 2008. "Minimising the Non-Working Distance Travelled by Machines Operating in a Headland Field Pattern", *Biosystems Engineering* (101:1), pp. 1–12. (https://doi.org/10.1016/j.biosystemseng.2008.06.008).

Bochtis, D., Griepentrog, H. W., Vougioukas, S., Busato, P., Berruto, R., and Zhou, K. 2015. "Route Planning for Orchard Operations", *Computers and Electronics in Agriculture* (113), Elsevier B.V., pp. 51–60. (https://doi.org/10.1016/j.compag.2014.12.024).

Cobo, Á., Llorente, I., Luna, L., and Luna, M. 2019. "A Decision Support System for Fish Farming Using Particle Swarm Optimization", *Computers and Electronics in Agriculture* (161:February 2018), Elsevier, pp. 121–130. (https://doi.org/10.1016/j.compag.2018.03.036).

Conesa-Muñoz, J., Bengochea-Guevara, J. M., Andujar, D., and Ribeiro, A. 2016. "Route Planning for Agricultural Tasks: A General Approach for Fleets of Autonomous Vehicles in Site-Specific Herbicide Applications", *Computers and Electronics in Agriculture* (127), pp. 204–220. (https://doi.org/10.1016/j.compag.2016.06.012).

Conesa-Muñoz, J., Pajares, G., and Ribeiro, A. 2016. "Mix-Opt: A New Route Operator for Optimal Coverage Path Planning for a Fleet in an Agricultural Environment", *Expert Systems with Applications* (54), pp. 364–378. (https://doi.org/10.1016/j.eswa.2015.12.047).

Florentino, H. de O., Irawan, C., Aliano, A. F., Jones, D. F., Cantane, D. R., and Nervis, J. J. 2018. "A Multiple Objective Methodology for Sugarcane Harvest Management with Varying Maturation Periods", *Annals of Operations Research* (267:1–2), Springer US, pp. 153–177. (https://doi.org/10.1007/s10479-017-2568-2).

Giusti, E., and Marsili-Libelli, S. 2015. "A Fuzzy Decision Support System for Irrigation and Water Conservation in Agriculture", *Environmental Modelling & Software* (63), Elsevier, pp. 73–86. (https://doi.org/10.1016/J.ENVSOFT.2014.09.020).

Gracia, C., Velázquez-Martí, B., and Estornell, J. 2014. "An Application of the Vehicle Routing Problem to Biomass Transportation", *Biosystems Engineering* (124), Academic Press, pp. 40–52. (https://doi.org/10.1016/J.BIOSYSTEMSENG.2014.06.009).

Han, E., Baethgen, W. E., Ines, A. V. M., Mer, F., Souza, J. S., Berterretche, M., Atunez, G., and Barreira,

C. 2019. "SIMAGRI: An Agro-Climate Decision Support Tool", *Computers and Electronics in Agriculture* (161), Elsevier, pp. 241–251. (https://doi.org/10.1016/J.COMPAG.2018.06.034).

Ben Jouida, S., and Krichen, S. 2018. "A DSS Based on Optimizer Tools and MTS Meta-Heuristic for the Warehousing Problem with Conflicts", *Information Processing Letters* (135), Elsevier B.V., pp. 14–21. (https://doi.org/10.1016/j.ipl.2018.02.001).

Marinakis, Y., Migdalas, A., and Sifaleras, A. 2017. "A Hybrid Particle Swarm Optimization – Variable Neighborhood Search Algorithm for Constrained Shortest Path Problems", *European Journal of Operational Research* (261:3), Elsevier B.V., pp. 819–834. (https://doi.org/10.1016/j.ejor.2017.03.031).

Seyyedhasani, H., and Dvorak, J. S. 2018. "Reducing Field Work Time Using Fleet Routing Optimization", *Biosystems Engineering* (169), Elsevier Ltd, pp. 1–10. (https://doi.org/10.1016/j.biosystemseng.2018.01.006).

Soni, N., and Kumar, T. 2014. "Study of Various Mutation Operators in Genetic Algorithms", *International Journal of Computer Science and Information Technologies* (5:3), pp. 7235–7238.

Sørensen, C. G., and Bochtis, D. D. 2010. "Conceptual Model of Fleet Management in Agriculture", *Biosystems Engineering* (105:1), Academic Press, pp. 41–50. (https://doi.org/10.1016/J.BIOSYSTEMSENG.2009.09.009).

