GENETIC ALGORITHM FOR MULTI-PRODUCT MULTI-PERIOD AGGREGATE PRODUCTION PLANNING AND VEHICLE ROUTING PROBLEMS WITH TIME WINDOWS

By

MISS Ratchadakorn POOHOI

A Thesis Submitted in Partial Fulfillment of the Requirements

for Doctor of Philosophy ENGINEERING MANAGEMENT

Department of INDUSTRIAL ENGINEERING AND MANAGEMENT

Silpakorn University

Academic Year 2023

ขั้นตอนวิธีเชิงพันธุกรรมสำหรับการวางแผนการผลิตแบบหลายผลิตภัณฑ์หลายช่วงเวลาและปัญหาการจัดเส้นทางยานพาหนะแบบมีกรอบเวลาในการขนส่ง

โดย
นางสาวรัชฎาภรณ์ ภู่ห้อย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปรัชญาดุษฎีบัณฑิต
สาขาวิชาการจัดการงานวิศวกรรม แบบ 2.1 ปรัชญาดุษฎีบัณฑิต
ภาควิชาวิศวกรรมอุตสาหการและการจัดการ
มหาวิทยาลัยศิลปากร
ปีการศึกษา 2566

GENETIC ALGORITHM FOR MULTI-PRODUCT MULTI-PERIOD AGGREGATE PRODUCTION PLANNING AND VEHICLE ROUTING PROBLEMS WITH TIME WINDOWS

By

MISS Ratchadakorn POOHOI

A Thesis Submitted in Partial Fulfillment of the Requirements

for Doctor of Philosophy ENGINEERING MANAGEMENT

Department of INDUSTRIAL ENGINEERING AND MANAGEMENT

Academic Year 2023

Title	Genetic Algorithm for multi-product multi-period aggregate production planning and vehicle routing problems with time windows
By	MISS Ratchadakorn POOHOI
Field of Study	ENGINEERING MANAGEMENT
Advisor	Assistant Professor Dr. KANATE PUNTUSAVASE
Co advisor	Associate Professor Dr. Shunichi Ohmori

Faculty of Engineering and Industrial Technology, Silpakorn University in Partial Fulfillment of the Requirements for the Doctor of Philosophy

.... Dean of Faculty of Engineering and Industrial Technology
(Assistant Professor Dr. Arunsri Leejeerajumnean)

Approved by

.... Chair person
(Associate Professor Dr. Thananya Wasusri)

.... Advisor
(Assistant Professor Dr. KANATE PUNTUSAVASE)

.... Co advisor
(Associate Professor Dr. Shunichi Ohmori)

.... Committee
(Dr. SITICHAI SAELEM)

.... External Examiner
(Assistant Professor Dr. Nattapong Kongprasert)

630930003 : Major ENGINEERING MANAGEMENT

Keyword : Genetic Algorithm, Aggregate production planning, Vehicle routing problems with time windows, K-mean clustering

MISS Ratchadakorn POOHOI : Genetic Algorithm for multi-product multi-period aggregate production planning and vehicle routing problems with time windows Thesis advisor : Assistant Professor Dr. KANATE PUNTUSAVASE

Genetic Algorithm is the search algorithms and optimization methods. The basic concept is based on the mechanisms of evolution and natural selection, according to Darwin's theory of survival of the fittest. A novel crossover operator is a combination of four crossover operators, including Single point crossover, Two points crossover, Arithmetic crossover, and Scattered crossover, which is called "Stas Crossover". The most important advantage of Stas crossover is that it provides greater diversity in the choice of methods for creating offspring and increases the opportunity for offspring to directly obtain good genetic information. It presents the performance of the crossover operator, which tests with multi-product and multi-period aggregate production planning problems (APP), provides optimal levels of inventory, backorders, overtime and regular production rates, and other controllable variables, and finally chooses appropriate crossover options. Moreover, Stas crossover in GA was modified to solve the standard Solomon's benchmark problem instances for vehicle routing problems with time windows (VRPTW) by developing the problem with K-mean clustering. Results from K-mean clustering show that it performs better for minimum distance and average distance than without K-mean clustering. The paths with K-mean clustering are arranged into groups and are orderly, but the paths without K-mean clustering are disordered in terms of location and dispersion characteristics of the customer. After that, the research presents a comparison of the performance of the crossover operator with the instance of the Solomon benchmark, and it is recommended to use the appropriate crossover operator for each type of problem. It has been shown that adding K-mean clustering to the Stas crossover efficiently contributes to its performance. In some instances, the

results of Stas crossover are better than the known solutions from previous studies. Furthermore, the proposed research will serve as a guideline for a real-world case study.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to all those who helped me complete this dissertation successfully.

First and foremost, I would like to express my sincere gratitude to my advisor, Assist. Prof. Dr. Kanate Puntusavase, for his constant support, valuable guidance, and encouragement throughout the process. I am almost grateful for his assistance, not only for my research but also for providing me with invaluable insights and direction. I would not have achieved this far and this dissertation would not have been completed without all the support that I have always received from him. Besides my advisor, I would like to extend my thanks to the co-advisor, Assoc. Prof. Dr. Shunichi Ohmori, for providing all the necessary continued help and kind support to me. In addition, I am grateful for the committee members: Assoc. Prof. Dr. Thananya Wasusri, Dr. Sitichai Saelem, and Assist. Prof. Dr. Nattapong Kongprasert for suggestions and detailed feedback that have been very important to me. I also would like to thank all my dissertation members and all the other members of my department.

To my parent, Klasuwan and Pornpimol Poohoi, their constant encouragement, patience, unwavering support, and understanding have motivated me and boosted my morale during the completion of this dissertation.

Finally, I am also thankful to my sister Krittiya Keanchan, and my brother Nontharat Akakulananta for everything, they are beside me. I also extend my thanks to my friends for their encouragement and cooperation throughout this dissertation.

Ratchadakorn POOHOI

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Motivation

Genetic Algorithm (GA) is a population-based metaheuristic optimization algorithm. John Holland developed GA based on Darwin's theory of evolution in 1988 and expanded it in 1992 (Goldberg, 1989; Holland, 1992). It is a natural selection that is based on evolutionary algorithms. GA searches by biologically inspired operations, starting from the initial population. This operator starts with a random population to determine the population’s fitness for each chromosome. To improve the poor individuals in the first population, selection occurs when the operator chooses the individuals for the next generation in the total population through chromosome selection from the commonly used roulette wheel. This operator is based on probability for individuals and selects the next generation, which is proportional to fitness values. Good individuals may be chosen many times in a generation, but poor individuals may not be chosen at all. The chromosomes of selected parents are recombined. The mechanisms of crossover and mutation are generally used. Termination criteria are defined when the solution has no improvement. However, GA has been a popular search technique to find good solutions for complicated optimization problems.

Aggregate production planning (APP) is a method for developing an overall production plan to ensure production in a facility is not interrupted. APP is associated with the determination of production, inventory, and labor levels to fulfill varying demand over a planning perspective that ranges from a period of six months to one year (Krajewski & Ritzman, 1999). The aggregate plan generally includes forecasts of target sales, production levels, inventory levels, and customer backlogs. The goal is to minimize operating costs by matching production demand with production capacity. In ord er to minimize costs, APP will determine which materials and other resources are required and when to procure them. APP helps manufacturers maximize plant productivity and achieve financial goals. To meet customer demand, minimize the expense of excess inventory, and meet production capacity to the maximum benefit, this can be helpful.

The primary goals of APP are to minimize costs and maximize profits. The strategic objectives of aggregate planning include minimizing inventory investment, minimizing workforce demand and fluctuation, maximizing production rates while minimizing fluctuation, and optimizing facility and production equipment utilization. An APP problem can be a form of transportation problem. The transportation model is one example of a linear programming model. Finding the optimization can determine the results of the optimal objective function (maximize problem and minimize problem). It can be found in linear programming tools such as Lindo, MATLAB, AMPL, and Excel Solver for optimization. However, Excel Solver is well known in the industry, but the free version currently available is limited in terms of variables. That means it can be used in a limited range, too. In addition, users must have a complete understanding of it; otherwise, it may cause errors in the work. In this research, it is a program that uses more variables than the current free version. Users need not have knowledge of Excel Solver to use it. This is useful in solving problems like APP in the industrial sector.

APP has received a lot of attention from both practitioners and academics. Certain constraints are imposed on solving the APP problem, which demand constraint optimization. Various meta heuristic algorithms like simulated annealing, particle swarm optimization, and genetic algorithm (GA) have been used by many researchers in solving the APP problem. Chakrabortty and Md.A.Akhtar. (2013) developed an interactive Multi-Objective Genetic Algorithm (MOGA) approach for solving the multi-product, multi-period APP with forecasted demand, related operating costs, and capacity. Savsani et al. (2016) described a GA with different selection methods and diverse crossover operators for solving APP. Natural selection and natural genetics are the foundations of GA search algorithms. They combine survival of the fittest among string structures with a structured yet randomized information exchange to form a search algorithm with some of the innovative flair of human search (Goldberg, 1989).

Dantzig and Ramser (1959) introduced Vehicle routing problem (VRP) in 1959 called the truck dispatching problem. VRP is significant in the transportation and logistics industries. It is classified as an NP-hard problem. Vehicle Routing Problem with Time Window (VRPTW) is an important kind of classical VRP in which time window constrain. The objective function minimizes the total distance traveled or the number of vehicles used. In the VRPTW a number of vehicles is limited, it starts from a central depot to serve dispersed customers in the time window with limited capacity and terminates at the depot (Cordeau et al., 2000).

Vehicle Routing Problem (VRP) is important in the transportation and logistics industries. One type of VRP with an additional time windows constraint in the model is Vehicle Routing Problem with Time Window (VRPTW). The goal is to minimize the total distance traveled, or the number of vehicles used, and identify vehicle routes. The problem can be described as finding routes with a limited number of vehicles and each vehicle having a limited capacity. It starts from a central depot to service only one customer within the time windows and ends at the depot (Kallehauge et al., 2005; Ariyani et al., 2018; Thangiah, 1995). According to Ahmed et al. (2023), VRPTW is categorized as an NP-hard problem, meaning that the computational complexity required to solve it increases exponentially with problem size.

Genetic Algorithm (GA) is a popular algorithm for solving VRPTW problems. May et al. (2021) research proposes developing the problem-specific crossover and seven various mutation operators to offer a new improved GA to solve various VRPTW with the hard time windows. Ghani et al. (2016) studied how to assign a number of vehicles to a customer and depot to minimize the total distance traveled and achieve delivery operations within the time windows that the customer required. Kinoshita & Uchiya

(2021) propose a method to ensure optimization accuracy while preserving dynamic switching with multiple crossovers based on gene population diversity.

In VRPTW, Solomon benchmarks are the most popular problem to solve. The problem is divided into six problem sets, suggested as C1, R1, RC1, C2, R2, and RC2, each of which represents a different problem type (Solomon, 1987; Solomon & Desrosiers, 1998). There are between eight to twelve 100-node problems in each set. These six sets of problems, including Set C has generated the customer cluster. Set R has generated uniformly random locations, whereas Set RC has a combination of Set C and Set R. According to Solomon (2005) and Gambardella (2000) Type 1 has narrow time windows and a small vehicle capacity, whereas Type 2 has large time windows and a large vehicle capacity.

Based on several studies that have been done before, it is a long evolution phase for GA algorithms. Therefore, the authors see that there are good opportunities for future contributions. In this study, the author used a popular meta heuristic, GA, to solve the APP problems and VRPTW problems. For APP problems considered a multi-product, multi-period to minimize total costs in terms of regular time, overtime, backordering, and inventory costs. A detailed comparison is presented of a GA approach for solving APP problems by using four different crossover options and new crossover to compare the behavior of the crossover and choose appropriate crossover options for solving the APP problems. For VRPTW problems considered GA with a new crossover to solve by developing the problem with K-mean clustering to perform better for minimum distance and average distance for Solomon's benchmark problem instances. A detailed comparison is presented with Solomon's benchmark problem instances for VRPTW to compare the results with and without K-mean clustering. This work developed a novel interactive crossover approach, considering four crossover options as well as creating a new crossover option for APP problems. After that, the author developed the new crossover with K-mean clustering to solve Solomon's benchmark problem instances for VRPTW.

1.2 Research Objectives

1.2.1 To create a new crossover operator in a genetic algorithm approach for using Microsoft Visual Studio.

1.2.2 Apply a new crossover and using four different crossover options to solve multi-product multi-period APP problems and compared behavior of crossover and based on different statistical values.

1.2.3 Apply a new crossover with K-mean clustering and using four different crossover options to solve Solomon's benchmark problem instances for VRPTW.

1.3 Research Contribution

1.3.1 Discover a new crossover operator in a genetic algorithm, it's called "Stas crossover".

1.3.2 Be able to choose appropriate crossover options for solving multi-product multi-period APP problems.

1.3.3 Be able to choose appropriate crossover options for solving Solomon's benchmark problem instances for VRPTW with K-mean clustering and without K-mean clustering.

CHAPTER 2
THEORY AND LITERATURE REVIEWED

In this study, the author used a popular meta heuristic, GA, to solve the APP problems and Solomon's benchmark problem instances for VRPTW. Here, the author considered two parts, including a multi-product and multi-period APP problems and Solomon's benchmark problem instances for VRPTW. The author study concepts, theories, and related research to be used as a guide to explain and study the following content.

2.1 Aggregate production planning

Aggregate production planning is a method to reduce costs and develop all overall manufacturing plans. It is concerned with the determination of production, inventory, and labor levels to fulfill shifting demand requirements over a six-month to one-year planning horizon. The following seasonal peak in demand is usually factored into the planning horizon. The planning horizon is often divided into periods (Gallego, 2021). The main goal of APP is to minimize costs of operation to optimize manufacturing which matches production demand with production capacity. It determines the level of production, inventory, and labor needed to meet changing demand and also informs manufacturers on the costs of labor, materials, productivity, timetable forecasts, and the budget.

The strategic objectives of APP with the main goal of minimizing costs and maximizing profits, include minimizing inventory costs, balancing efforts to minimize inventory management and storage, ensuring enough inventory to meet needs, and minimizing the workforce. APP uses data from forecasting demand to calculate a balanced workforce. Moreover, maximizing facility utilization determines the maximum facility utilization for the over-planned period. By achieving the strategic objectives, manufacturers can balance short- and long-range production plans that meet demands and optimize manufacturing profits.

Short-range plans, intermediate-range plans, and long-range plans are the three levels of APP. The responsibilities of the top management, operations managers with the sales and operations planning team, and operations managers, supervisors, and foremen are depicted in Figure 1 (Heizer et al., 2017). The term APP refers to the planning that is done for a single measure of overall output; at least it only includes a few product categories. Each forecast is appropriate for each planned period.

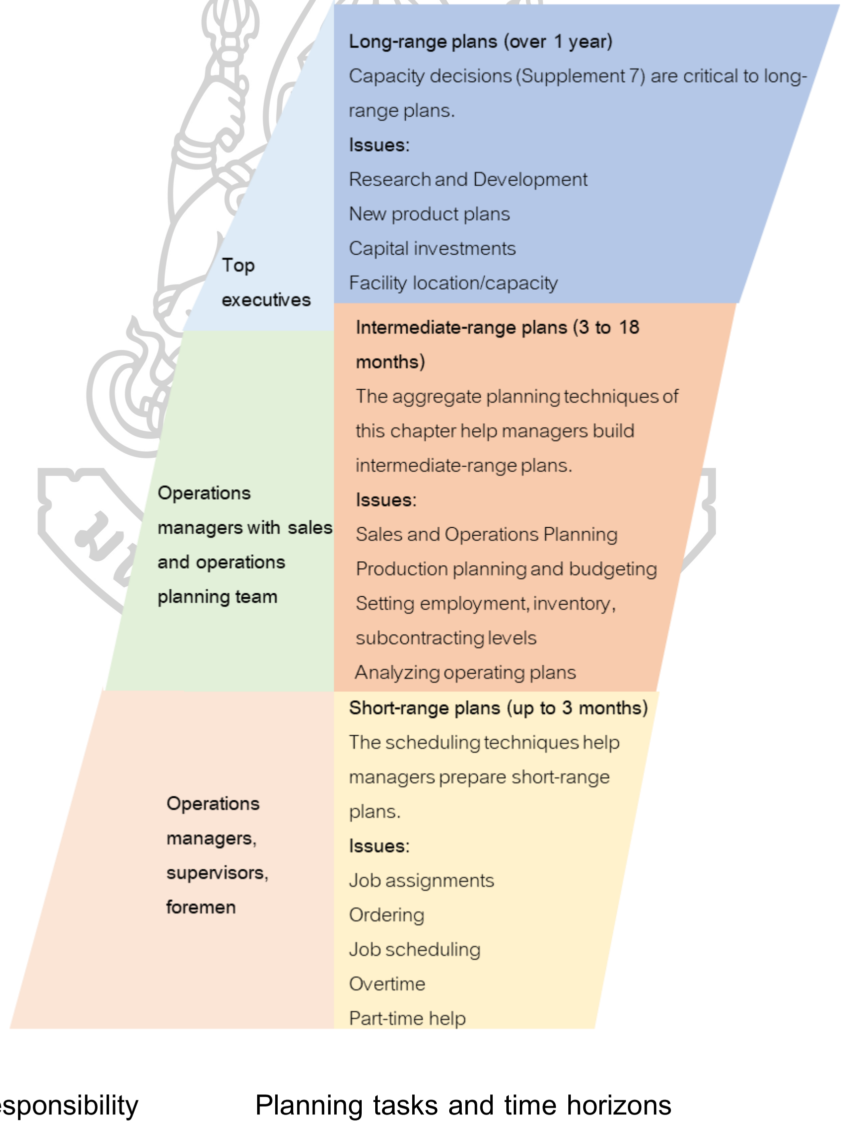

Figure 1 Planning Tasks and Responsibilities

The formulation of a linear programming model for the APP planning problem with multi-product multi-period is discussed.

