

**Hybrid Genetic Algorithm Approach
for Production Inventory Routing Problem with
Perishable Products**

by

Ege Somer

A Dissertation Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in

Industrial Engineering



**KOÇ
ÜNİVERSİTESİ**

August 18, 2023

Hybrid Genetic Algorithm Approach for Production Inventory Routing Problem with Perishable Products

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Ege Somer

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

Prof. Ceyda Oğuz

Assist. Prof. Barış Yıldız

Assist. Prof. Hande Küçükaydın

Date: 18.08.2023



To my dear mother & father, Leyla & Tevfik Somer

ABSTRACT

Hybrid Genetic Algorithm Approach for Production Inventory Routing Problem with Perishable Products

Ege Somer

Master of Science in Industrial Engineering

August 18, 2023

Production, inventory and distribution decisions have always been challenging for decision-makers who are trying to minimize the cost at all levels of a supply chain. There has been a growing interest in both industry and academia for jointly optimizing these problems which becomes a very complex problem. In this study, integrated optimization of production, inventory and distribution decisions for perishable products is considered. This perishable production-inventory routing problem (PPIRP) includes a single production plant, a set of retailers with a time-varying and deterministic demands, a fleet of homogenous capacitated vehicles to make the deliveries, a single and perishable product, a set of quality levels related to the shelf life of the product and a finite planning horizon. A mixed integer linear programming (MILP) model is developed to solve this problem, which naturally combines the well-known lot-sizing problem and vehicle routing problem. Given that a mathematical model is not viable to solve instances with more than 10 retailers to optimality within a reasonable time, a Hybrid Genetic Algorithm (HGA) is developed to get high-quality and near-optimal solutions.

Extensive computational experiments have been performed by comparing our algorithm with an adapted benchmark matheuristic in the literature, Two-Phase Iterative Heuristic (TIH). The results show that our algorithm outperforms TIH in general and can produce high-quality solutions. Then, managerial insights are discussed over the results.

ÖZETÇE

Yüksek Lisans Tez Başlığı

Ege Somer

Endüstri Mühendisliği, Yüksek Lisans

August 18, 2023

Üretim, envanter ve dağıtım kararları, bir tedarik zincirinin tüm seviyelerinde maliyeti minimize etmeye çalışan karar vericiler için her zaman zorlu olmuştur. Bu problemlerin birlikte optimize edilmesi oldukça karmaşık bir probleme dönüşmektedir ve dolayısıyla bu probleme yönelik endüstride ve akademide artan bir ilgi oluşmuştur. Bu çalışmada, bozulabilir ürünler için üretim, envanter ve rotalama kararlarının entegre olarak optimize edilmesi ele alınmıştır. Bu bozulabilir üretim-envanter-rotalama problemi (BÜERP), tek bir üretim tesisi, değişken zamanlı ve deterministik taleplere sahip bir perakendeci kümesi, teslimatları gerçekleştirmek için homojen kapasiteli bir araç filosu, tek ve bozulabilir bir ürün, ürünün raf ömrüyle ilişkili kalite seviyeleri ve sonlu bir planlama dönemini içermektedir. Bu problemi çözmek için karışık tamsayılı doğrusal programlama (KTDP) modeli geliştirilmiştir. Bu model bilinen üretim miktarı belirleme problemini ve araç rota problemini doğal olarak birleştirir. Matematiksel modelin makul bir süre içinde büyüklüğü 10 perakendeciyi geçen örneklerde optimal sonucu bulması mümkün olmadığından, yüksek kaliteli ve optimale yakın çözümler elde etmek için Hibrit Genetik Algoritma (HGA) geliştirilmiştir.

Literatürden uyarlanan matsezzgisel, İki Aşamalı İteratif Sezgisel (IAİS), ile algoritmanızın karşılaştırıldığı kapsamlı hesaplama deneyleri gerçekleştirilmiştir. Sonuçlar, algoritmanızın genel olarak IAİS'yi geride bıraktığını ve yüksek kaliteli çözümler üretebildiğini göstermektedir. Ardından sonuçlar üzerinde yönetimsel içgörüler tartışılmıştır.