Sørensen, C. G., Fountas, S., Nash, E., Pesonen, L., Bochtis, D., Pedersen, S. M., Basso, B., and Blackmore, S. B. 2010. "Conceptual Model of a Future Farm Management Information System", *Computers and Electronics in Agriculture* (72:1), pp. 37–47. (https://doi.org/10.1016/j.compag.2010.02.003).

Utamima, A., Reiners, T., and Ansaripoor, A. H. 2019a. "Evolutionary Estimation of Distribution Algorithm for Agricultural Routing Planning in Field Logistics", *Procedia Computer Science* (Forthcoming).

Utamima, A., Reiners, T., and Ansaripoor, A. H. 2019b. "Optimisation of Agricultural Routing Planning in Field Logistics with Evolutionary Hybrid Neighbourhood Search", *Biosystems Engineering* (184), pp. 166–180. (https://doi.org/10.1016/j.biosystemseng.2019.06.001).

Utamima, A., Reiners, T., Ansaripoor, A., and Seyyedhasani, H. 2018. "The Agricultural Routing Planning in Field Logistics", in *Contemporary Approaches and Strategies for Applied Logistics*, IGI Global, pp. 261–283. (https://doi.org/10.4018/978-1-5225-5273-4.ch010).

Zhou, K., Jensen, A. L., Bochtis, D. D., and Sørensen, C. G. 2015. "Quantifying the Benefits of Alternative Fieldwork Patterns in a Potato Cultivation System", *Computers and Electronics in Agriculture* (119), Elsevier B.V., pp. 228–240. (https://doi.org/10.1016/j.compag.2015.10.012).

Zhou, K., Leck Jensen, A., Sørensen, C. G., Busato, P., and Bothtis, D. D. 2014. "Agricultural Operations Planning in Fields with Multiple Obstacle Areas", *Computers and Electronics in Agriculture* (109), Elsevier B.V., pp. 12–22. (https://doi.org/10.1016/j.compag.2014.08.013).

Appendix C Conference Paper (ICCAE 2020)

This part contains details of the author's conference article in Association for Computing Machinery (ACM). The article was presented in 2020 12th International Conference on Computer and Automation Engineering, Sydney, Australia. The full article can be found in this link:

https://dl.acm.org/doi/abs/10.1145/3384613.3384621.

C.1 Attribution Statement

Title : Automation in Agriculture: A Case Study of Route Planning Using an Evolutionary Lovebird Algorithm

	Conception and Design	Identify Math Model	Build the Algorithms	Experi ments	Result Analysis	Conclu sion	Final Approval
Co-Author 1 (Amalia Utamima)	✔	✔	✔	✔	✔	✔	✔
Co Author 1 Acknowledgment: I acknowledge that these represent my contribution to the above research output Signed: [redacted]							
Co-Author 2 (Torsten Reiners)	✔					✔	✔
Co Author 2 Acknowledgment: I acknowledge that these represent my contribution to the above research output Signed: [redacted]							
Co-Author 3 (Amir H. Ansaripoor)						✔	✔
Co Author 3 Acknowledgment: I acknowledge that these represent my contribution to the above research output Signed: [redacted]							

C.2 Copyright Permission

Association for Computing Machinery

Digital Library CACM Blue Diamond

Publish with ACM, the world's largest computing society

HOME PROCEEDINGS JOURNALS MAGAZINES BOOKS NEWSLETTERS AUTHOR RESOURCES OPEN ACCESS CO

Home > ACM Author Rights

ACM Author Rights

ACM exists to support the needs of the computing community. For over sixty years ACM has developed publications and publication policies to maximize the visibility, impact, and reach of the research it publishes to a global community of researchers, educators, students, and practitioners. ACM has achieved its high impact, high quality, widely-read portfolio of publications with:

- Affordably priced publications
- Liberal Author rights policies
- Wide-spread, perpetual access to ACM publications via a leading-edge technology platform
- Sustainability of the good work of ACM that benefits the profession

Choose

Authors have the option to choose the level of rights management they prefer. ACM offers three different options for authors to manage the publication rights to their work.

- Authors who want ACM to manage the rights and permissions associated with their work, which includes defending against improper use by third parties, can use ACM's traditional copyright transfer agreement.
- Authors who prefer to retain copyright of their work can sign an exclusive licensing agreement, which gives ACM the right but not the obligation to defend the work against improper use by third parties.
- Authors who wish to retain all rights to their work can choose ACM's author-pays option, which allows for perpetual open access through the ACM Digital Library. Authors choosing the author-pays option can give ACM non-exclusive permission to publish, sign ACM's exclusive licensing agreement or sign ACM's traditional copyright transfer agreement. Those choosing to grant ACM a non-exclusive permission to publish may also choose to display a Creative Commons License on their works.