Parameters

Rijt Regular time production cost for product i manufactured from source j in period t ($/units)

Oijt Overtime production cost for product i manufactured from source j in period t ($/units)

Vijt Inventory carrying cost for product i from source j in period t ($/units)

Bijt Backorder cost for product i m from source j in period t ($/units)

Variables

Dit Forecasted demand of product i in period t (units)

Xijt Production Quantity of product i manufactured from source j at regular time in period t (units)

Yijt Production Quantity of product i manufactured from source j at overtime in period t (units)

Wijt Inventory of product i in source j at the end of period t (units)

Mijt Backorder of product i in source j at the end of period t (units)

Objective function

$$\text{Min } Z = \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{t=1}^{T} \left[R_{ijt}X_{ijt} + O_{ijt}Y_{ijt} + V_{ijt}W_{ijt} + B_{ijt}M_{ijt} \right] \quad (1)$$

Constraint

$$W_{ijt} - M_{ijt} = W_{ij(t-1)} - M_{ij(t-1)} + X_{ijt} + Y_{ijt} - D_{it} \quad \text{for } \forall i \forall t \quad (2)$$

$$W_{ijt} \geq W_{ijt\,min} \quad \text{for } \forall i \forall t \quad (3)$$

$$M_{ijt} \leq M_{ijt\,max} \quad \text{for } \forall i \forall t \quad (4)$$

$$X_{ijt}, Y_{ijt}, W_{ijt}, M_{ijt} \geq 0 \quad \text{for } \forall i, \forall t \quad (5)$$

Equation (1) is the objective function to find the optimal APP that minimizes the sum of production costs, including regular time production, overtime production, carrying inventory, and backordering costs. Equation (2) – (4) represents the constraint on carrying inventory in which the forecast demand Dit cannot be obtained precisely in a dynamic market. Demand for a specific period can be fulfilled or backordered, but backorders already made must be completed in the next period. Equation (5) determines the constraint of non-negative.

2.1.1 Aggregate Planning Strategies

These are valid planning techniques. Inventory, production rates, workforce levels, capacity, and other controllable variables are all manipulated (Heizer et al., 2017). There are two types of planning methodologies for aggregate planning: capacity options and demand options. The fundamental capacity options include inventory levels that change, different workforce sizes through hiring or layoffs, production rates that change with overtime or idle time, subcontracting, and part-time labor. Influencing demand, backordering during high-demand seasons, and counter-seasonal product and service mixing are the core demand choices. Table 1 summarizes the advantages and disadvantages of aggregate planning options (Heizer et al., 2017).

Table 1 Aggregate Planning Options: Advantages and Disadvantages

Option	Advantages	Disadvantages	Comments
Changing inventory levels	Human resources or none, the shift is gradual. There will be no dramatic changes in manufacturing.	The cost of storing inventory may rise. Sales may be lost during periods of excessive demand.	This is mostly a production issue, not a service issue.

Table 1 Aggregate Planning Options: Advantages and Disadvantages (Continue)

Option	Advantages	Disadvantages	Comments
Varying workforce size by hiring or layoffs	It avoids the costs associated with other options.	Costs of hiring, laying off, and training may be incurred.	When the labor pool is vast, this method is used.
Varying production rates through overtime or idle time	Seasonal variances can be accommodated without the need for additional employment or training.	High overtime rates and weary staff may make it difficult to meet demand.	This allows for some flexibility within the overall strategy.
Subcontracting	This allows for more flexibility and improves company efficiency.	Quality of goods suffers, profits suffer, and future business opportunities are lost.	It's mostly used in production environments.
Using part-time workers	Full-time employees are more expensive and have less flexibility.	High turnover/training costs; poor quality; problematic scheduling	Unskilled jobs in locations with a high temporary workforce are ideal.
Influencing demand	Make the most of the extra capacity. New clients are attracted by discounts.	Demand is erratic. It's difficult to precisely match supply and demand.	Produces marketing concepts. Some businesses use overbooking.

Table 1 Aggregate Planning Options: Advantages and Disadvantages (Continue)

Option	Advantages	Disadvantages	Comments
Back ordering during high-demand periods	It's possible to avoid working overtime. Maintain a consistent capacity.	Customers must be willing to wait, or else their goodwill will be lost.	Many businesses are prepared to wait.
Counter seasonal product and service mixing	Utilize all available resources and maintain a consistent workforce level.	Outside of the company's competence, skills or equipment may be necessary.	Finding items or services with opposing demand patterns might be dangerous.

The required information is shown in Figure 2, and the APP's results include demand forecasting with appropriate techniques, capacity, and financial constraints, as well as strategic objectives (Heizer et al., 2017). This information must be accurate and reliable. The APP's outputs include workforce size (the number of personnel needed), production levels, inventory levels, and other costs such as subcontract wages and backorder delivery penalties.

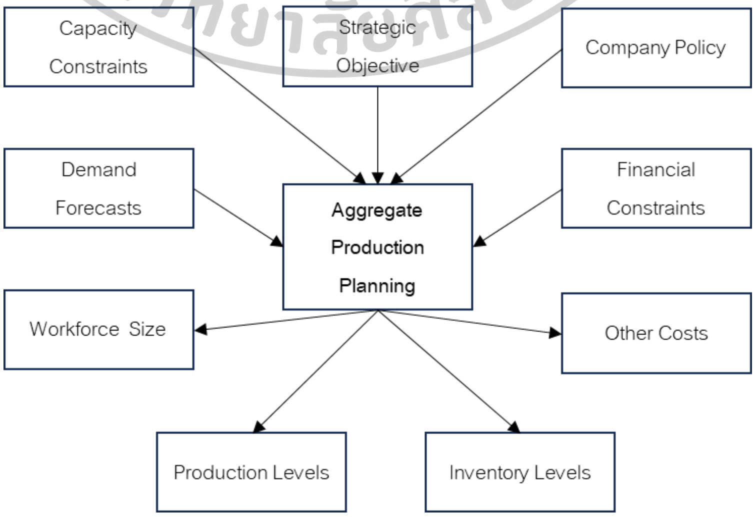

Figure 2 Relationships of the Aggregate Production Planning

2.2 Transportation Modeling

When considering alternative facility locations within the framework of an existing distribution system, the transportation model described in this module may prove useful (Heizer et al., 2017). It is an iterative problem-solving process that involves minimizing the cost of shipping products from multiple sources to multiple destinations. It must be aware of the following to use the transportation model:

1. The sources, as well as their capability or supply per period.
2. The destinations and demands for each period.
3. From each source to each destination, transportation costs one unit.

The transportation model is one form of a linear programming model. Software is available to solve both transportation problems and the more general class of linear programming problems. So, the aggregate production planning problem can be a form of transportation model, using the transportation simplex method for optimal.

Figure 3 shows that the n units required by destination may be shipped in various combinations from m source and how much it costs to ship from each source to each destination.

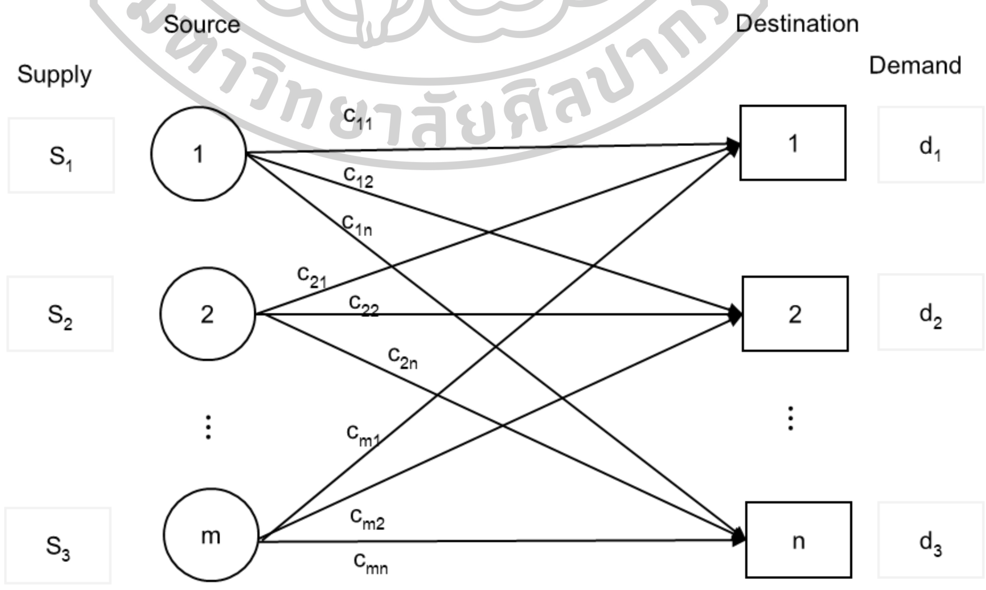

Figure 3 Transportation Problem

In the modeling process, the goal is to set up a transportation matrix. Its purpose is to summarize all relevant data and keep track of algorithm computations. It can construct a transportation matrix as shown in Table 2.

Table 2 Transportation Matrix

To From	Destination 1	Destination 2	Destination 3	Supply
Source 1	c_{11}	c_{12}	c_{13}	a_1
Source 2	c_{21}	c_{22}	c_{23}	a_2
Source 3	c_{31}	c_{32}	c_{33}	a_3
Demand	b_1	b_2	b_3	Total supply Total demand

2.3 Travelling Salesman Problem

The traveling salesman problem (TSP) is an optimization problem and one of the most popular problems of NP problem. The purpose of TSP is to minimized the total distance travelled by the travelers. The problem is finding the shortest path possible through a set of n vertices such that each vertex is visited only once. The problem is to find the shortest path possible which considered a number of cities N and distance between cities (Tawanda et al., 2023). The travelers have traveled all cities or find a closed tour, each city can only be traveled through once which returned to the same city from the starting point as described in Figure 4.

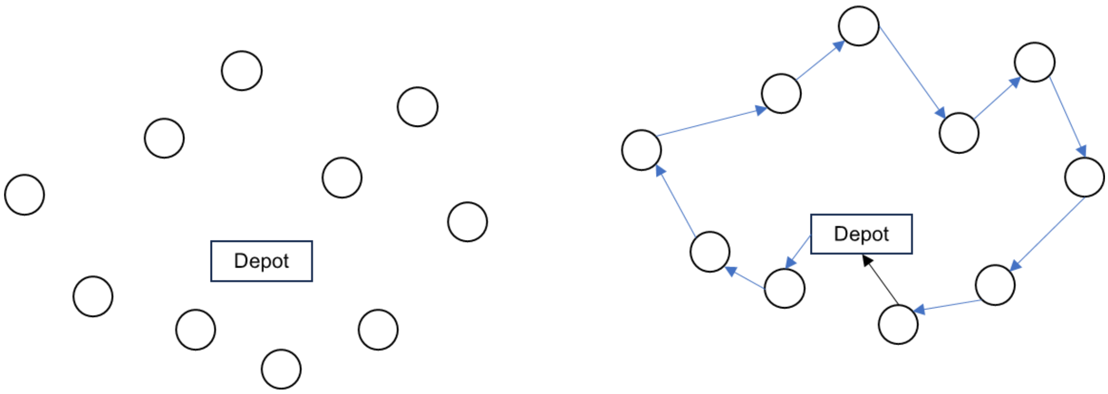

Figure 4 Traveling Salesman Problem

In a complete weighted undirected can be illustrated by graph G = (V, A), where V = {1, …, N} is set of node and A is node of connected. The distance Cij is the distance of traveling from node i to node j, $(i, j) \in V | i \neq j$ and S is subsets, V: $S \subset V$, $S \neq \emptyset$, $S \neq V$.

Parameter

Cij total distance of traveling from node i to node j

Decision Variables

$$X_{ij} = \begin{cases} 1, & \text{the path traveling from node i to node j} \\ 0, & \text{otherwise} \end{cases}$$

Objective Function

$$\min \sum_{i=1}^{n} \sum_{j\neq i,\, j=1}^{n} C_{ij} X_{ij} \qquad (1)$$

$$\sum_{i=1, i\neq j}^{n} X_{ij} = 1 \qquad \forall_{j = (1,2,\ \ldots,N)} \qquad (2)$$

$$\sum_{j=1, j\neq i}^{n} X_{ij} = 1 \qquad \forall_{i = (1,2,\ \ldots,N)} \qquad (3)$$

$$\sum_{i\in S} \sum_{j\neq i, j\in S} X_{ij} \leq |S| - 1 \qquad \forall_{S \subset V} \qquad (4)$$

$$X_{ij} \in \{0,1\} \qquad \forall_{i = (1,2,\ \ldots,N),\ j = (1,2,\ldots,N)} \qquad (5)$$

Equation (1) is represented the shortest closed tour from the total distance travelled by the travelers. Equation (2) and (3) is constraint of each city include a single tour entrance and single tour exit. Equation (4) is a subtour elimination constraint and Equation (5) is defined binary decision variables.

2.4 Vehicle Routing Problems with Time Windows

Vehicle routing problem (VRP) is the most important in the transportation and logistics industries. It is related to Traveling Salesman Problem. Dantzig and Ramser (1959) started the first in VRP problem which included the routing of a fleet of fuel tanker trucks between the bulk terminal and the various service stations that the terminal offered. After that, it has been many research which expanded about this scope. VRP is NP-hard problem that determines the optimal solution. It may be limited by the size of the problems to find the optimal solution by using mathematical programming or combinatorial optimization, so these heuristics have been used to solve real-world problems. The objectives of VRP are different; it depends on the specific application of the results. However, the common objectives include minimizing transportation costs based on transportation plans and routes, minimizing the number of vehicles that can be serviced by all customers, and minimizing travel time (Toth & Vigo, 2002). Figure 5 illustrates VRP; it can describe the vehicles starting from the depot, visiting each customer exactly once, meeting the demand of the customer, and returning to the end at the depot (Kumar & Panneerselvam, 2015).

Figure 5 Vehicle Routing Problem

The VRPTW is defined by a set of homogeneous vehicles denoted by V, a set of customers C, and directed graph G = (V, C). Each vehicle has a capacity C(k) which each customer i a delivery demand (di) for service time at customer i s(i). A vehicle has arrival time at customer i before T(i). It can arrive before e(i) but can't arrive after l(i). The distance can be traveled from customer i to customer j (Dij). The formulation model for VRPTW is described. The following are the symbols to describe the model.

Parameter

N = number of customers

K = number of vehicles

D(ij) = distance that can be traveled from customer i to customer j

d(i) = delivery demand of customer i

C(k) = capacity of vehicle

T(i) = arrival time at customer i

e(i) = earliest arrival time at customer i

l(i) = latest arrival time at customer i

s(i) = service time at customer i

Decision Variable:

$$x(ijk) = \begin{cases} 1, & \text{if the vehicle k travels from customer i to customer j} \\ 0, & \text{otherwise.} \end{cases}$$

$i \neq j$; i,j $\in$ (Tawanda, Nyamugure, Kumar, & Munapo, 2023…,N);

0 refers to depot.

Objective function:

$$\text{minimize}\left(\sum_{i=0}^{N}\sum_{j=0}^{N}\sum_{j\neq 1,k=1}^{K} D(ij).x(ijk)\right) \quad (1)$$

Subject to

$$\sum_{j=0}^{N} x(ijk) = 1 \quad (2)$$

$$i = 0 \text{ and } \forall\, k \in K$$

$$\sum_{j=0}^{N} x(ijk) \leq K \quad (3)$$

$$i = 0$$

$$\sum_{k=1}^{N}\sum_{j=0,j\neq i}^{N} x(ijk) = 1 \quad (4)$$

$$\sum_{k=1}^{N}\sum_{i=0,i\neq j}^{N} x(ijk) = 1$$

$$\forall\, i \in N$$

$$\forall\, j \in N$$

$$\sum_{i=1}^{N} x(ijk) - \sum_{i=1}^{N} x(ijk) = 0 \quad (5)$$

$$\forall\, i \in N$$

$$\forall\, k \in K$$

$$\sum_{i=1}^{N}\sum_{j=0,j\neq 1}^{N} d(i).\, x(ijk) \leq C(k) \quad (6)$$

$$\forall\, k \in K$$

$$e(i) \leq T(i) + s(i) \leq l(i) \quad (7)$$

Equation (1) is the objective function to minimize the distance traveled to serve customers by all vehicles in which each vehicle has limited capacity within the time windows requested by the customer. Equation (2) defines each vehicle starting from the central depot and ending at the depot. Equation (3) represents the vehicles number at the depot, that means the number of routes. Equation (4) requires that each customer can be visited only once by one of the vehicles from the depot. Equation (5) constraint that the same vehicle must enter and leave from that customer. Equation (6) states that the demand of each customer on each vehicle route needs to be less than or equal to the vehicle capacity. Equation (7) determines that vehicles cannot arrive before the earliest arrival time and must not be later than the latest arrival time.

One type of VRP is the Vehicle Routing Problems with Time Windows (VRPTW), which is an additional time windows constraint in the model. The goal is to design a set of routes that minimize the total distance traveled, or the number of vehicles used, and identify vehicle routes. The problem can be described as follows: each customer is served exactly once in the time windows, where every route originates at the central depot and terminates at the central depot, and the capacity of each vehicle has not been exceeded.

For the VRPTW, a set of well-known benchmark problems is Solomon benchmarks. Solomon benchmark instances are divided into six sets, which include C1, C2, R1, R2, RC1, and RC2 (Solomon, 1987; Solomon & Desrosiers, 1998). Each set contains between eight and twelve instances. In C1 and C2, the customer has been generated and placed in a cluster. R1 and R2 have randomly generated the locations with random distribution, and finally, in RC1 and RC2, some customers have been placed in clusters and others have been placed randomly. For sets C1, R1, and RC1, the problems have narrow time windows to be serviced, and few accommodate customers. For sets C2, R2, and RC2, the problems have large time windows to be serviced, and a lot of accommodate customers. For each instance has 25 customers, 50 customers, and 100 customers. (Kallehauge et al., 2005; Solomon, 2005; Gambardella, 2000).

2.4 Genetic Algorithm

The genetic algorithm (GA) is a natural selection-inspired optimization algorithm. According to Darwin's theory, natural selection favors the fittest individuals who reproduce. This concept was developed by Goldberg (Goldberg, 1989). It is a population-based search algorithm that uses the concept of the survival of the fittest. The new populations are created by iteratively using genetic operators with individuals present in the population. In general, GA consists of five basic components (Carr, 2014), including a fitness function for optimization, a population of chromosomes, selection of which chromosomes will reproduce, crossover to produce the next generation of chromosomes, and random mutation of chromosomes in the new generation. These are all attempts to copy the mechanisms of nature. GA is mostly used to heuristically find and locate the global optimal solution. The steps of GA are summarized in Figure 6.