AKNOWLEDGEMENTS

I would like to thank my advisor Prof. Ceyda Oğuz for her invaluable guidance during my graduate studies and teaching assistantship. I feel extremely lucky to have the opportunity to work under her supervision.



TABLE OF CONTENTS

List of Tables	ix
List of Figures	x
Abbreviations	xi
Chapter 1: INTRODUCTION.....	1
1.1 Our Contribution	3
1.2 Thesis Structure.....	4
Chapter 2: LITERATURE REVIEW	5
2.1 Studies Related to PIRP.....	5
2.2 Studies of PIRP and IRP with Perishable Products	7
2.3 Matheuristic Algorithms for PIRP and IRP	11
2.4 Metaheuristic Algorithms for PIRP and IRP	15
Chapter 3: PROBLEM DEFINITION.....	19
3.1 Problem Statement	19
3.2 Mathematical Model.....	21
Chapter 4: SOLUTION METHODOLOGY	25
4.1 Hybrid Genetic Algorithm.....	25
4.1.1 General structure of the HGA.....	26
4.1.2 Chromosome Representation.....	28
4.1.3 Encoding.....	30
4.1.4 Decoding.....	30
4.1.4.1 LP Model.....	31
4.1.4.2 Trimming Process.....	33
4.1.4.3 Obtaining Routing Variables.....	33
4.1.5 Fitness Function	34
4.1.6 Infeasibility Repairing Mechanisms.....	34
4.1.6.1 Repairing Infeasible Production Set Up Schedule.....	35

4.1.6.2	Repairing Infeasible Visit Schedule.....	36
4.1.7	Initial Population Construction	37
4.1.8	Crossover	39
4.1.9	Mutation.....	40
4.1.9.1	Mutation Operators.....	41
4.1.9.2	Operator Selections.....	43
4.1.10	Enlarged Sampling Space Formation and Selection Process.....	44
4.1.11	Stopping Criteria	45
4.1.12	Elite Set Improvement.....	45
4.2	Adaptation of the Two-Phase Iterative Heuristic for PIRP	46
4.2.1	Variant of the Lot Sizing Problem	49
4.2.2	Vehicle Routing Problem	51
4.2.2.1	Tabu Search.....	52
Chapter 5: COMPUTATIONAL EXPERIMENTS		55
5.1	Instance Generation.....	55
5.2	Parameter Settings.....	57
5.3	Numerical Results	59
5.3.1	Performance Analysis of Small Instances	59
5.3.2	Performance Analysis of Medium and Large Instances.....	64
5.3.3	Managerial Insights	75
5.3.3.1	Insights into the Production-Oriented Decisions.....	75
5.3.3.2	Insights into the Inventory-Oriented Decisions.....	77
5.3.3.3	Insights into the Transportation-Oriented Decisions.....	78
Chapter 6: CONCLUSION		79
Bibliography.....		81

LIST OF TABLES

Table 2. 1: Literature Review	18
Table 5. 1: Fixed Parameters of HGA	58
Table 5. 2: Population Size per Instance Size	58
Table 5. 3: TIH and HGA performances on class I-small instances	61
Table 5. 4: TIH and HGA performances on class II-small instances	62
Table 5. 5: TIH and HGA performances on class III-small instances	63
Table 5. 6: TIH and HGA performances on class IV-small instances	64
Table 5. 7: Average relative differences (%) of the objective values of TIH and HGA runs on medium and large instances.....	65
Table 5. 8: TIH and HGA performances on class I-II instances with 3 periods.....	688
Table 5. 9: TIH and HGA performances on class III-IV instances with 3 periods.....	690
Table 5. 10: TIH and HGA performances on class I-II instances with 6 periods.....	71
Table 5. 11: TIH and HGA performances on class III-IV instances with 6 periods.....	72
Table 5. 12: TIH and HGA performances on class I-II instances with 10 periods	73
Table 5. 13: TIH and HGA performances on class III-IV instances with 10 periods.....	74
Table 5. 14: Average relative gaps (%) of production costs between TIH and HGA for each instance class.....	76
Table 5. 15: Average relative gaps (%) of inventory costs between TIH and HGA for each instance class.....	77
Table 5. 16: Average relative gaps (%) of transportation costs between TIH and HGA for each instance class	78