Post

Otherwise known as "Self-Archiving" or "Posting Rights", all ACM published authors of magazine articles, journal articles, and conference papers retain the right to post the pre-submitted (also known as "pre-prints"), submitted, accepted, and peer-reviewed versions of their work in any and all of the following sites:

- Author's Homepage
- Author's Institutional Repository
- Any Repository legally mandated by the agency or funder funding the research on which the work is based
- Any Non-Commercial Repository or Aggregation that does not duplicate ACM tables of contents. Non-Commercial Repositories are defined as Repositories owned by non-profit organizations that do not charge a fee to access deposited articles and that do not sell advertising or otherwise profit from serving scholarly articles.

C.3 Capture of Article

Automation in Agriculture: A Case Study of Route Planning Using an Evolutionary Lovebird Algorithm

Amalia Utamima[1,2], Torsten Reiners[1], Amir H. Ansaripoor[1]
[1] Information Systems, Curtin University, Perth, Australia
[2] Institut Teknologi Sepuluh Nopember, Indonesia

ABSTRACT

A recent trend in the agricultural sector is the integration of computers to support automation in the operation of small and large-scale farms. The utilization of computers for decision making is critical for farmers wanting to lower their operative costs and control their machines. The focus of this paper is on the optimization of route planning for agricultural machines that are applying fertilizer on fields. The output of this research is expected to support automation in agriculture by helping farmers to choose the most efficient route for their machines.

This study formalizes the decisional problem with a mathematical formula and presents a new improved algorithm, Evolutionary Lovebird Algorithm, to solve the problem. The experimental results show that the proposed algorithm can save 8.45% of the non-working distance compared to other algorithms. Moreover, on average, the running time of the proposed algorithm is only one-third of other algorithms, thereby making the Evolutionary Lovebird Algorithm three times more efficient than other algorithms.

CCS Concepts

• **Applied computing→Operations research.**

Keywords

Agriculture; route planning; Evolutionary Lovebird Algorithm

1. INTRODUCTION

Agricultural Routing Planning (ARP) has the objective of optimizing the design of the movements of agricultural machines operating inside the farmer's field. The optimized design can minimize the routing distance of the machines, thereby saving cost and time in terms of agricultural field operations [1]. Utamima et al. in [2] gathered the published ARP cases and used this dataset for optimization purposes. Seyyedhasani and Dvorak in [3] reduced the fieldwork time by minimizing the total travel duration of every machine. Bochtis et al. [4] generated route plans for orchard operations, while Backman et al. [5] generated smooth turning paths for agricultural vehicles in the headland areas of the fields.

Currently, ARP studies dealing with the spraying of chemicals or fertilizers on fields have not been comprehensive and tend to focus only on a single field. Conesa-Muñoz et al. [6] solved route planning in site-specific herbicide applications in one field while Jensen et al. [7] optimized the field operations of the application of liquid fertilizer on a single field.

The focus of this study is on the route planning for machines applying fertilizer inside several agricultural fields. In this research, each agricultural field has several established tracks with crops planted symmetrically. These tracks can be traversed by agricultural machines. The decision-maker needs to determine the sequence of tracks that will cover the shortest distance. ARP belongs to the class of NP-hard problems; hence, it is impossible to apply an exact optimization as it is too time-consuming and complex [8]. Therefore, this research develops an Evolutionary Lovebird algorithm.

This research makes two contributions: it establishes a mathematical ARP model for fertilizer application in multiple fields and develops an improved algorithm named the Evolutionary Lovebird Algorithm. The rest of this paper is organized as follows. Section 2 presents the literature review of current studies in ARP. Section 3 gives the mathematical formula for ARP and explains the proposed algorithm. Section 4 presents the experimental results and analysis. Section 5 suggests future research directions and concludes the paper.

2. LITERATURE REVIEW

The ARP problem involves minimizing the distance that machines need to travel when performing operations inside an agricultural field [9]. This problem has been altered and extended in terms of the objectives related to improvement of time [3], minimization of the non-working distance [10], specific field operations (e.g. herbicide application) [6], fertilizer application [7], or biomass transportation [11]) and limitations (e.g., restricted machine limit [12], and obstacles [13]).