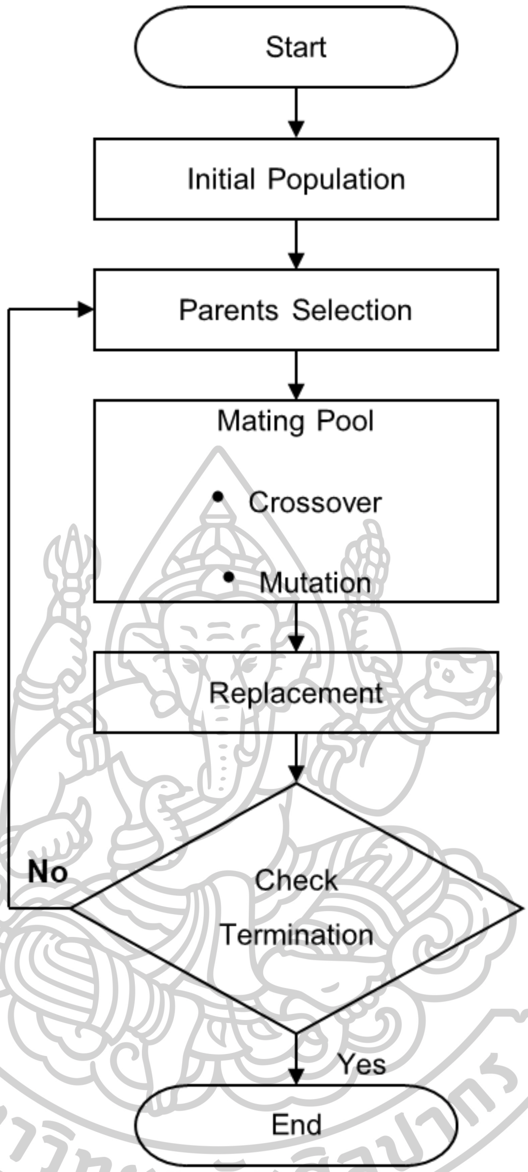

Figure 6 Genetic algorithm steps

2.4.1 Foundation of Genetic Algorithms

GA are based on comparisons with the genetic structure and behavior of chromosomes of the population. It has the following general procedure (Michalewicz Z., 1994):

Procedure: Genetic Algorithms

```
begin
        t ← 0;
        initialize P(t);
        evaluate P(t);
        while (not termination condition) do
        begin
                recombine P(t) to yield C(t);
                evaluate c(t);
                select P(t+1) from P(t) and C(t);
                t ← t+1;
        end
end
```

In the GA, the initial population is the first step. The individuals in the population, which represent legal solutions to a problem, are created and initialized at this step. In the current generation, the population is a subset of solutions. The initial population technique is depicted in Figure 7.

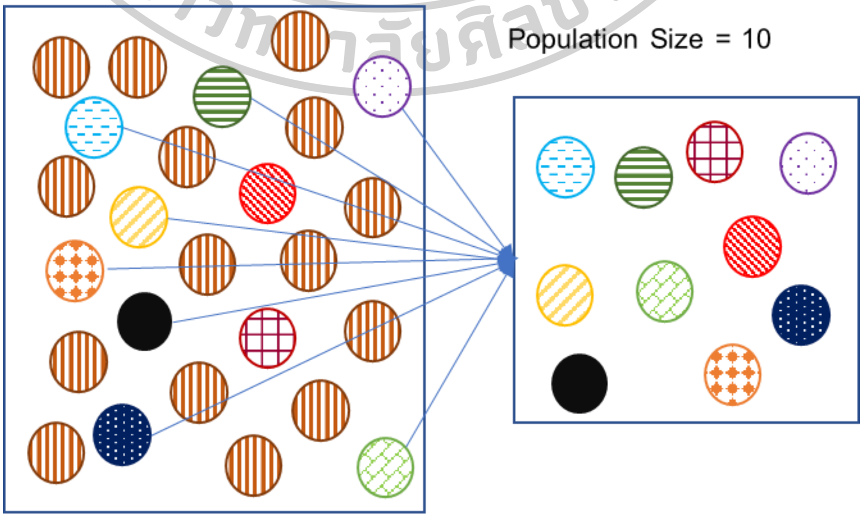

Figure 7 Initial Population Procedure

The fitness function determines the individual's fitness level (an individual's ability to complete with others). It gives each individual a fitness score. The fitness score determines the probability of an individual being chosen for reproduction.

Selection is when the operator chooses the individuals for the next generation in the total population chromosome selection. Here the author used only the Roulette Wheel selection. Roulette Wheel selection is based on probability. It simulates a roulette wheel, with the areas proportional to their fitness. Then, the roulette wheel is rotated, and the individuals are selected at random. With it, an individual with better fitness is, the more likely to be selected.

Roulette wheel selection, the basic part of the selection process is to stochastically select from one generation to create the basis of the next generation (UNSW, 2022). The criteria state that the fittest individuals have a better chance of surviving than those who are weaker. Fitter individuals will have a better chance of survival and will go on to form the mating pool for the next generation, just as they do in nature. Individuals that are weaker have a chance. Such individuals may have genetic coding that will be valuable to future generations in nature.

This method of selection is based on chance and is also known as fitness proportionate selection. Essentially, the probability of a hypothesis being chosen is determined by the ratio of the hypothesis' fitness to the sum of the fitness values for all the members of the population. One way of implementing this is to use the following algorithm:

1. Determine the total S of the hypotheses' fitness values in the population.
2. Select a random number r from the interval (0, S)
3. Iterate through the population and, for each hypothesis, sum the fitness values of all preceding hypotheses to get a value s. When s is greater than r, stop and return to the current hypothesis as the selected hypothesis.

This can also be compared to a roulette wheel in a casino, where each hypothesis is given a fraction of the wheel equal to its fitness divided by the sum of fitness values in the population, and thus fitter hypotheses have a larger area of the wheel and a higher likelihood of selection, as shown in Figure 8 (Dalton, 2007).

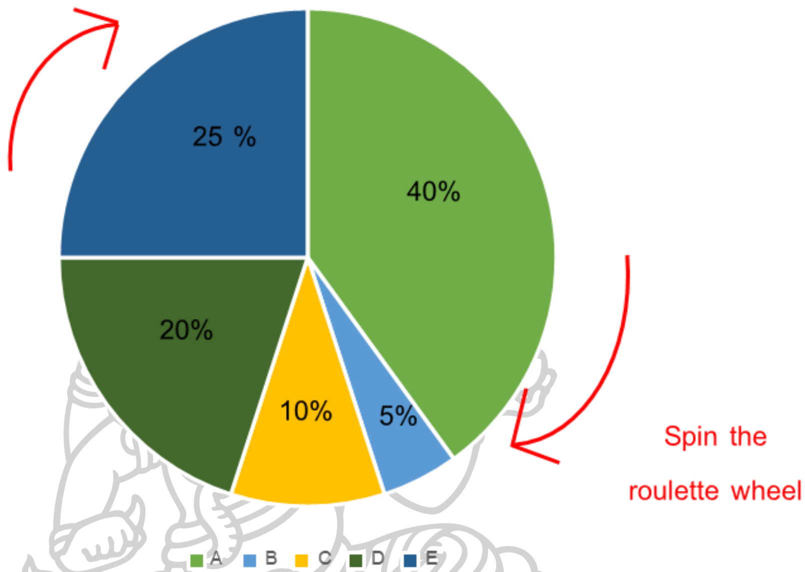

Figure 8 Roulette wheel approach: based on fitness

Crossover is a genetic operator used to specify a new individual or child by combining two individuals, or parents for the next generation (Chakrabortty & Hassin, 2013). Two individuals, or parents, are randomly selected from the mating pool to crossover to produce offspring. Here the authors choose four different crossover options including Single point crossover, Two points crossover, Arithmetic crossover, and Scattered crossover. Single point crossover and Two point crossover have a drawback when the chromosomes have similar characteristics in that chromosomes cannot be passed good genetic information, and none of the offspring get directly two good features. Arithmetic crossover is defined with linear constraints by taking the weighted average of two parents. If the parent has a better fitness value, it will return to the new offspring. Scattered crossover can remove similar characteristics in the chromosome representation to get new offspring with good genetic information (Shukla et al., 2010; Sivanandam and Deepa, 2008; Kaya et al., 2011).

Single point crossover: it chooses a random integer as the point of crossover and chooses the data beyond that point to swap between two parents to get new offspring (Shukla et al., 2010). For example, if p1 and p2 are the parents such as p1 = [a b c d e f g h] and p2 = [1 2 3 4 5 6 7 8] and the random integer for the point of crossover is 3. Then, the child would have [a b c 4 5 6 7 8] in Figure 9.

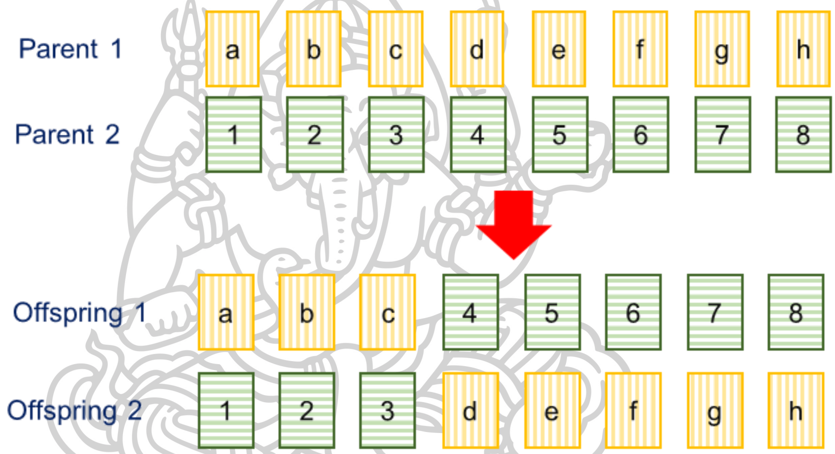

Figure 9 Single point crossover

Two points crossover: it chooses two random integers as points of crossover, which means the N-point crossover technique. Two random points are selected on the chromosome. The values for the head section and tail section are taken from the first parent, and the middle section is taken from the second parent to get new offspring (Shukla et al., 2010). For example, if p1 and p2 are the parents such as p1 = [a b c d e f g h] and p2 = [1 2 3 4 5 6 7 8] and the random integer for point of crossover are 2 and 4. Then, the child would have [a b c 4 5 6 g h] in Figure 10.

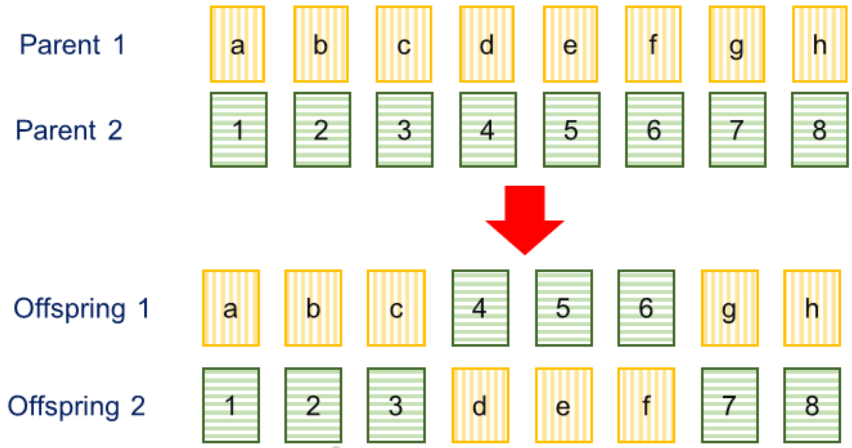

Figure 10 Two points crossover

Arithmetic crossover: it creates a new offspring that is the weighted arithmetic of two parents. The operator concerns a linear combination; according to the following equation, a is a random weighting factor between [0,1] and shows in Figure 11: (Kaya et al., 2011)

Offspring1 = a*Parent1+(1-a) *Parent2

Offspring2 = (1–a) *Parent1+a*Parent2

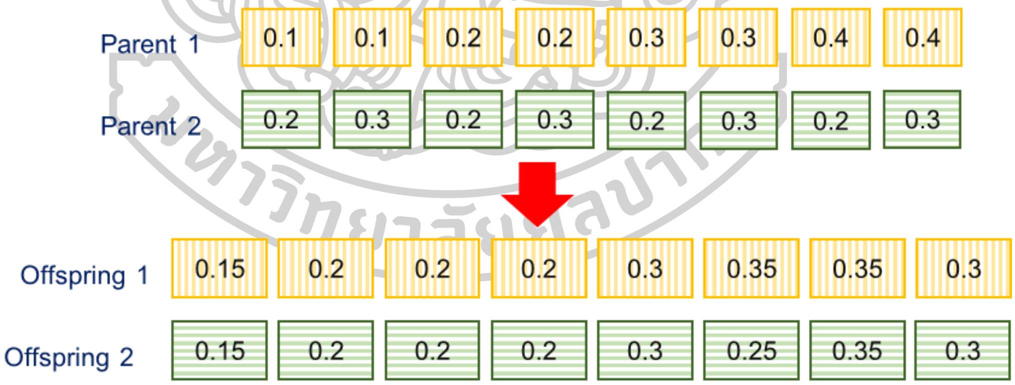

Figure 11 Arithmetic crossover

Scattered crossover, it creates a random binary vector by generating the size of the chromosome. To get the new offspring, it is taken from the first parent, where the vector is 1, and from the second parent, where the vector is 0 (Stoykova & Spasov, 2019). For example, if p1 and p2 are the parents such as p1 = [a b c d e f g h] and p2 = [1 2 3 4 5 6 7 8] and the binary vector is [1 1 0 0 0 1 0 0 0]. Then, the function returns the following child1 = [a b 3 4 e 6 7 8] in Figure 12.

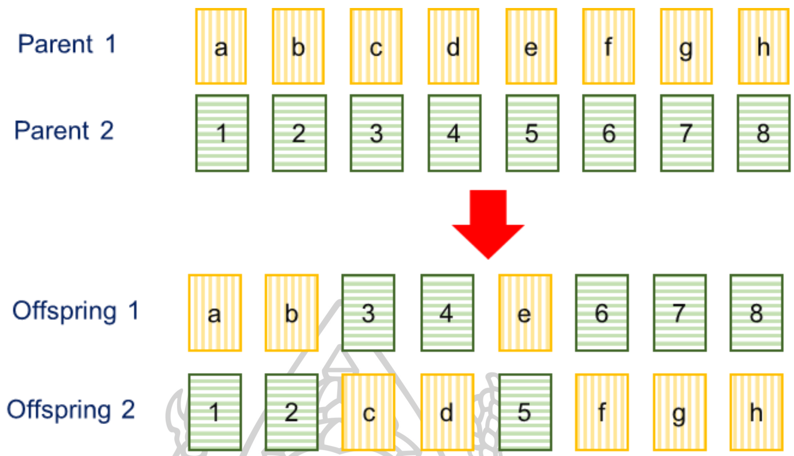

Figure 12 Scattered crossover

Mutation is the next variation operator after the crossover is done. This operator is based on the chromosome representation, but it is up to you to decide how to apply mutation. It is creating good new offspring while avoiding local optimal maintenance of diversity within a population and avoiding early convergence. Figure 13 shows example of mutation operator.

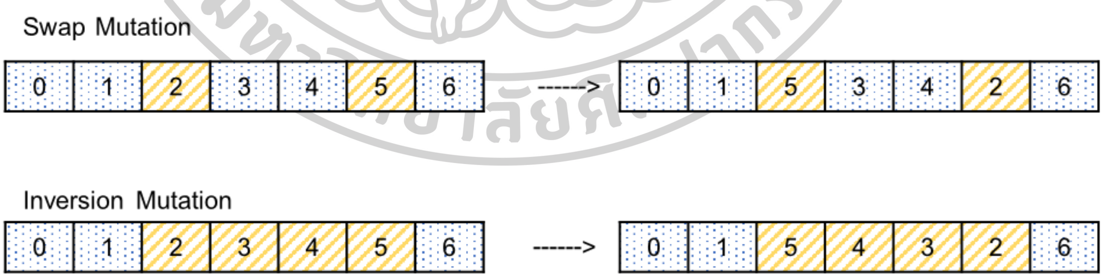

Figure 13 Mutation: Before and After

Stopping criteria when the stopping criteria are met (REGUANT, 2021). It can be defined as a set of rules to stop computation because the best solution is unknown. Accordingly, there is a solution that meets the minimum criteria. There are already too many iterations. The algorithm can't seem to find something better. The solution is good enough to satisfy the results.

Replacement is the process in which a previous chromosome is replaced by a new chromosome. This new generation of population's chromosomes will be better because they passed the selection options. The best result should be a chromosome that has the highest fitness function value.

Termination condition is a common practice of GA when a predetermined number of generations is reached. The quality of the best members of the population is tested against the problem definition. If no acceptable solutions are found, the GA may be restarted or a fresh search be initiated. Termination criteria include no improvement in the solution for x iterations, reaching a prespecified number of generations, and the objective function value reaching a prespecified threshold.

2.5 K-mean clustering algorithm

K-mean clustering algorithm was discovered in 1967 by James MacQueen (1967). K-mean is the most popular clustering algorithm. The algorithm starts by generating a number of k cluster centers randomly; each data is initially assigned to the nearest cluster. The Euclidean distance metric measure is typically applied to the standard K-means method to calculate the distance between data objects and cluster centers. Therefore, the cluster depends on the first cluster center selections. Selecting an appropriate value for k is important in K-mean algorithm. The performance of the algorithm depends on the specified k values, with different k values providing different results (Ikotun et al., 2023: Hartono et al.,2015: Suryawanshi & Puthran,2016).

After all data objects are completed, some data objects may be moved from one cluster to another. The centroid of each cluster is updated depending on the newly included data items. This process repeats the definition and updating of centroids until the movement of data objects between clusters stops and convergence is achieved, as shown in Figure 14 (Awawdeh et al., 2019). However, K-mean has many advantages, such as being easy to implement and easy to understand, but it also has some disadvantages, such as being sensitive to the initial position of the centroid and having to specify the number of clusters before.