Variations have been introduced by several authors [3,6,7,10–13]. Seyyedhasani and Dvorak [3] focused on minimizing the route of agricultural machines, thereby reducing the fieldwork time, while Bochtis and Vougioukas [10] minimized the non-working distance of machines used in a field. Meanwhile, Bakhtiari et al. [12] examined an extension of the problem whereby the harvester unloads at a stationary facility located outside the field area, while Conesa-Muñoz et al. [6] defined a new problem in herbicide application, adding variability in the field. Jensen et al. [7] developed an algorithmic approach for the optimization of capacitated field operations concerned with liquid fertilization. Zhou et al. [13] focused on a field containing multiple obstacles, while Gracia, Velázquez-Martí, and Estornell [11] applied the routing problem concept to biomass transportation.

ICCAE 2020, February 14–16, 2020, Sydney, NSW, Australia

ACM ISBN 978-1-4503-7678-5/20/02…$15.00
https://doi.org/10.1145/3384613.3384621

Appendix D Conference Paper (ISICO 2019)

This part contains details of the author's conference article in Procedia Computer Science 2019, Elsevier. The article was presented in The Fifth Information Systems International Conference, Indonesia, 2019. The full article can be found in this link: https://www.sciencedirect.com/science/article/pii/S1877050919318678.

D.1 Attribution Statement

Title : Evolutionary estimation of distribution algorithm for agricultural routing planning in field logistics

	Conception and Design	Identify Math Model	Build the Algorithm	Experi ments	Analyse Result	Conclu sion	Final Approval
Co-Author 1 (Amalia Utamima)	✔	✔	✔	✔	✔	✔	✔
Co Author 1 Acknowledgment: I acknowledge that these represent my contribution to the above research output Signed: [redacted]							
Co-Author 2 (Torsten Reiners)	✔					✔	✔
Co Author 2 Acknowledgment: I acknowledge that these represent my contribution to the above research output Signed: [redacted]							
Co-Author 3 (Amir H. Ansaripoor)		✔				✔	✔
Co Author 3 Acknowledgment: I acknowledge that these represent my contribution to the above research output Signed: [redacted]							

D.2 Copyright Permission

D.3 Capture of Article

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 161 (2019) 560–567

Procedia Computer Science

www.elsevier.com/locate/procedia

The Fifth Information Systems International Conference 2019

Evolutionary Estimation of Distribution Algorithm for Agricultural Routing Planning in Field Logistics

Amalia Utamima[a,b,*], Torsten Reiners[a], Amir H. Ansaripoor[a]

[a]*Information Systems, Curtin University, Perth 6102, Australia*
[b]*Institut Teknologi Sepuluh Nopember, Surabaya 60111, Indonesia*

Abstract

Agricultural Routing Planning (ARP), a problem in field logistics, has the objective to minimize the headland distance used by machines when performing agricultural tasks. This study gathers for its datasets the data for several fields obtained from previous research. The Estimation of Distribution Algorithm (EDA) is an algorithm that employs a probabilistic model to produce candidate solutions. This paper extends the EDA to become the Evolutionary EDA that combines a general EDA, a neighborhood search, and an elitism technique. Evolutionary EDA is tested on the optimization of ARP. The experimental results show that Evolutionary EDA can get the same or outperform the solutions generated by previously applied algorithms on ARP problems.

Keywords: Evolutionary algorithm; estimation distribution algorithm; agricultural routing planning; field logistics

1. Introduction

Agricultural routing planning in farm management is intended to design or schedule the movements of machines inside fields for agricultural tasks. A good design can minimize the distance of the machine's tours, thereby leading to cost savings. Hence, it is essential to have an optimized plan for the routing of the machines to complete agricultural field operations [1]. Figure 1 illustrates the layout of an agricultural field. The field has several established tracks with symmetrically planted crops. These tracks can be traversed by both agricultural machines and harvesters.

* Corresponding author. Tel.: +61-89-266-5141.
E-mail address: amalia.utamima@postgrad.curtin.edu.au

10.1016/j.procs.2019.11.156

Appendix E Book Chapter (IGI 2018)

This part contains details of the author's book chapter in IGI Global, 2018. The chapter I part of the book entitled "Contemporary Approaches and Strategies for Applied Logistics". The full article can be found in this link: https://www.igi-global.com/chapter/the-agricultural-routing-planning-in-field-logistics/196931.

E.1 Attribution Statement

Title : The Agricultural Routing Planning in Field Logistics

	Conception and Design	Collecting Data	Identification and Grouping	Interpretation	Discussion and Conclusion	Final Approval
Co-Author 1 (Amalia Utamima)	✔	✔	✔	✔	✔	
Co Author 1 Acknowledgment: I acknowledge that these represent my contribution to the above research output Signed: [redacted]						
Co-Au[redacted] (Torsten Reiners)	✔				✔	✔
Co Author 2 Acknowledgment: I acknowledge that these represent my contribution to the above research output Signed: [redacted]						
Co-Author 3 (Amir H. Ansaripoor)					✔	✔
Co Author 3 Acknowledgment: I acknowledge that these represent my contribution to the above research output Signed: [redacted]						
Co-Auth[redacted] (Hasan Seyyedhasani)		✔			✔	
Co Author 4 Acknowledgment: I acknowledge that these represent my contribution to the above research output Signed: [redacted]						

E.2 Copyright Permission

https://www.igi-global.com/about/rights-permissions/content-reuse/

IGI Global
DISSEMINATOR OF KNOWLEDGE

Shopping Cart Amalia Utamima

Search title, author, ISBN...

Books Journals InfoSci®-Databases Articles/Chapters Publish with Us Resources Catalogs About Us

IGI Global's Fair Use Policy

Although all authors and editors of IGI Global sign an Author Warranty and Transfer of Copyright Agreement, IGI Global supports a Fair Use Policy allowing authors and editors to:

IGI GLOBAL AUTHORS, UNDER FAIR USE CAN:

- Post the final typeset PDF (which includes the title page, table of contents and other front materials, and the copyright statement) of their chapter or article (NOT THE ENTIRE BOOK OR JOURNAL ISSUE), on the author or editor's secure personal website and/or their university repository site.

IGI GLOBAL AUTHORS, AS PER THE AUTHOR'S WARRANTY AND TRANSFER OF COPYRIGHT CANNOT:

- Cannot post the article, chapter, full book publication, or journal issue on general open access sites for free download and distribution, such as ResearchGate, arXiv, Academia.edu, SSRN, or society-sponsored sites.
- Cannot use in a coursepack without first securing the consent of IGI Global.
- Cannot copy or distribute for any monetary consideration.
- Cannot be given to a commercial third party to post, copy, distribute, sell or give for free or for any monetary consideration.

For those authors and editors wishing to republish their chapter or article in a future book or journal publication they are authoring or editing, they will need to seek written permission from IGI Global.

IGI Global abides by international copyright law, and as such all contributors are asked to sign an IGI Global Author Warranty and Transfer of Copyright agreement prior to publication. The warranty portion ensures that materials submitted to IGI Global for consideration are original and have not been published elsewhere previously.

The copyright portion clearly states: Author(s) cannot post the contents of the article/chapter on any website or distribute the work to others in either electronic or print forms without the written permission of IGI Global. This includes any article or chapter currently under consideration by IGI Global and the final published work.

To inquire about permissions and rights of IGI Global publications, please contact:

Jan Travers
IGI Global
Director of Intellectual Property and Contracts
701 E Chocolate Avenue
Hershey PA 17033-1240, USA
jtravers@igi-global.com
717.533-8845 x112 (tel); 717.533-8661 (fax)

E.3 Capture of Article

Chapter 10

The Agricultural Routing Planning in Field Logistics

Amalia Utamima
Curtin University, Australia

Torsten Reiners
Curtin University, Australia

Amir Ansaripoor
Curtin University, Australia

Hasan Seyyedhasani
University of Kentucky, USA

ABSTRACT

The agricultural sector is facing the need to gain a higher yield on their fields while optimising their operations to stay competitive and satisfy the continuously increasing demand for produce. Cost reductions can be achieved by increasing the effective field size and reducing the operations without gain (e.g., driving longer distance to harvest the field). The agricultural routing planning (ARP) problem represents a specialisation of the travelling salesman problem (TSP) or vehicle routing problem (VRP) with focus on the agricultural operations and considerations of the field and vehicles configurations. In addition, various adaptations of the problem can be found in the literature that define a new problem class with specialised optimisation needs. This chapter introduces the ARP and reviews the past and current research and developments.

DOI: 10.4018/978-1-5225-5273-4.ch010

TEMPORARY SECTION - DO NOT REMOVE (YET)!

This 1-page section is temporary but **important**!

Do NOT remove it until you've created a full skeleton of the thesis, i.e. until you have a section (with title line) set up for every chapter, appendix, front matter/back matter section in your thesis.

Only then should you delete this page **AND** the preceding section break.