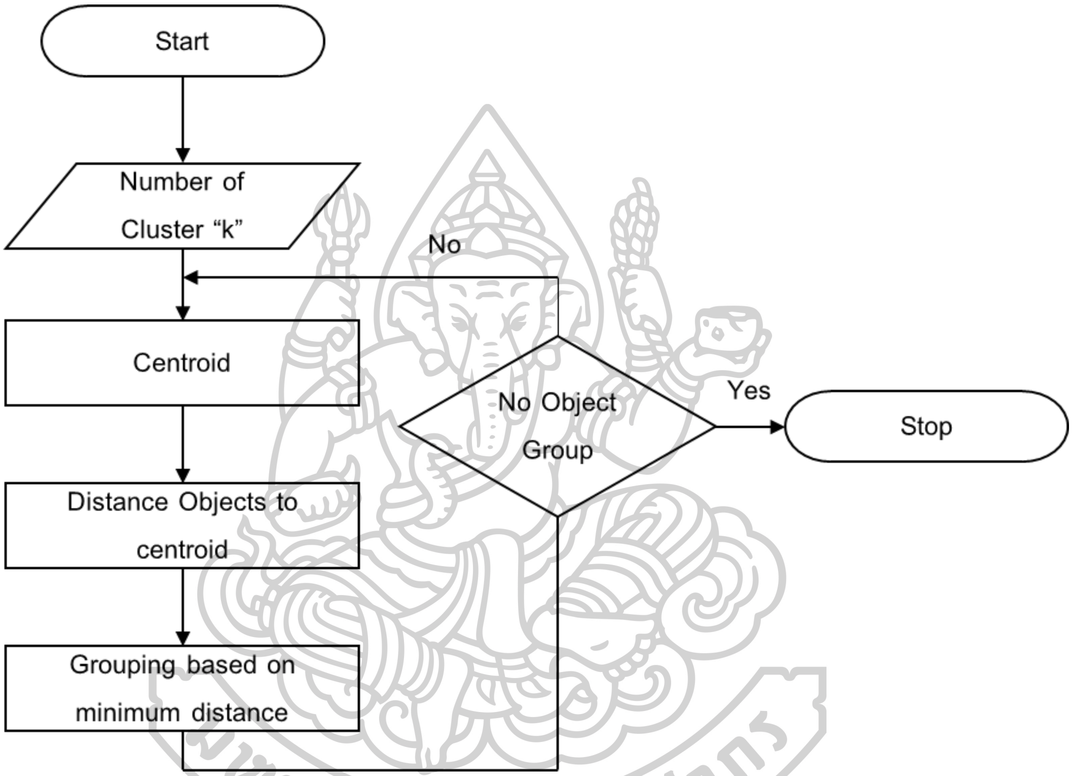

Figure 14 Flowchart of K-Means Clustering

The theory and case study for the research guidelines found in the reviewed literature show that the genetic algorithm is used to solve the proposed aggregate product model. This algorithm is a heuristic algorithm for searching that is based on the concept of natural selection, which occurs in genetic and evolutionary processes and has become very popular in solving APP.

APP is the medium-term planning horizon that determines the optimum production and workforce levels for each period to meet customer demand with minimum cost. Cheraghalikhania et al., (2019) reviewed the literature on APP models and classification schemes, which categorize them into two main purposes. The first perspective is the structure of the APP model, including the level of uncertainty contained in the APP model and the number of objective functions contained in the model. The second perspective is based on some extra problems that are added to the basic problems of the APP model. Jamalnial et al., (2019) reviewed the literature on APP under uncertainty. First, the preliminary analysis of the literature regarding the classification schemes with advantages and disadvantages of these methodologies was presented. Finally, more detailed analysis of the surveyed literature from management science and operations management perspectives was followed. Mohamad et al., (2012) presented a multi-objective model to deal with a multi-period multi-product multi-site APP problem for a medium-term planning horizon under uncertainty. The objective function attempts to minimize the sum of the expected value and the variability of total costs and maximize worker productivity, by calculating a weighted average of productivity levels in all factories and in all periods, which is weighted by the number of k-level labor. The results show the practical feasibility of the proposed multi-objective stochastic model as well as the proposed algorithm.

Certain constraints are defined for solving APP problems, which demand constraint optimization. Stockton and Quinn (1995) examined the aggregate planning process and described the basic tasks involved. The main limitations of the current approaches to the aggregate plan method are then compared. Then, it explains how GAs can be used to develop such plans and illustrates how this type of algorithm provides the means by which the limitations of existing aggregate planning techniques can be overcome. Tavakkoli-Moghaddam and Safaei (2006) have presented a genetic algorithm (GA) for solving a generalized model of single-item resource constrained APP with linear cost functions. Ramezanian, Rahmani, and Barzinpour (2012) developed a mixed integer linear programming (MILP) model for general two-phase aggregate production planning systems and presented genetic algorithm and tabu search for solving the NP-hard class of APP. The result found that the proposed algorithms obtained high-quality solutions for APP in a reasonable amount of computational time. Nowak (2013) designed the interactive procedure for aggregate production planning. It was assumed that cost minimization is the most important objective. Finally, an interactive procedure is used to identify the final solution to the problem. Ahmed, Biswas, and Nundy (2019) proposed a model that makes an effort to include in the optimization model all relevant cost factors that are affected by the APP, directly or indirectly.

Charles Darwin's theory of natural selection served as the foundation for John Holland's 1960s and 1970s model or abstraction of biological evolution, known as the Genetic Algorithm. (Yang, 2014). GA is a search algorithm based on the mechanisms of natural selection and natural genetics. They combine the survival of the fittest in string structures with the exchange of structured and random data to create a search algorithm with innovative human search intelligence (Goldberg, 1989). Umbarkar and Sheth (2015) described crossover operators that help researchers in selecting an appropriate crossover operator for better results. Hassanat et al. (2019) reviewed various methods for choosing mutation and crossover ratios in GAs and defined new deterministic control approaches for crossover and mutation rate, namely Dynamic Decreasing of high mutation ratio/dynamic increasing of low crossover ratio (DHM/ILC), and Dynamic Increasing of Low Mutation/Dynamic Decreasing of High Crossover (ILM/DHC). The experiments showed both proposed dynamic methods outperformed the predefined methods in most cases tested. Podvalny, Chizhov, Gusev, and Gusev (2019) proposed the crossover operator of a genetic algorithm that can be applied to tasks of production planning. The proposed crossover operators are presented in a formal way. Based on this operator, the optimizing software was developed. The proposed operator successfully passed the tests, and it will be used for solving real tasks in the future. Katoch, Chauhan, and Kumar (2021) discussed recent advances in GAs that are helpful for research and graduate teaching.

Many researchers have developed an integrated approach to solve APP problems and present many models that combine different algorithms and techniques to solve the problem. In this study, the authors used GA to solve the APP problem. Dakka, Aswin, and Siswojo (2017) presented a GA approach for solving APP with different selection methods and crossover procedures. They combined three selection methods and three crossover procedures. The results reveal the best performance was obtained by combining the rank selection procedure with scattered crossover. Mahmud, Hossain, and Hossain (2018) developed an interactive possibilistic environment-based GA for multi-product and multi-period APP and it has been solved by the MOGA to minimize the production cost and the rate of changing imprecise parameters. Yuliastuti, Rizki, Mahmudy, and Tama (2019) used a hybrid approach that combined GA and Simulated Annealing. The function of Simulated Annealing is to improve every solution produced by GA. The proposed hybrid method has been proven to provide better solutions. Liu and Yang (2021) purposed to deal with the multi-product APP, problem considering stability in the workforce and total production costs, and were established to minimize total production costs and instability in the work force. The result showed the NSGA-II algorithm based on local search has better performance in the multi-objective APP problem.

Some recent studies have attempted to find better methods for solving VRPTW, such as the exact method, the heuristics method, and meta heuristics method. For instance, Shi & Weise (2013) proposed ant colony optimization (ACO) to decrease the vehicle number and distance traveled to serve customers. Kosolsombat & Ratanavilisagul (2022) present a novel ACO-based optimization method for VRPTW using customer selection to decrease or solve the customer selection inefficiency of the ACO process and propose a reinitialization technique to decrease or optimize local trapping. Gambardella (2000) described a multiple-ant colony system for vehicle routing problems with time windows to minimize vehicle numbers and traveled distances. Revanna & Al-

Nakash (2022) defined the objective to decrease the number of vehicles employed, the cost of total logistics, and to reduce carbon emissions by approaching ACO with K-mean clustering. Amini (2011) proposed a particle swarm optimization algorithm (PSO) to solve VRPTW with Solomon benchmark problem instances and used it in a real-case study of a Chlorine Capsule distribution company at the water reservoir in Tehran. Duan et al. (2022) designed a new model disturbance of travel time and built a robust multi-objective VRPTW in which the travel time perturbation range is determined by the maximum perturbation level with two conflicting objectives, including minimizing both total distance and vehicle number. Mohammadi & Mahmoodian (2022) studied to select the best routes for a specific vehicle number to reduce fuel cost, driver wages, driver distance, and time needed to provide the products to the customers by using simulated annealing algorithm (SA).

Moreover, Dixit et al. (2019) reviewed some of the recent advances in VRPTW using meta-heuristic techniques. GA has been a popular algorithm for VRPTW problems. May et al. (2021) proposed an improved GA by developing a problem-specific crossover with seven different mutation operators to solve VRPTW. Ghani et al. (2016) studied GA and applied the random population method to assign the vehicle numbers to the routes connecting the customer and depot so that the overall distance traveled is minimized and the delivery operations are completed within the time windows requested by the customer. Kinoshita & Uchiya (2021) focused on a method to optimize while dynamically switching multiple crossovers based on the diversity of the gene population by using GA. Sripriya et al. (2015) proposed work that simultaneously minimizes the number of vehicles and total distance traveled, which achieves several objectives, and proposed a new hybrid genetic search with diversity control for GA to solve a large class of VRPTW. Anggodo et al. (2016) studied the application of multi trip VRPTW on the problems of the tourist routes in Banyuwangi by using GA. Gocken & Yaktubay (2019) proposed a multi objective GA approach for VRPTW solution and the effect on the initial population generation step of GA when using different clustering algorithms, including K-means, centroid-based heuristic, DBSCAN, and SNN clustering algorithms. Ibrahim et al. (2020) developed GA with crossover and mutation operators optimized for solving VRPPDTW cases. It is related to the constraints of vehicle capacity and the time windows of the destination node. Ibrahim et al. (2021) discussed the VRPPDTW model for reducing distance travel and penalties with an improved GA that was developed and applied to solve.

Based on several studies that have been done before, GA has been used by many researchers in solving the APP problem and VRPTW problem. Therefore, the author sees that there are good opportunities for future contributions. Here, the model of APP problems contains a large size of problem, including regular time cost, overtime cost, backordering cost, and inventory cost. The model of VRPTW problems use Solomon benchmark problem instances.

CHAPTER 3
RESEARCH METHODOLOGY

In this chapter, the author designed method and introduced a novel interactive crossover approach, considering four crossover options as well as creating a new crossover option, are described for APP problems and considered GA with a new crossover to solve by developing the problem with K-mean clustering to perform better for VRPTW. The research design can be developed based on the research objectives shown in Figure 15 Process Flow Chart.

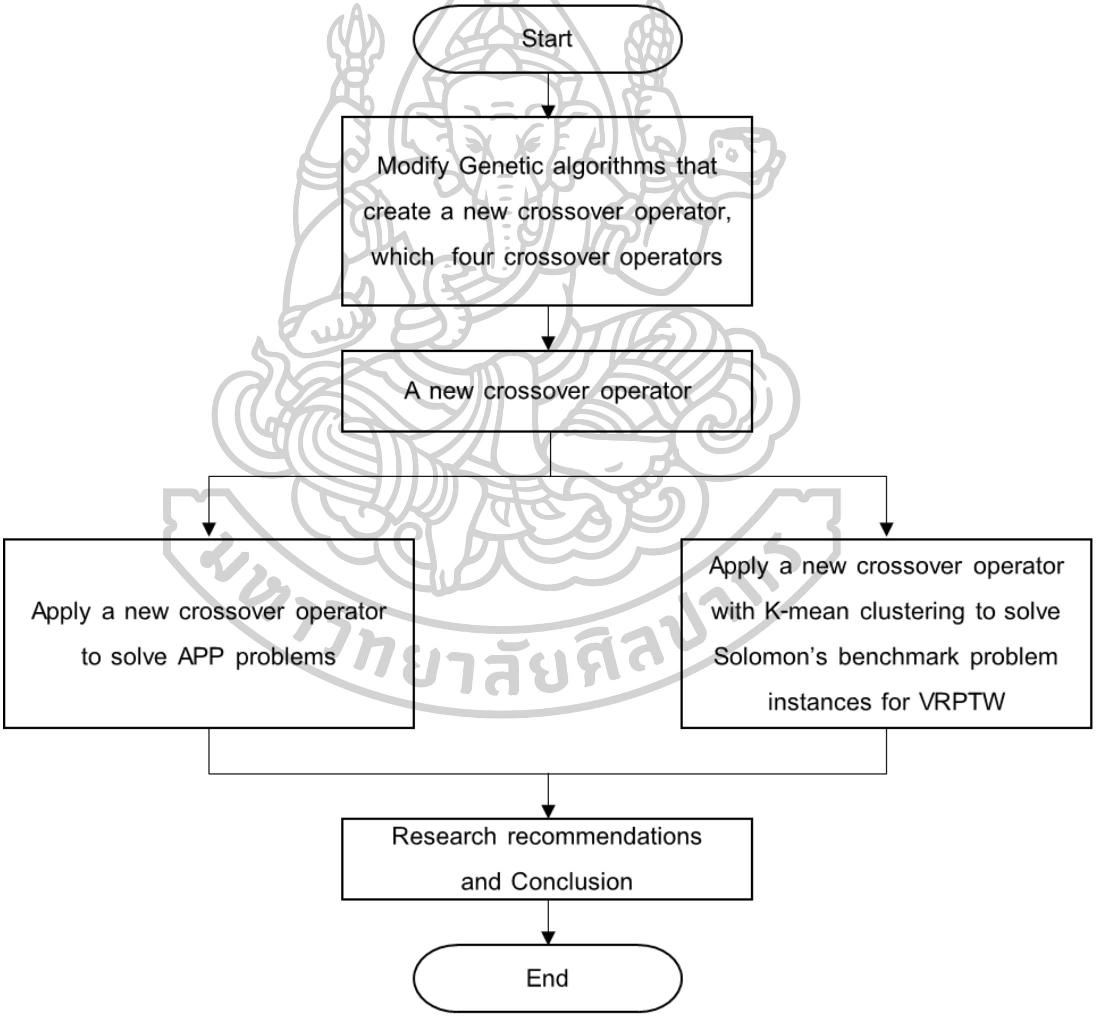

Figure 15 Process Flow Chart

3.1 Create a new crossover operator

In genetic algorithm, crossover is a recombination operator, combined the genetic information of two parent solutions to create new offspring. The concept is to taking two parents which randomly from the mating pool to crossover to produce a better offspring.

In this study, to create a new crossover used Single point crossover, Two points crossover, Arithmetic crossover, and Scatter crossover. It is a combination of four crossover operators. After that, placed each operator on the roulette wheel which the equal size of the probability area. Moreover, it can be adjusted according to the size of the probability area in each operator. It can be described in Figure 15.

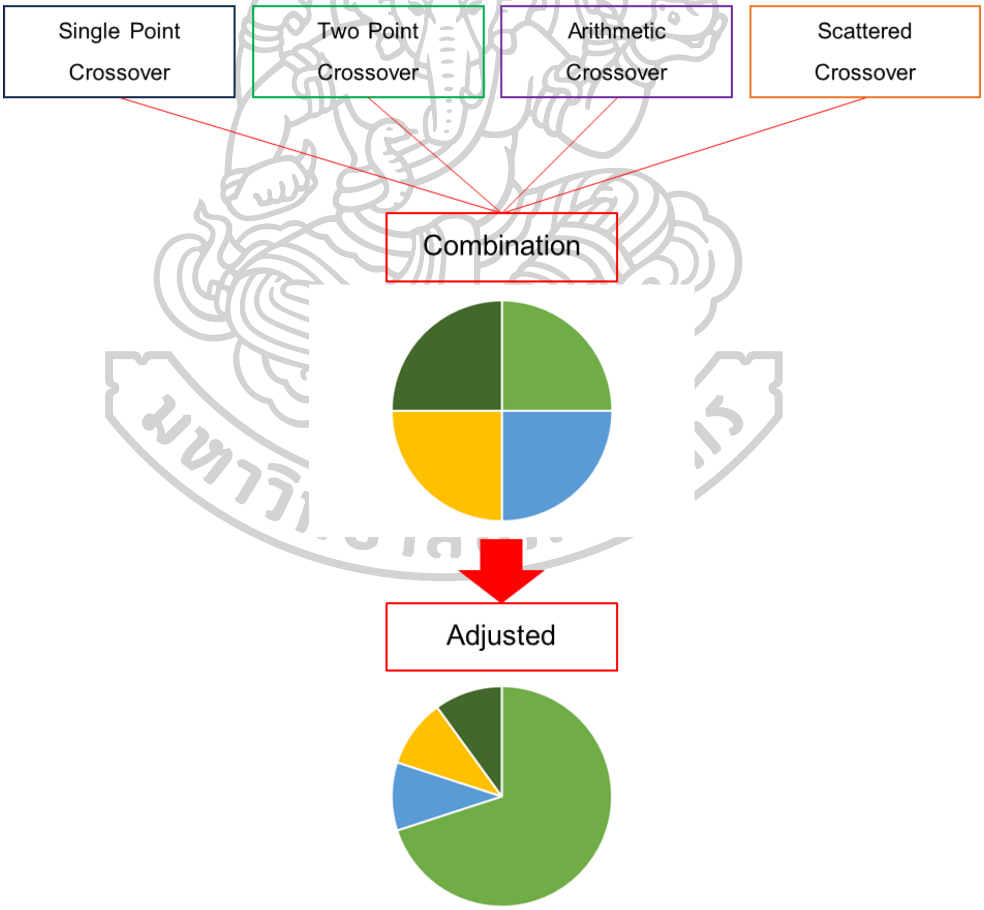

Figure 16 The process of creating a new crossover

3.2 Genetic algorithm for APP problems and Solomon’s benchmark problem instances for VRPTW.

The term Genetic Algorithm (GA) describes a set of methods, that can be used to optimize complex problems. As the name suggests, the processes employed by GAs are inspired by natural selection and genetic variation. To achieve this, a GA uses a population of possible solutions to a problem and applies a series of processes.

In keeping with the evolutionary theme, each individual in a GA population is represented by a chromosome. As in nature, this chromosome contains genetic information relating to each individual’s characteristics, it’s described in Figure 17.

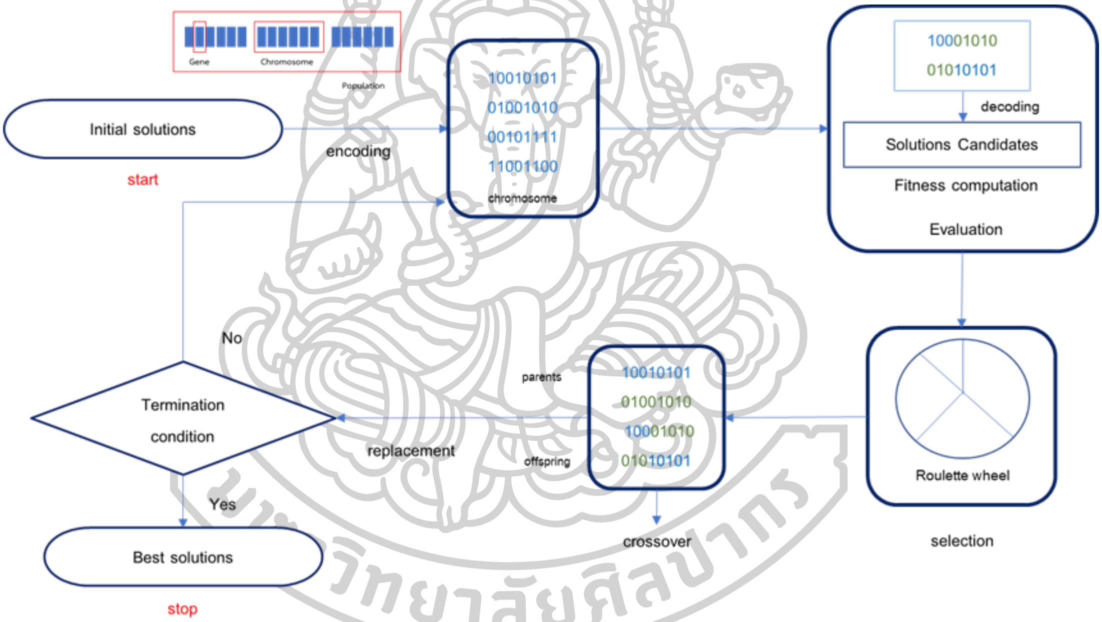

Figure 17 Genetic Algorithm Process

Based on several studies that have been done before, many researchers have been used GA to solve the APP problems and Solomon’s benchmark problem instances for VRPTW. Therefore, in this study applied four crossover and new crossover to solve which focused on the model of APP problems contains a large size of problem, including regular time cost, overtime cost, backordering cost, and inventory cost. The model of VRPTW problems use Solomon benchmark problem instances.

3.2.1 Genetic algorithm for APP problems

In this study, APP problem with multi-product and multi-period was used to test the algorithm. The author had studied the APP problems, which is the point to making use of the limited resources and answering customer satisfaction. The resources include the tools of manufacture, such as machines and tools, labor, and raw materials. APP is the plan to connect the production ability with the product needed in that section of time. It also needs good support, such as the appropriate appointing process for the best cost of investment and lowest cost of production possible.

3.2.1.2 Applied Transportation Problems to Aggregate Production Planning Problems

APP problems can be a form of transportation problems. The transportation model is one example of a linear programming model. It is the problem used to make the decision to choose the right material from origin to destination. This research will use the transportation problem to solve the APP problem by finding the most appropriate product for production planning.

Applying the APP problems to the transportation problem must be analyzed. First, if the problem is an unbalanced transportation problem, it must be balanced as a balanced transportation problem before finding an initial basic, feasible solution to an optimal solution in Table 3. Otherwise, total supply is equal to total demand.

Table 3 Balance transportation problem

	product1	product1	product1	product1	product1	product1	Capacity
	period1	period2	period3	period4	period5	period6	
RT1	10	12	14	16	10	12	80
RT2	12	14	16	18	12	14	80
RT3	12	10	12	14	16	10	80
RT4	14	12	14	16	18	12	80
RT5	14	12	10	12	14	16	80
RT6	16	14	12	14	16	18	80
Demand	80	90	60	70	100	80	480

Therefore, we need to learn how to make a problem balanced if it is not as in Table 4 and it was meant to cover two cases.

Table 4 Unbalance transportation problem

	product1	product1	product1	product1	product1	product1	Capacity	
	period1	period2	period3	period4	period5	period6		
RT1	10	12	14	16	10	12	80	
RT2	12	14	16	18	12	14	80	
RT3	12	10	12	14	16	10	80	
RT4	14	12	14	16	18	12	80	
RT5	14	12	10	12	14	16	80	
RT6	16	14	12	14	16	18	80	
Demand	80	75	60	70	60	55	→ ↓	400
							480	

For the first case, total supply exceeds total demand. They are balancing a transportation problem by creating a dummy demand that has a demand equal to the amount of excess supply. The dummy demand points are not real shipments, they are assigned zero cost. Table 5 shows a dummy demand.

Table 5 Balance transportation problem with dummy demand

	product1 period1	product1 period2	product1 period3	product1 period4	product1 period5	product1 period6	Dummy	Capacity
RT1	10	12	14	16	10	12	100	80
RT2	12	14	16	18	12	14	100	80
RT3	12	10	12	14	16	10	100	80
RT4	14	12	14	16	18	12	100	80
RT5	14	12	10	12	14	16	100	80
RT6	16	14	12	14	16	18	100	80
Demand	80	75	60	70	60	55	80	480

For the second case total supply is less than total demand. If a transportation problem has a total supply that is less than its total demand, then the problem has no feasible solution. Sometimes, it is desirable to allow the possibility of leaving some demand unmet. In such a situation, a penalty is often associated with unmet demand.

3.2.1.2 Chromosome Encoding

Chromosome Encoding is the first step in solving the problem. Encoding is one of the most important processes as it determines the effectiveness and efficiency of the genetic algorithm on a particular problem.

3.2.1.3 Chromosome Representation

In this study, the production cost of a product for each period is determined by the decision variables, which are positive integers, which contribute to the implementation of chromosome encoding. In a series of chromosomes, Table 6 shows a simple sample of chromosome with I as the products type and T as periods. PP_{it} denotes the chromosome of a production cost of product in term of periods.

Table 6 A simple example of chromosome

	Period (t) / Product (i)	1	2	...	T
	1	PP_{11}	PP_{12}	...	PP_{1T}
P	2	PP_{21}	PP_{22}	...	PP_{2T}
	...	...			
	I	PP_{I1}	PP_{I2}	...	PP_{IT}

	Period (t) / Product (i)	1	2	3	4	...	T
	1	10	12	14	16	...	16
P	2	12	10	12	14	...	14
	...	...					
	I	20	18	16	14	...	14

A gene in chromosome representation for the production cost of product i in period t as described in Figure 18.

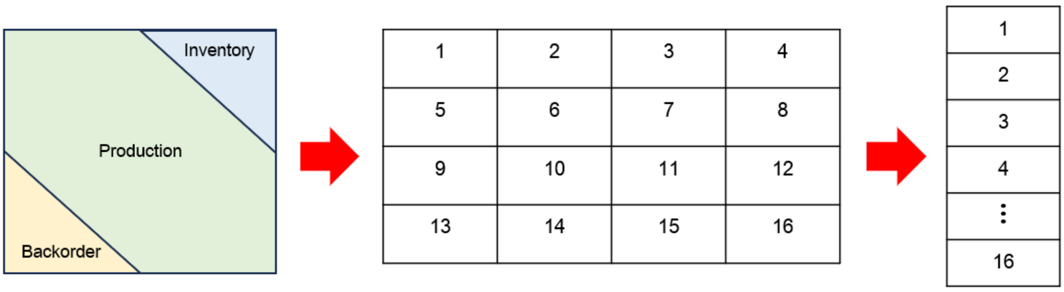

Figure 18 Chromosome representation in APP problems

3.2.1.4 Fitness Function

The fitness value is used to check the quality and feasibility of the solution. The higher the fitness value, the better the solution will be. The objective function is usually used as a measure of the survival probability of a chromosome. It can be obtained from the equation.

$$\text{Fitness}[i] = 1 / \text{Objective}[i]$$

3.2.1.5 Selection

Selection is the procedure of selecting individuals with the best fitness, while others are discarded because they will be used to produce offspring. The best fittest chromosomes have a higher chance of being selected for the next generation. Here the author used only the Roulette Wheel selection. Roulette-wheel selection associates each individual with a probability depending on its function value. In the first step, the selection probability of each chromosome is calculated. Selection probability the Prob [i] of a chromosome is calculated by:

Prob [i] = Fitness [i] / Cummulative_Fitness [i]

After calculating selection probability, divide a circular wheel into x segment, and the width of each segment is relative to selection probability. In the last step, a random number is generated and selected from a chromosome in which the random number.

3.2.1.6 Crossover

Crossover is used to create new offspring from individuals selected in the selection step. The crossover options used Single point crossover, Two points crossover, Arithmetic crossover, Scatter crossover and new crossover.

3.2.1.7 Problem Description for APP problems

In this study, the authors model and analyze the APP problems with multi-product multi-period. The orders consist of the type of product, quantity, and duration. Every specified period of time, decision-makers will create an initial APP. In the planning process, it has to consider the manufacturing capacity, invento ry level, and other factors to fulfill forecasted demand. According to the production plan, the factories are assigned a list of products with quantities to be produced at each time interval.

The model of APP planning problem with multi-product multi-period has the production cost, inventory cost, and back-order cost for different products, which means the actual value of the answer is ignored as shown in Table 7.

Table 7 Production, inventory and backorder cost

Product i	Cost of RT (bath/unit)	Cost of OT (bath/unit)	Inventory cost (bath/unit)	Backorder cost (bath/unit)
1	10	12	2	2
2	12	14	2	3
3	14	16	2	4
4	10	12	2	2
5	12	14	2	3

The forecast demand for each size of problem is shown in Table 8. The model has 6 period, 12 period, and 24 periods planning horizon. Following that, P denotes the period and S denotes the source. Therefore, 2P12P6S means 2 products, 12 periods, and 6 sources.

Table 8 Forecast Demand

Size of Problem	Product i	Period i											
		1	2	3	4	5	6	7	8	9	10	11	12
2P12P6S	Product 1	45	40	60	25	45	55	30	40	35	35	40	30
	Product 2	55	40	55	30	30	35	45	30	55	30	40	35
2P12P12S	Product 1	45	40	60	25	45	55	30	40	35	35	40	30
	Product 2	55	40	55	30	30	35	45	30	55	30	40	35
2P12P24S	Product 1	70	85	85	60	75	80	65	75	65	85	80	80
	Product 2	75	60	90	70	90	75	70	75	60	70	85	75
3P12P6S	Product 1	35	25	35	30	40	30	35	30	35	35	30	40
	Product 2	30	25	30	35	40	35	25	30	25	30	40	35
	Product 3	30	35	45	30	25	30	35	40	45	40	35	30
3P12P12S	Product 1	60	65	45	60	35	65	45	60	55	50	45	35
	Product 2	40	35	50	55	45	50	65	35	60	35	45	55
	Product 3	35	55	35	60	60	65	55	35	55	65	35	55

Table 8 Forecast Demand (Continue)

3P12P24S	Product 1	50	65	50	45	55	40	55	45	50	55	45	50
	Product 2	65	40	50	65	50	50	55	45	40	40	65	40
	Product 3	50	40	45	45	40	45	60	65	60	45	45	50
4P12P6S	Product 1	20	25	15	30	25	25	30	15	15	25	20	25
	Product 2	30	15	30	25	15	30	25	25	30	20	15	30
	Product 3	20	20	30	15	15	20	25	15	30	30	15	25
	Product 4	25	15	25	20	15	25	15	30	20	30	25	15
4P12P12S	Product 1	30	20	35	30	35	25	30	30	40	35	25	30
	Product 2	25	35	35	40	35	25	25	40	45	40	25	40
	Product 3	25	35	20	25	30	25	40	25	35	20	30	25
	Product 4	40	25	25	25	40	25	30	25	20	25	25	25
4P12P24S	Product 1	35	45	40	30	45	35	40	35	35	30	45	40
	Product 2	45	40	45	30	30	45	30	40	40	35	30	40
	Product 3	30	35	45	40	45	35	35	40	45	50	30	35
	Product 4	40	35	30	30	40	45	30	45	40	35	30	30
5P12P6S	Product 1	20	25	15	30	25	25	30	15	15	25	20	25
	Product 2	30	15	30	25	15	30	25	25	30	20	15	30
	Product 3	20	20	30	15	15	20	25	15	30	30	15	25
	Product 4	25	15	25	20	15	25	15	30	20	30	25	15
	Product 5	25	20	25	20	15	25	15	20	20	15	20	20
5P12P12S	Product 1	30	20	35	30	35	25	30	30	40	35	25	30
	Product 2	25	35	35	40	35	25	25	40	45	40	25	40
	Product 3	25	35	20	25	30	25	40	25	35	20	30	25
	Product 4	40	25	25	25	40	25	30	25	20	25	25	25
	Product 5	25	30	30	20	25	30	30	25	40	40	30	35
5P12P24S	Product 1	35	45	40	30	45	35	40	35	35	30	45	40
	Product 2	45	40	45	30	30	45	30	40	40	35	30	40
	Product 3	30	35	45	40	45	35	35	40	45	50	30	35
	Product 4	40	35	30	30	40	45	30	45	40	35	30	30
	Product 5	30	35	25	30	25	25	25	35	40	35	35	20

The sources of production capacities in the planning horizon are including regular time and overtime as shown in Table 9. For 24 sources of production capacities, it has regular time and over time because this study aims to show the large size of the problem.

Table 9 Production capacity

Source	Capacity per Period							
	2 Products		3 Products		4 Products		5 Products	
6 Sources	160		200		180		220	
12 Sources	80		150		120		150	
24 Sources	RT	OT	RT	OT	RT	OT	RT	OT
	100	50	100	50	100	50	120	60

For APP problems, the aim of this research is to check the effect of crossover options and create new crossover options on multi-product multi-period APP problems. Various combinations of crossovers are tested for APP problems. Results are obtained for 10 iterations and compared according to different statistical values with a line chart. The population size is 1500, 1000, 600, 500, 150, and 100, respectively. The number of generations is 20, 30, 50, 60, 200, and 300, respectively, and the number of runs which have been considered for the experimental run is 30,000.

3.2.2 Genetic algorithm for Solomon's benchmark problem instances for VRPTW.

Many researchers have been studied of using GA to solve VRPTW. However, in the research used the new crossover in GA to solve Solomon's benchmark problem instances for VRPTW with apply K-mean clustering consists of three phases; initial population and chromosome representation, selection, and crossover. The detail of three phases is described as follows.

3.2.2.1 K-mean clustering algorithm

K-mean clustering is a method for dividing all data into k clusters. The centroid of each cluster is random which determines a number of clusters. The objects were grouped to minimize the sum of the squared distances between them and their assigned cluster mean. The steps of K-means algorithm are explained as follows: determine K random point as initial cluster centres, define each point to the nearest centroid of cluster. Then examine the possible of capacity constraint, repeat calculate the centroid of each cluster finally, repeat until stop when maximum number of iterations.

In this study, the author defined the number of clusters in the same for set R, set C, and set RC with type 1 and type 2. The number of clusters is 10 groups.

3.2.2.2 Initial population and chromosome representation

Solomon's benchmark problem instances for VRPTW can be applied to TSP model as described in Figure 19.

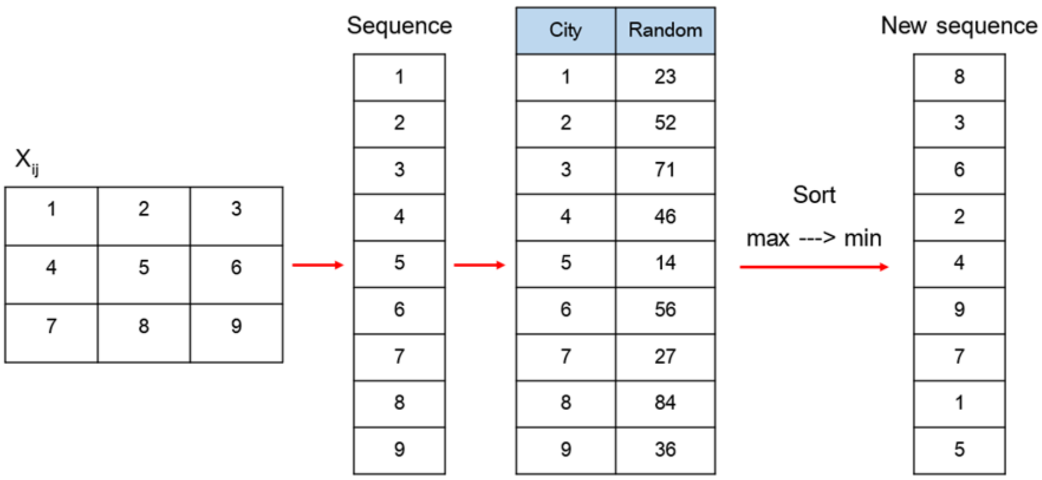

Figure 19 TSP model for Solomon's benchmark problem instances for VRPTW

The first step of GA is initialization of population. The initial population is generated randomly and the size of initial population used is 2,000. In this research, chromosome representation is represented by a series of number that shown in Figure 20. This gene keeps information about which vehicles traveled to which customer and in which order.

Figure 20 Chromosome representation in Solomon's benchmark problem instance for VRPTW

3.2.2.3 Selection

Roulette wheel is used. The workflows of the roulette wheel selection are divided the circle wheel, where the probability for selection an individual is depend on fitness value. Therefore, the larger fitness of individual has more chance to selection when the wheel is rotated.

3.2.2.4 Crossover

Crossover operator is a combination of two individuals and creates a new offspring. To reproduce the good combinations, there should normally be a high chance of a crossover operation occurring. Single point crossover, Two point crossover, Arithmetic crossover, and Scattered crossover are used to create a new offspring in this research. In addition to that, there is also a Stas crossover.

3.2.2.5 Problem Description for Solomon benchmark Instances for VRPTW

In this study, the authors model and analyze the Solomon benchmark Instances for VRPTW with TSP model.

For VRPTW, Solomon benchmark Instances is a set of well-known benchmark problems. Solomon benchmark instances are divided into six sets, which include C1, C2, R1, R2, RC1, and RC2. Each set contains between eight and twelve instances. In this study chooses 100 customers, and the Table 11 shows Solomon benchmark Instances. Various combinations of crossovers with and without K-mean clustering are tested for Solomon's benchmark problem instances.

Table 10 Solomon benchmark Instances.

Geographical data	Problem sets	Instance
Randomly generated	R101-R112 R201-R211	23 problems
Clustered structures	C101-C109 C201-C208	23 problems
A mix of random and clustered structures	RC101-RC108 RC201-RC208	16 problems

Results are obtained for 10 run iterations the average value is to be reported for each combination. The population size is 2,000, the maximum generation is 350, the number of clusters is 10 groups, and the number of runs which have been considered for the experimental run is 2,000 iterations.

CHAPTER 4

A NOVEL CROSSOVER OPERATOR FOR GENETIC ALGORITHM

In this chapter, the results were divided into three parts, including a new crossover operator, Stas Crossover for Aggregate Production Planning Problem, and Stas Crossover with K-mean Clustering for Vehicle Routing Problem with Time Window. A novel crossover operator for genetic algorithm is called "Stas Crossover". It is a combination of four crossover operators, including Single point crossover, Two points crossover, Arithmetic crossover, and Scattered crossover. It then presents the performance of this crossover operator, and finally tests multi-product and multi-period APP problems with an interactive decision variable and chooses approximate crossover options. Moreover, the research used the new crossover to solve VRPTW by developing the problem with K-mean clustering. After that, we compared Stas crossover with four crossover operators and adjusted the area of probability for Stas crossover. It has been tested on Solomon benchmark instances for VRPTW with six problem sets of different problem types.

4.1 A new crossover operator

The proposed crossover operator in this study is a combination of four crossover operators, including Single point crossover, Two points crossover, Arithmetic crossover, and Scattered crossover. Crossover operators are used to combine the genetic information of two parents to create a new offspring. The probability that each operator is the same in creating a new offspring is equal to 25%. The new crossover operator is called "Stas Crossover". The following Figure 21 illustrates the process of Stas Crossover, and shows that the new offspring will have the same probability of occurring.

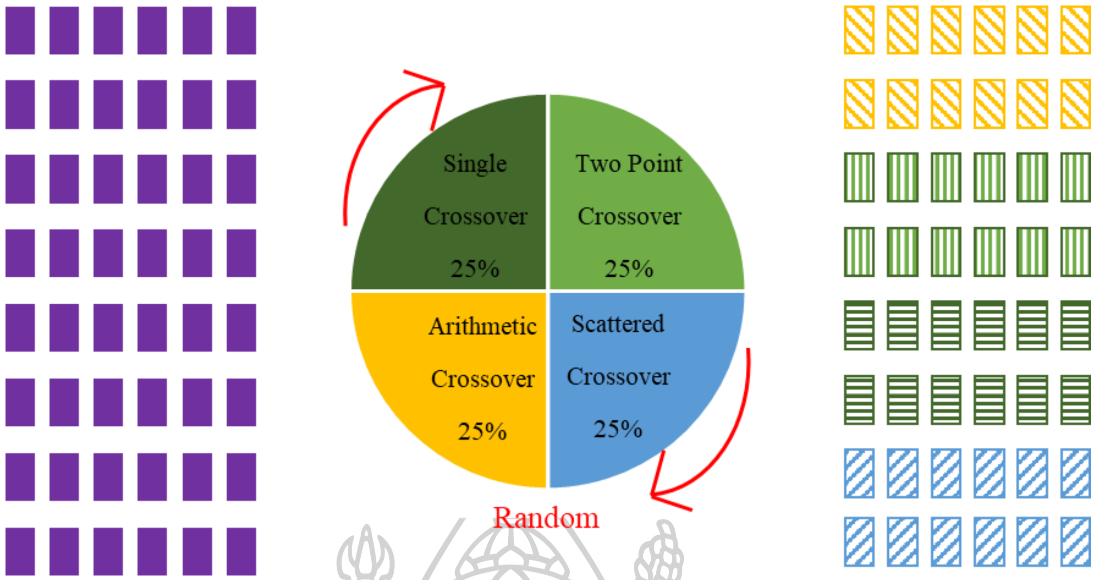

Figure 21 Illustration of Stas crossover process

Stas crossover can be adjusted according to the size of the probability area in each operator. The probability of creating a new offspring for each operator is not the same. This results in more variety than traditional crossover. To adjust the size of the probability area, the number after Stas represents the probability of each operator. There is also an adjusted probability of Single point crossover of 70%, Two points crossover of 10%, Arithmetic crossover of 10%, and Scattered crossover of 10%. It is called Stas7111 crossover. Figure 22 illustrates Stas7111 crossover process, and shows that the offspring have not equal probability of occurring. The chance of occurrence depends on the size of the probability area.

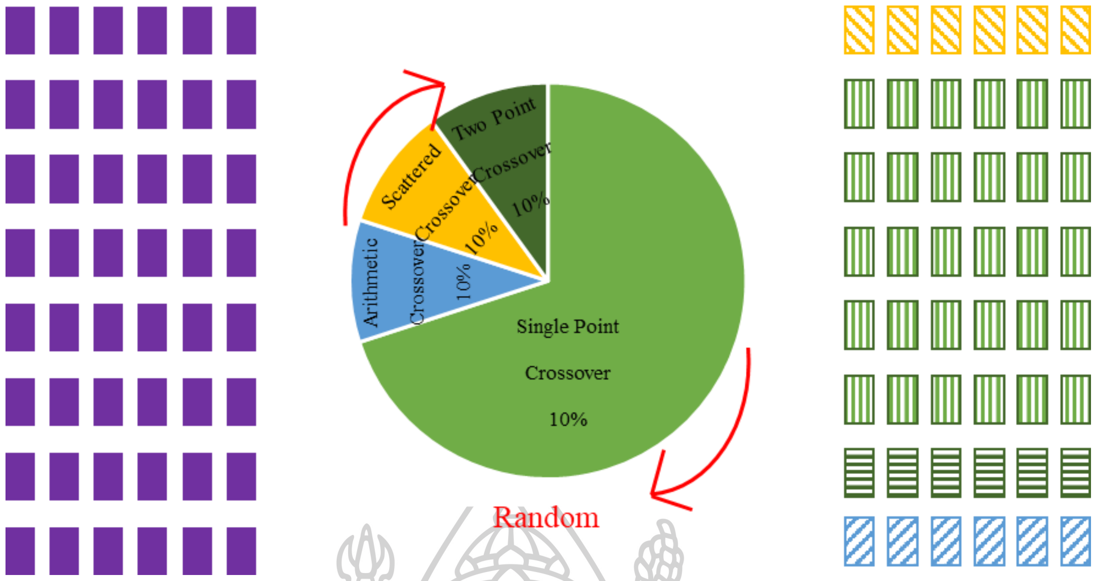

Figure 22 Illustration of Stas7111 crossover process

Stas crossover operator chooses two parents, such as p1 and p2. In order to arrange all of the crossover operators on a roulette wheel with an equal area probability, the roulette is turned, and in order to determine which way to create offspring, the crossover operators are chosen at random and include Single point crossover, Two points crossover, Arithmetic crossover, and Scattered crossover. The relevant crossover options are selected to create the new offspring according to the following conditions: Each experiment is randomized, which means that if 10 runs are given, a different crossover operator will be chosen on each run. For example, if the first experiment run is selected as a Single point crossover, that means the second experiment run may be selected as a Single point crossover again or select different crossover operators.

4.2 Stas Crossover for Aggregate Production Planning Problems

In this study, the algorithm was tested by using the APP problems with multi-product and multi-period. The problems are generated with various condition parameters that determine the size of the problem. The crossover operators have 7 types: include Single point crossover, Two point crossover, Arithmetic crossover, Scattered crossover, Stas crossover, Stas1117 crossover, and Stas0055 crossover. This study attended on the large size of the problem and focused on the optimal solution by changing production costs, backorder costs, production capacity, and forecast demand, which means the actual value of the answer is ignored.

The problems are defined based on the size of the problems, in which 40 different crossover options are tested for 12 types of problems and 10 iterations. Results are obtained for 10 run iterations and compared according to different statistical values with a line chart. The scenarios for each product include 2 products, 3 products, 4 products, and 5 products as shown in Figure 23 – 26, respectively.

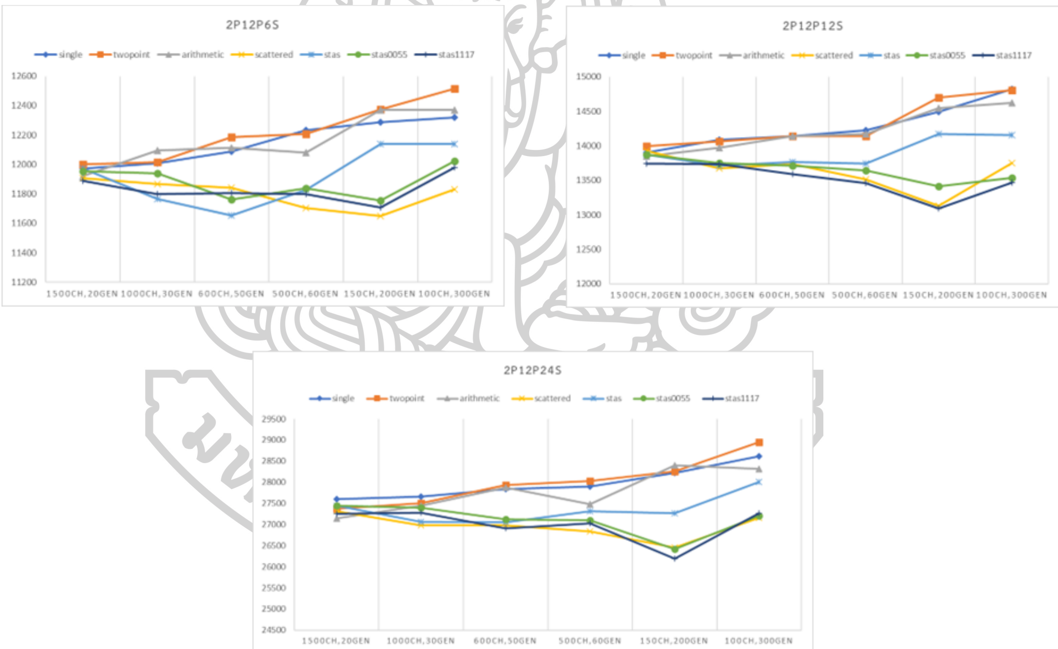

Figure 23 Scenario of 2 products

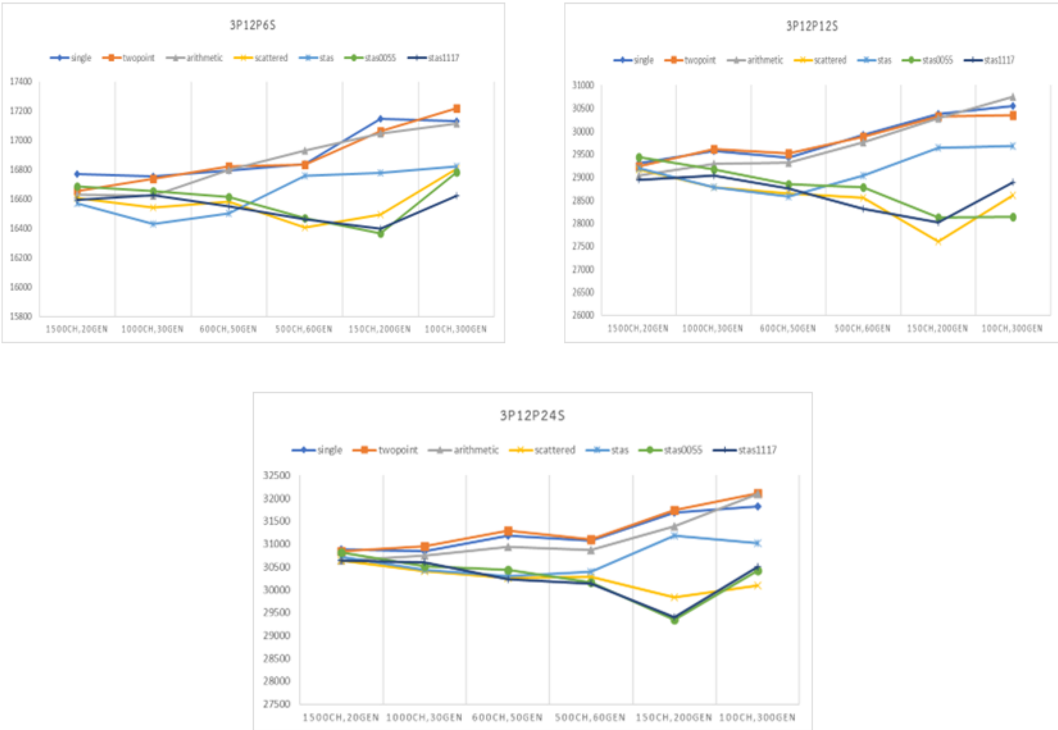

Figure 24 Scenario of 3 products

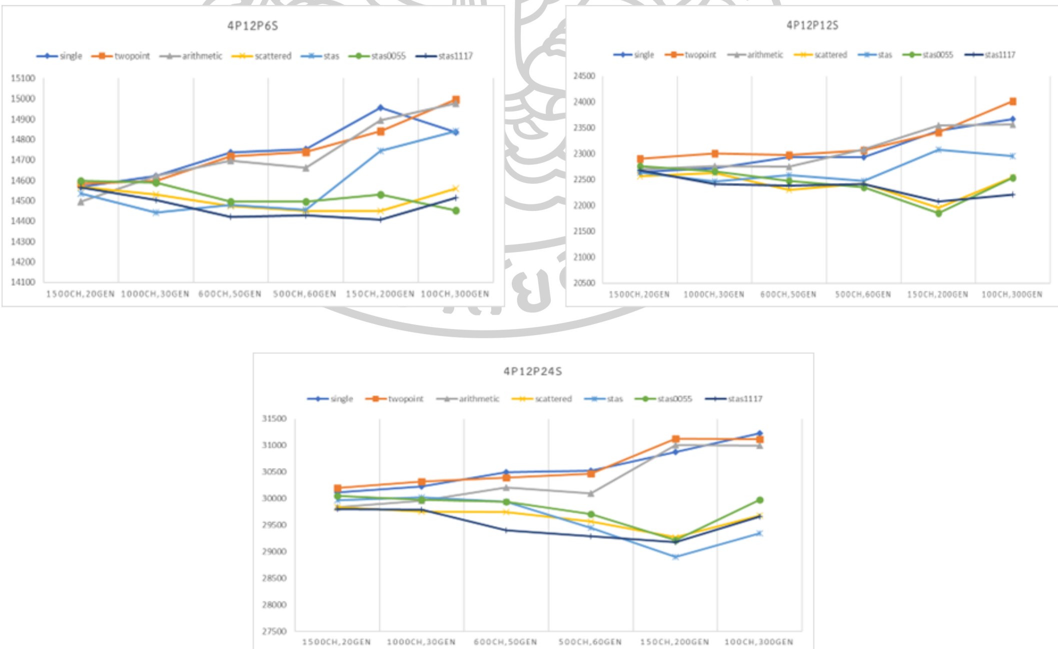

Figure 25 Scenario of 4 products

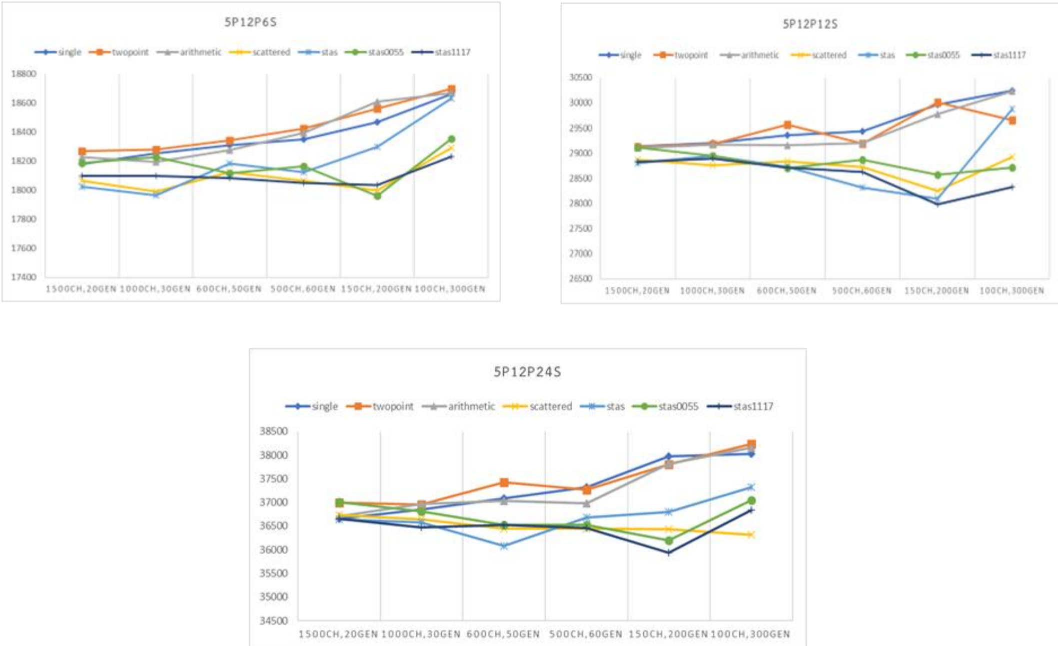

Figure 26 Scenario of 5 products

Furthermore, Figure 27 shows the proportion of the number needed to find the best performance for each crossover operator. The results found that, Stas1117 crossover was able to find the best performance in 28 out of 72 results, which means that by changing the area probability, it's better than other Stas crossover and other operators, too.

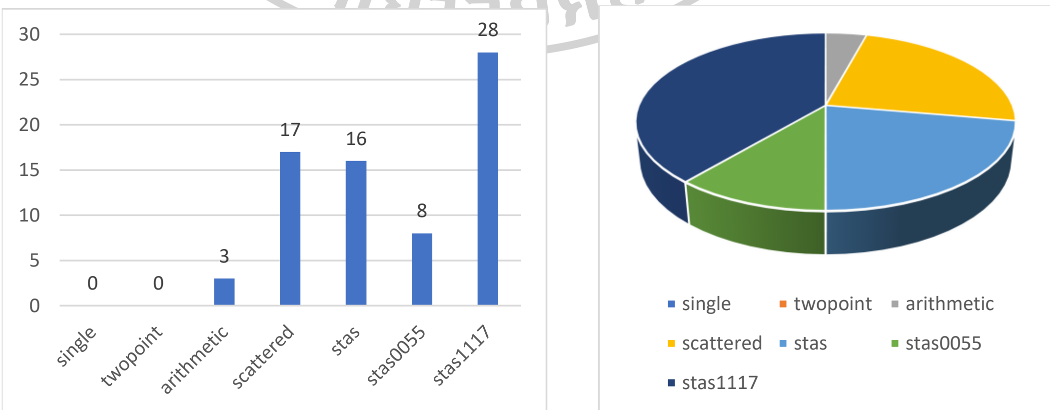

Figure 27 Proportion of the best performance for each crossover operators

Table 11 shows the cost value obtained using different combinations of crossover procedures. The results show that Stas0055 crossover and Stas1117 crossover perform better than Single point crossover, Two points crossover, and Arithmetic crossover. Moreover, the population size is 150 chromosomes and the number of generations is 200 generations, both of which are the best possible answers. It has achieved better performance than other crossovers for 9 results out of 12 results, which are shown with bold values according to the minimum cost value.

Table 11 Cost value obtained by using different combinations of crossover procedures

Size of Problem	Chromosome/ Generation	Crossover Operator	Cost value
2p12p6s	150ch,200gen	Scattered	11,652
2p12p12s	150ch,200gen	Stas1117	13,093
2p12p24s	150ch,200gen	Stas1117	26,194
3p12p6s	500ch,60gen	Stas0055	16,368
3p12p12s	150ch,200gen	Scattered	27,606
3p12p24s	150ch,200gen	Stas0055	29,347
4p12p6s	1000ch, 30gen	Stas1117	14,407
4p12p12s	150ch, 200gen	Stas0055	21,852
4p12p24s	150ch, 200gen	Stas	28,901
5p12p6s	1000ch, 30gen	Stas0055	17,960
5p12p12s	150ch, 200gen	Stas1117	27,983
5p12p24s	600ch, 50gen	Stas1117	35,940

4.3 Stas Crossover with K-mean Clustering for Vehicle Routing Problems with Time Windows

In this study, Stas crossover in GA was modified to solve VRPTW by developing the problem with K-mean clustering. The crossover operators have 8 types: include Single point crossover, Two point crossover, Arithmetic crossover, Scattered crossover, Stas crossover, Stas1117 crossover, Stas7111 crossover, and Stas0055 crossover. The standard Solomon's benchmark problem instances for VRPTW were used for the experiments, which are described as C1, R1, RC1, C2, R2, and RC2. Each set contains between eight to twelve 100-node problems. The population size is 2,000, the maximum generation is 350, the number of clusters is 10 groups, and the code has been run for up to 2,000 iterations. Results are obtained for 10 run iterations and reported as the minimum distance and average distance for each option. The scenario of six problem sets with the location and dispersion characteristics of the cluster as shown in Figure 28 - 30, respectively. The results found that K-mean clustering has better performance in routing than without K-mean clustering. As both type 1 and type 2 in set C can be seen obviously, the paths with K-mean clustering are arranged into groups and are orderly, but the paths without K-mean clustering are disordered. Furthermore, Set R and Set RC are the same for both types.

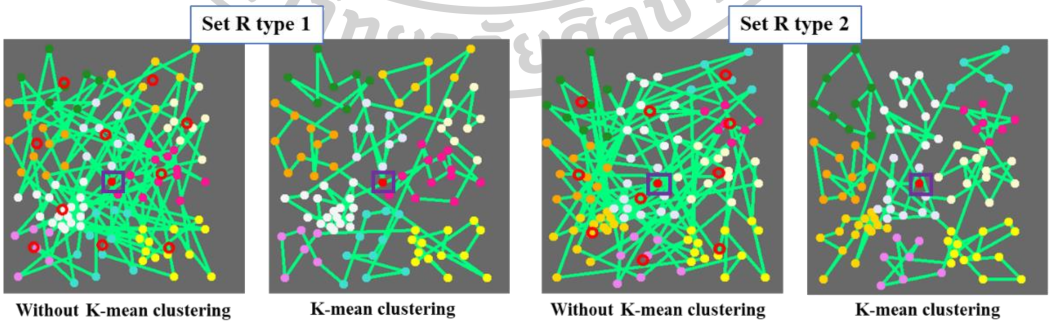

Figure 28 Scenario of Set R type 1 and type 2

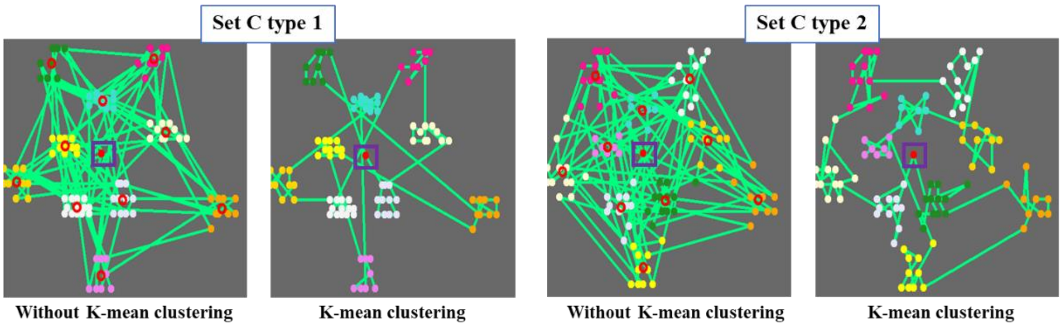

Figure 29 Scenario of Set C type 1 and type 2

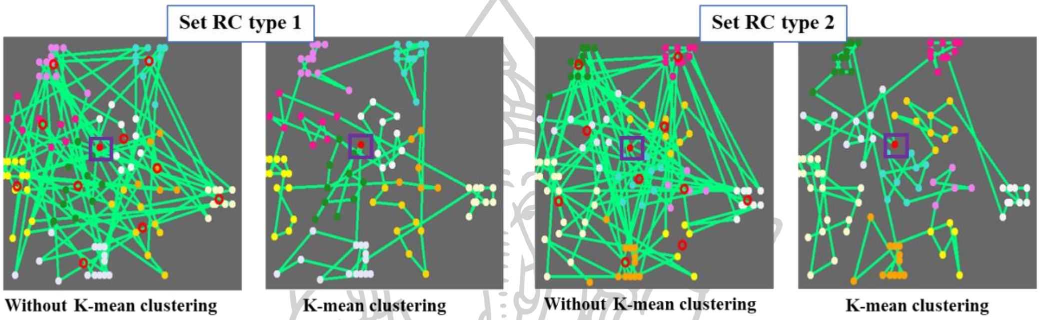

Figure 30 Scenario of Set RC type 1 and type 2

Table 12 compares the crossover operator performance on instances of the Solomon benchmark with Set R type 1 and type 2. Set R is generated randomly from the customer locations. The results show that in minimum distance without K-mean clustering, Single point crossover is recommended for Set R type 1, Single point crossover or Scatter crossover is recommended for Set R type 2. In minimum distance with K-mean clustering, Stas1117 crossover is recommended for Set R type 1, Stas5005 crossover is recommended for Set R type 2. In average distance without K-mean clustering, Scatter crossover is recommended for Set R type 1, Scatter crossover is recommended for Set R type 2. In average distance with K-mean clustering, Stas1117 crossover is recommended for Set R type 1, and Stas1117 crossover is recommended for Set R type 2.

Table 12 Comparison of the crossover operator performance on instances of the Solomon benchmark with Set R type 1 and type 2.

Instance	Minimum Distance				Average Distance			
	Without K-mean clustering		K-mean clustering		Without K-mean clustering		K-mean clustering	
R101	Stas1117	2661.54	Stas7111	1176.85	Scatter	2741.76	Stas7111	1203.86
R102	Stas5005	2624.53	Scatter	1207.91	Scatter	2737.38	Stas1117	1236.01
R103	Scatter	2631.42	Stas1117	1272.30	Single	2742.82	Stas7111	1297.40
R104	Single	2687.98	Stas5005	1160.19	Scatter	2740.94	Stas5005	1201.04
R105	Single	2626.82	Single	1255.75	Scatter	2712.62	Stas1117	1274.15
R106	Single	2638.98	Stas1117	1177.49	Stas5005	2721.67	Scatter	1194.37
R107	Single	2655.12	Scatter	1269.40	Single	2753.53	Single	1280.79
R108	Stas5005	2658.86	Stas1117	1149.87	Scatter	2736.72	Scatter	1186.05
R109	Scatter	2597.00	Stas1117	1160.15	stas1117	2711.99	Stas1117	1186.65
R110	Scatter	2636.11	Stas5005	1275.23	Stas7111	2716.47	Stas1117	1314.48
R111	Scatter	2695.78	Stas7111	1176.79	Single	2742.46	Scatter	1197.66
R112	Single	2690.56	Single	1204.94	Stas7111	2739.33	Single	1221.31
R201	Single	2569.49	Scatter	1203.32	Stas1117	2727.72	Stas5005	1225.15
R202	Scatter	2661.34	Single	1252.15	Single	2720.70	Scatter	1287.60
R203	Stas1117	2633.12	Stas5005	1289.26	Scatter	2716.15	Stas5005	1308.46
R204	Single	2678.35	Stas1117	1214.08	Scatter	2739.97	Single	1234.28
R205	Stas1117	2666.82	Stas5005	1138.55	Stas1117	2719.96	Stas1117	1161.81
R206	Scatter	2527.57	Scatter	1168.98	Stas1117	2721.75	Single	1202.86
R207	Single	2657.99	Stas	1221.14	Stas7111	2712.93	Stas	1241.62
R208	Scatter	2652.14	Single	1160.64	Scatter	2704.79	Scatter	1184.85
R209	Scatter	2670.66	Stas5005	1191.89	Scatter	2728.46	Stas1117	1226.74
R210	Stas1117	2666.41	Stas7111	1234.09	Single	2727.43	Stas7111	1258.13
R211	Single	2669.43	Stas1117	1129.32	Scatter	2727.98	Stas1117	1146.94

Table 13 compares the crossover operator performance on instances of the Solomon benchmark with Set C type 1 and type 2. Set C clusters all of the customer location coordinates. The results show that in minimum distance without K-mean clustering, Single point crossover is recommended for Set C type 1, Single point crossover or Scatter crossover is recommended for Set C type 2. In minimum distance with K-mean clustering, Stas7111 crossover is recommended for Set C type 1, Single point crossover is recommended for Set C type 2. In average distance without K-mean clustering, Scatter crossover is recommended for Set C type 1, Single point crossover, Scatter crossover or Stas1117 crossover is recommended for Set C type 2. In average distance with K-mean clustering, Stas7111 crossover is recommended for Set C type 1, and Scatter crossover is recommended for Set C type 2.

Table 13 Comparison of the crossover operator performance on instances of the Solomon benchmark with Set C type 1 and type 2.

Instance	Minimum Distance				Average Distance			
	Without K-mean clustering		K-mean clustering		Without K-mean clustering		K-mean clustering	
C101	Stas7111	3085.22	Stas7111	931.92	Scatter	3162.29	Stas7111	948.93
C102	Single	3009.74	Stas	946.25	Single	3143.85	Stas	957.96
C103	Single	3068.65	Stas7111	931.45	Scatter	3151.30	Stas7111	942.93
C104	Stas5005	3020.30	Single	903.38	Stas7111	3133.30	Scatter	919.49
C105	Single	2916.59	Stas1117	834.57	Stas7111	3126.38	Stas5005	854.71
C106	Single	3013.41	Stas7111	887.89	Scatter	3123.65	Stas7111	908.07
C107	Scatter	3049.58	Single	880.56	Single	3125.00	Scatter	894.64
C108	Single	3005.27	Stas	856.68	Scatter	3125.57	Stas	872.13
C109	Single	3036.13	Stas5005	917.99	Stas	3126.49	Stas5005	932.37
C201	Single	3030.71	Stas1117	1131.23	Scatter	3183.54	Stas1117	1151.99
C202	Stas1117	3039.12	Stas5005	1146.95	Stas1117	3166.01	Stas5005	1190.10
C203	Stas	3083.54	Single	1169.85	Stas7111	3194.94	Scatter	1202.78
C204	Single	3053.46	Stas7111	1149.25	Stas1117	3178.60	Scatter	1181.14
C205	Scatter	3118.02	Single	1117.09	Stas	3196.28	Scatter	1144.59

Table 13 Comparison of the crossover operator performance on instances of the Solomon benchmark with Set C type 1 and type 2. (Continue)

Instance	Minimum Distance				Average Distance			
	Without K-mean clustering		K-mean clustering		Without K-mean clustering		K-mean clustering	
C206	Stas7111	3083.63	Scatter	1189.92	Scatter	3191.40	Stas5005	1213.57
C207	Stas5005	3098.08	Single	1109.08	Single	3207.74	Stas5005	1136.67
C208	Scatter	3124.39	Stas	1172.29	Single	3201.71	Scatter	1208.24

Table 14 compares the crossover operator performance on instances of the Solomon benchmark with Set RC type 1 and type 2. Set RC is a mix of Set R and Set C. The results show that in minimum distance without K-mean clustering, Single point crossover is recommended for Set RC type 1, Scatter crossover, Stas crossover or Stas7111 crossover is recommended for Set RC type 2. In minimum distance with K-mean clustering, Stas7111 crossover is recommended for Set RC type 1, Stas1117 crossover is recommended for Set RC type 2. In average distance without K-mean clustering, Stas1117 crossover is recommended for Set RC type 1, Scatter crossover is recommended for Set RC type 2. In average distance with K-mean clustering, Stas7111 crossover is recommended for Set RC type 1, and Stas1117 crossover is recommended for Set RC type 2.

Table 14 Comparison of the crossover operator performance on instances of the Solomon benchmark with Set RC type 1 and type 2.

Instance	Minimum Distance				Average Distance			
	Without K-mean clustering		K-mean clustering		Without K-mean clustering		K-mean clustering	
RC101	Stas1117	3388.78	Single	1317.54	Scatter	3563.52	Stas7111	1347.91
RC102	Single	3404.87	Scatter	1301.06	Stas7111	3561.81	Scatter	1326.59
RC103	Scatter	3417.54	Stas7111	1231.68	Stas1117	3543.58	Stas7111	1257.56
RC104	Stas5005	3422.21	Stas7111	1312.75	Stas5005	3569.62	Stas7111	1353.25
RC105	Single	3364.39	Stas1117	1233.67	Stas1117	3543.06	Single	1281.75

Table 14 Comparison of the crossover operator performance on instances of the Solomon benchmark with Set RC type 1 and type 2. (Continue)

Instance	Minimum Distance				Average Distance			
	Without K-mean clustering		K-mean clustering		Without K-mean clustering		K-mean clustering	
RC106	Stas7111	3428.35	Stas5005	1290.10	Scatter	3576.80	Stas5005	1323.63
RC107	Single	3467.97	Single	1344.91	Stas1117	3544.65	Scatter	1374.80
RC108	Stas1117	3468.41	Stas7111	1204.56	Stas1117	3559.11	Single	1264.83
RC201	Stas1117	3464.40	Stas1117	1249.62	Single	3574.83	Stas1117	1272.89
RC202	Stas7111	3497.97	Stas	1195.54	Scatter	3590.15	Scatter	1234.26
RC203	Stas7111	3470.91	Single	1240.28	Stas7111	3582.81	Stas1117	1266.12
RC204	Single	3493.42	Stas1117	1196.61	Scatter	3548.85	Stas1117	1231.88
RC205	Stas	3423.66	Stas7111	1287.49	Scatter	3578.89	Stas1117	1339.55
RC206	Scatter	3364.38	Stas1117	1263.11	Stas1117	3566.50	Stas1117	1279.64
RC207	Scatter	3414.01	Scatter	1309.64	Scatter	3570.89	Single	1341.80
RC208	Stas	3436.73	Single	1234.96	Stas	3540.55	Single	1269.57

Table 15 compares the algorithm performance on the Solomon Benchmark instance for type 1 and type 2. Stas crossover applied with K-mean clustering is significantly improved as it allows more diversity to select how to create offspring and arrange orderly paths. Directly increases the opportunity of creating offspring with good genetic information. Some of the results performed better in comparison to the previous best-published studies. However, it is shown that the proposed algorithm outperforms Set R and Set RC in some instances, which means that K-means clustering affects Set R and Set RC due to the paths being arranged into orderly groups. Moreover, K-mean clustering has not affected Set C due to the location coordinates of all customers already clustered. It has been shown that adding K-mean clustering to the Stas crossover efficiently contributes to its performance. The bolder results show the best performance in minimizing the number of vehicles and the total distance traveled.

Table 15 Comparison of the algorithm performance on instances of the Solomon Benchmark for type 1 and type 2.

Instance	Best-know solution		Ref.	Proposed Stas crossover with K-mean Clustering	
	vehicles	distance		vehicles	distance
R101	**11**	**1125.00**	May et al. (2021)	10	1176.85
R102	**11**	**1128.00**	May et al. (2021)	10	1207.91
R103	**11**	**1212.00**	May et al. (2021)	10	1272.30
R104	**9**	**1007.31**	Mester et al. (2007)	10	1160.19
R105	11	1260.00	May et al. (2021)	**10**	**1255.75**
R106	12	1251.00	May et al. (2021)	**10**	**1177.49**
R107	**10**	**1104.66**	Shaw (1997)	10	1269.40
R108	**9**	**960.88**	Berger et al. (2001)	10	1149.87
R109	11	1194.73	Homberger & Gehring (1999)	**10**	**1160.15**
R110	**10**	**1104.00**	May et al. (2021)	10	1275.23
R111	**10**	**1096.72**	Rousseau et al. (2002)	10	1176.79
R112	**9**	**982.14**	Gambardella et al. (1999)	10	1204.94
C101	**10**	**828.94**	Rochat & Taillard (1995)	10	931.92
C102	**10**	**828.94**	Rochat & Taillard (1995)	10	946.25
C103	**10**	**828.06**	Rochat & Taillard (1995)	10	931.45
C104	**10**	**824.78**	Rochat & Taillard (1995)	10	903.38
C105	**10**	**828.94**	Rochat & Taillard (1995)	10	834.57
C106	**10**	**828.94**	Rochat & Taillard (1995)	10	887.89
C107	**10**	**828.94**	Rochat & Taillard (1995)	10	880.56
C108	**10**	**828.94**	Rochat & Taillard (1995)	10	856.68
C109	**10**	**828.94**	Rochat & Taillard (1995)	10	917.99
RC101	12	1474.00	May et al. (2021)	**10**	**1317.54**
RC102	11	1338.00	May et al. (2021)	**10**	**1301.06**
RC103	11	1250.00	May et al. (2021)	**10**	**1231.68**
RC104	**10**	**1135.48**	Cordeau et al. (2000)	10	1312.75
RC105	11	1274.00	May et al. (2021)	**10**	**1233.67**

Table 15 Comparison of the algorithm performance on instances of the Solomon Benchmark for type 1 and type 2. (Continue)

Instance	Best-know solution		Ref.	Proposed Stas crossover with K-mean Clustering	
	vehicles	distance		vehicles	distance
RC106	11	1270.00	May et al. (2021)	10	1290.10
RC107	11	1230.48	Shaw (1997)	10	1344.91
RC108	10	1139.82	Taillard et al. (1997)	10	1204.56
R201	2	791.00	May et al. (2021)	10	1203.32
R202	2	740.00	May et al. (2021)	10	1252.15
R203	2	738.00	May et al. (2021)	10	1289.26
R204	2	734.00	May et al. (2021)	10	1214.08
R205	2	726.00	May et al. (2021)	10	1138.55
R206	2	728.00	May et al. (2021)	10	1168.98
R207	2	742.00	May et al. (2021)	10	1221.14
R208	2	732.00	May et al. (2021)	10	1160.64
R209	2	733.00	May et al. (2021)	10	1191.89
R210	2	732.00	May et al. (2021)	10	1234.09
R211	2	751.00	May et al. (2021)	10	1129.32
C201	3	591.56	Rochat & Taillard (1995)	10	1131.23
C202	3	591.56	Rochat & Taillard (1995)	10	1146.95
C203	3	591.17	Rochat & Taillard (1995)	10	1169.85
C204	3	590.60	Rochat & Taillard (1995)	10	1149.25
C205	3	588.88	Rochat & Taillard (1995)	10	1117.09
C206	3	588.49	Rochat & Taillard (1995)	10	1189.92
C207	3	588.29	Rochat & Taillard (1995)	10	1109.08
C208	3	588.32	Rochat & Taillard (1995)	10	1172.29
RC201	2	708.00	May et al. (2021)	10	1249.62
RC202	2	717.00	May et al. (2021)	10	1195.54
RC203	2	722.00	May et al. (2021)	10	1240.28
RC204	2	711.00	May et al. (2021)	10	1196.61
RC205	2	713.00	May et al. (2021)	10	1287.49
RC206	2	718.00	May et al. (2021)	10	1263.11

Table 15 Comparison of the algorithm performance on instances of the Solomon Benchmark for type 1 and type 2. (Continue)

Instance	Best-know solution		Ref.	Proposed Stas crossover with K-mean Clustering	
	vehicles	distance		vehicles	distance
RC207	2	718.00	May et al. (2021)	10	1309.64
RC208	2	717.00	May et al. (2021)	10	1234.96

CHAPTER 5
CONCLUSION

It is a long evolution phase for GA algorithms. GA is the search algorithms and optimization methods. The basic concept is based on the mechanism of evolution and natural selection, according to the Darwin's theory of survival of the fittest. John Holland in the 1970s introduced this idea by considering difficult optimization problems. It can be solved with such an evolutionary approach. Therefore, the author saw that there are good opportunities for future contributions. In this study, the author proposed to create a new crossover operator to solve multi-product and multi-period AGG problems and Solomon's benchmark problem instances for VRPTW.

A novel crossover operator for genetic algorithm is called "Stas Crossover". It is a combination of four crossover operators, including Single point crossover, Two points crossover, Arithmetic crossover, and Scattered crossover. It can be adjusted according to the size of the probability area in each operator, which means the probability of creating a new offspring for each operator is not an equal chance. The most important advantage of Stas crossover is that it provides greater diversity in the choice of methods for creating offspring and increases the opportunity for offspring to directly obtain good genetic information.

Stas crossover for aggregate production planning problems, the algorithm was tested by using the multi-product and multi-period APP problems to minimize total costs in terms of regular time, overtime, backordering, and inventory costs. This study focused on the large size of the problem and the optimal solution by changing production costs, backorder costs, production capacity, and forecast demand, which means the actual value of the answer is ignored. A detailed comparison is presented of GA approach for solving APP problems using four different crossover options and a new crossover to compare the behavior of the crossover and choose appropriate crossover options for solving the APP problem. For APP problems, Scattered crossover is shown as having the best performance, but Stas crossover performs better than another crossover. The initial assumption is reduced the percentage of Single point crossover and Two points crossover resulting in an improvement in the answer's better performance than adjusting the size of the area probability of other Stas crossovers, including Stas0055 crossover and Stas1117 crossover.

Stas crossover with K-mean clustering for vehicle routing problems with time windows was tested by the standard Solomon's benchmark problem instances for VRPTW. This study focused on 8 types of crossover operators, which are described with six problem sets of different problem types C1, R1, RC1, C2, R2, and RC2. Each set contained between eight to twelve 100-node problems, and appropriated crossover operators are recommended for each type of problem. In this part, the author divided the results into four parts, including minimum distance with K-mean clustering, minimum distance without K-mean clustering, average distance with K-mean clustering, and average distance without K-mean clustering. The results shown that K-mean clustering is better than without K-mean clustering for minimum distance and average distance for Set R, Set C, and Set RC with type 1 and type 2. The paths with K-mean clustering are arranged into groups and are orderly, but the paths without K-mean clustering are disordered in terms of location and dispersion characteristics of the customer. Moreover, the results shown that the proposed algorithm outperforms Set R and Set RC in some instances, which meant that K-means clustering affects Set R and Set RC due to the paths being arranged into orderly groups. Moreover, K-mean clustering has not affected Set C due to the location coordinates of all customers already clustered. It has been shown that adding K-mean clustering to the Stas crossover efficiently contributes to its performance. Furthermore, the proposed research will serve as a guideline for a real-world case study. This model can be applied to linear and non-linear programming which is a large problem.

REFERENCES

Ahmed, Z. H., Maleki, F., Yousefikhoshbakht, M., & Haron, H. (2023). Solving the vehicle routing problem with time windows using modified football game algorithm. *Egyptian Informatics Journal, 24*, 1-13.

Anggodo, Y. P., Ariyani, A. K., Ardi, M. K., Firdaus, W., & Mahmudy, W. F. (2016). OPTIMIZATION OF MULTI-TRIP VEHICLE ROUTING PROBLEM WITH TIME WINDOWS USING GENETIC ALGORITHM. *Journal of Environmental Engineering & Sustainable Technology, 3*(2), 92-97. doi:http://dx.doi.org/10.21776/ub.jeest.2017.003.02.4

Ariyani, A. K., Mahmudy, W. F., & Anggodo, Y. P. (2018). Hybrid Genetic Algorithms and Simulated Annealing for Multi-trip Vehicle Routing Problem with Time Windows. *International Journal of Electrical and Computer Engineering (IJECE), 8*(6), 4713-4723. doi:10.11591/ijece.v8i6

Awawdeh, S., Edinat, A., & Sleit, A. (2019). An Enhanced K-means Clustering Algorithm for Multi-attributes Data. *International Journal of Computer Science and Information Security (IJCSIS), 17*(2), 1-6.

Carr, J. (2014). An Introduction to Genetic Algorithms. Retrieved from http://www.edc.ncl.ac.uk/highlight/rhjanuary2007.php

Chakrabortty, R. K., & Md.A.Akhtar. (2013). Solving an aggregate production planning problem by using multi-objective genetic algorithm (MOGA) approach. *International Journal of Industrial Engineering Computations, 4*, 1-12.

Dalton, J. (2007). Genetic Algorithms (GAs). Retrieved from http://www.edc.ncl.ac.uk/highlight/rhjanuary2007.php

Dantzig, G., & Ramser, J. (1959). The Truck Dispatching Problem. *Management Science, 6*, 80-91. doi:http://dx.doi.org/10.1287/mnsc.6.1.80

Duan, J., He, Z., & Yen, G. G. (2022). Robust Multiobjective Optimization for Vehicle Routing Problem With Time Windows. *TRANSACTIONS ON CYBERNETICS, 52*(8), 8300-8314. doi:10.1109/TCYB.2021.3049635

Gallego, G. (2021). Production Management. Retrieved from http://www.columbia.edu/~gmg2/4000/pdf/lect_05.pdf

Gambardella, L. M. (2000). MACS-VRPTW: A Multiple Ant Colony Optimization System for Vehicle Routing Problems with Time Windows (VRPTW). Retrieved from https://people.idsia.ch/~luca/macs-vrptw/solutions/welcome.htm

Ghani, N. E. A., Shariff, S. S. R., & Zahari, S. M. (2016). An Alternative Algorithm for Vehicle Routing Problem with Time Windows for Daily Deliveries. *Advances in Pure Mathematics, 6*, 342-350. doi:http://dx.doi.org/10.4236/apm.2016.65025

Gocken, T., & Yaktubay, M. (2019). COMPARISON OF DIFFERENT CLUSTERING ALCORITHMS VIA GENETIC FOR VRPTW. *International Journal of Simulation Modelling, 18*(4), 574-585. doi:10.2507/IJSIMM18(4)485

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization & Machine Learning*. Pearson Education Pvt. Ltd.: Singapore.

Hartono, Ongko, E., & Abdullah, D. (2015). Determining a Cluster Centroid of K-Means Clustering Using Genetic Algorithm. *International Journal of Computer Science and Software Engineering (IJCSSE), 4*(6), 160-164.

Heizer, J., Render, B., & Munson, C. (2017). *Operations Management: Sustainability and Supply Chain Management*. Pearson Education Limited: England.

Holland, J. H. (1992). Genetic Algorithm. *Scientific American, 267*(1), 66-73.

Ibrahim, M. F., Nurhakiki, F. R., Utama, D. M., & Rizaki, A. A. (2020). Optimised Genetic Algorithm Crossover and Mutation Stage for Vehicle Routing Problem Pick-Up and Delivery with Time Windows *International Conference on Advanced Science and Technology (ICAST 2020), 1071*. doi: doi:10.1088/1757-899X/1071/1/012025

Ibrahim, M. F., Putri, M. M., Farista, D., & Utama, D. M. (2021). An Improved Genetic Algorithm for Vehicle Routing Problem Pick-up and Delivery with Time Windows. *Jurnal Teknik Industri, 22*(1), 1-17. doi:10.22219/JTIUMM.Vol22.No1.1-17

Ikotun, A. M., Ezugwua, A. E., Abualigah, L., Abuhaija, B., & Heming, J. (2023). K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. *Information Sciences, 622*, 178-210. doi:https://doi.org/10.1016/j.ins.2022.11.139

Kallehauge, B., Larsen, J., O.B.G., M., & Solomon, M. M. (2005). The Vehicle Routing Problem with Time Windows. *Column Generation*, 67-98.

Kaya, Y., Uyar, M., & Tekin, R. (2011). A Novel Crossover Operator for Genetic Algorithms: Ring Crossover. *Neural and Evolutionary Computing*. doi:https://doi.org/10.48550/arXiv.1105.0355

Kinoshita, T., & Uchiya, T. (2021). Diversity Maintenance Method Using Multiple Crossover in Genetic Algorithm for VRPTW. *2021 IEEE 10th Global Conference on Consumer Electronics (GCCE)*, 563-565. doi:10.1109/GCCE53005.2021.9621864

Krajewski, L. J., & Ritzman, L. P. (1999). *Operations Management: Strategy and Analysis*. United States of America: Wesley Publishing Company.

Kumar, S. N., & Panneerselvam, R. (2015). A Comparative Study of Proposed Genetic Algorithm-Based Solution with Other Algorithms for Time-Dependent Vehicle Routing Problem with Time Windows for E-Commerce Supply Chain. *Journal of Service Science and Management, 8*, 844-859. doi:10.4236/jssm.2015.86085

MacQueen, J. B. (1967). Some Methods for classification and Analysis of Multivariate Observations. *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, 1*, 281-297.

May, A. T., Jariyavajee, C., & Polvichai, J. (2021). An Improved Genetic Algorithm for Vehicle Routing Problem with Hard Time Windows. *Proc. of the International Conference on Electrical, Computer and Energy Technologies (ICECET)*, 1-6. doi:10.1109/ICECET52533.2021.9698698

Michalewicz, Z. (1994). *Genetic Algorithms + Data Structures = Evolution Programs*. Springer: Verlag Berlin Heidelberg.

REGUANT, R. (2021). GENETIC ALGORITHM PYTHON EXAMPLE. Retrieved from https://rocreguant.com/genetic-algorithm- python- example/1977/

Revanna, J. K. C., & B.Al-Nakash, N. Y. (2022). Vehicle Routing Problem with Time Window Constrain using KMeans Clustering to Obtain the Closest Customer. *Global*

Journal of Computer Science and Technology, 12(1), 49-65. doi:https://doi.org/10.34257/GJCSTDVOL22IS1PG49

Savsani, P., Banthia, G., Gupta, J., & Vyas, R. (2016). Optimal Aggregate Production Planning by using Genetic Algorithm. *Proceedings of the 2016 International Conference on Industrial Engineering and Operations Management Kuala Lumpur, Malaysia*, 863-874.

Shukla, A., Tiwari, R., & Kala, R. (2010). Towards Hybrid and Adaptive Computing. *307*. doi:10.1007/978-3-642-14344-1

Sivanandam, S. N., & Deepa, S. N. (2008). *Introduction to Genetic Algorithmsx*. Springer: Verlag Berlin Heidelberg.

Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research, 35*, 254-265.

Solomon, M. M. (2005). Best Known Solutions Identified by Heuristics, Northeastern University, Massachusetts, Boston. Retrieved from http://web.cba.neu.edu/~solomon/heuristic.htm.

Solomon, M. M., & Desrosiers, J. (1998). Time Window Constrained Routing and Scheduling Problems. *Transportation Science, 22*(1), 1-11.

Sripriya, J., Ramalingam, A., & Rajeswari, K. (2015). A HYBRID GENETIC ALGORITHM FOR VEHICLE ROUTING PROBLEM WITH TIME WINDOWS *International Conference on Innovations in Information Embedded and Communication Systems (ICIIECS), Coimbatore, India*, 1-4. doi:10.1109/ICIIECS.2015.7193072

Stoykova, S., & SpasovChoice, V. (2019). Choice of fitness functions and parameter settings in Genetic Algorithms for analysis of induction motors. *IOP Conference Series: Materials Science and Engineering, 618*, 1-8. doi:10.1088/1757-899X/618/1/012024

Suryawanshi, R., & Puthran, S. (2016). A Novel Approach for Data Clustering using Improved K-means Algorithm. *International Journal of Computer Applications (0975 – 8887) 142*(12), 13-18.

Tawanda, T., Nyamugure, P., Kumar, S., & Munapo, E. (2023). ALabelling Method for the Travelling Salesman Problem. *Applied Sciences, 13*(6417), 1-12. doi:https://doi.org/10.3390/app13116417

Thangiah, S. R. (1995). Vehicle Routing with Time Windows using Genetic Algorithms. *Applications Handbook of Genetic Algorithms: New Frontiers*, 253 – 278. doi:10.1201/9781420050073.ch11

Toth, P., & Vigo, D., eds. . (2002). The Vehicle Routing Problem. *Monographs on Discrete Mathematics and Applications, Philadelphia: Society for Industrial and Applied Mathematics, 9*.

UNSW. (2022). ROULETTE WHEEL SELECTION. Retrieved from http://www.cse.unsw.edu.au/~cs9417ml/GA2/selection_roulette.html

VITA

NAME	Ratchadakorn Poohoi
INSTITUTIONS ATTENDED	Master of Engineering (Engineering Management) Silpakorn University
PUBLICATION	Poohoi, R., Puntusavase K., and Ohmori S. (2023). “Stas crossover with K-mean clustering for vehicle routing problem with time window” Decision Science Letters, Vol. 13(3): pp. 525 – 534. (SCOPUS). Poohoi R., Puntusavase K., and Saicharoen K. (2024). “Genetic algorithm to develop decision support system for police officer assignment” in Proceeding of International Conference on Artificial Intelligence Research, May 22 -23, 2024, Tokyo, Japan, pp. 6. Sangpanin, T., Hmuanjainuk, N., Phoonsup, K., Saicharoen, K., Poohoi, R. (2024). “A comparison of the forecasting methods to increase accuracy in sales forecasts for case study company” in Proceeding of the Conference of Industrial Engineering Network (IE NETWORK 2024), Thailand, May 15 – 17, 2024, pp. 685 – 690. Poohoi, R., Puntusavase K., and Ohmori S. (2023). “A novel crossover operator for genetic algorithm: Stas crossover” Decision Science Letters, Vol. 12(3): pp. 515 – 524. (SCOPUS). Poohoi, R., Pornsing, C., and Klomjit, P. (2021). “Queuing

System Analysis: A Case Study of Fast Fashion Business" Thai Journal of Industrial Technology, Vol.17, No.3 (Sep. – Dec.): pp.124 – 133. (TCI 1).

Poohoi, R., and Klomjit, P. (2021). "Operational risk analysis: case study of plastic bag manufacturer" Journal of Engineering and Innovation, Vol.15, No.4 (Oct. – Dec.): pp. 120 – 128. (TCI 1).