LIST OF FIGURES

Figure 1. 1: Production Inventory Routing Problem.....	2
Figure 3. 1: Network Representation of PPIRP	20
Figure 4. 1: Framework of HGA.....	27
Figure 4. 2: Chromosome representation of the instance with	31
Figure 4. 3: Decoding Framework of a Chromosome	32
Figure 4. 4: (a) Repairing of the production schedule where there is a tour before the first production period. (b) Repairing of the visit schedule where there are more than $ L $ consecutive periods without delivery, (c) Repairing of the visit schedule where there is no single visit until the latest required period	37
Figure 4. 5: Crossover operation for 10 retailers and 3 periods, one-point crossover (a) with the red cell is repaired, two-point crossover (b).....	41
Figure 4. 6: Mutation Operators.....	44
Figure 4. 7: Framework of the TIH algorithm	53
Figure 4. 8: Exchange move, (a) intra-route and (b) inter-route.....	54
Figure 5. 1: Average Run times (seconds) of TIH and HGA for 3, 6 and 10 periods....	67

ABBREVIATIONS

PIRP	Production Inventory Routing Problem
PPIRP	Perishable Production Inventory Routing Problem
IRP	Inventory Routing Problem
VRP	Vehicle Routing Problem
LSP	Lot Sizing Problem
IPDP	Integrated Production Distribution Problem
MILP	Mixed Integer Linear Programming
LP	Linear Programming
HGA	Hybrid Genetic Algorithm
GA	Genetic Algorithm
TIH	Two-Phase Iterative Heuristic
TS	Tabu Search

Chapter 1: INTRODUCTION

Supply chain planning has always been a challenging task for decision makers, yet is extremely critical for the sake of success and sustainability of especially medium and large companies. Planners specifically focus on decisions regarding production, replenishment and distribution of products. The integration of these different planning decisions have significant advantages for minimizing the overall operational costs and allowing companies to streamline their operations. Hence, integrated supply chain planning is a key factor for companies to stay competitive within the business environment.

The production inventory routing problem (PIRP) optimizes production, inventory and distribution decisions simultaneously. In other words, it combines two well-known problems, the lot-sizing problem (LSP) and the vehicle routing problem (VRP), respectively. *Wagner and Within (1958)* introduced the lot-sizing problem that consists of minimizing production, lot-sizing and inventory costs over a planning horizon. They were focused on the various parts of the supply chain planning apart from the distribution phase, which is related to delivering the products to customers concerning their demand quantities within discrete planning periods. To be more specific, PIRP is literally the integrated version of the LSP with direct shipments and VRP. *Federgruen and Tzur (1999)* were the first to discuss this problem which is the variant of the LSP called integrated lot-sizing with direct shipment. In addition to production lot-sizing and inventory decisions, they integrated the delivery quantities and schedule of distribution planning over a discrete and finite time horizon. They reviewed the problem with one central plant and multi-retailers. In the problem they study, direct shipment costs are assumed to be fixed, thus routing decisions are disregarded. On the other hand, *Dantzig and Ramser (1959)* studied the well-known vehicle routing problem which addresses the routing decisions to make the deliveries by vehicles in each period. LSP and VRP have been studied widely and separately in the literature. However, only a very small part of the studies has been devoted to the integration of the two problems as in the PIRP. It has

been noticed that focusing on a specific part highly affects the overall operational costs. For instance, keeping low inventory levels may increase production and distribution cost. Hence integrated version of these problems has been strongly encouraged as a research area. On the other perspective, the PIRP also contains another well-known problem called inventory routing problem (IRP). *Bell et al. (1983)* discussed IRP for the first time within a gas delivery study. In IRP, production quantities are not a concern, and typically known at the beginning of the planning horizon. IRP integrates inventory, delivery and routing decisions. From that perspective, IRP is the reduced version of the PIRP where production decisions are dropped. Figure 1.1 demonstrates the boundaries of these three integrated problems. Note that, all the studies discussed in this work have arisen in the context of vendor managed inventory (VMI) in which supplier is the central decision maker who specifically plans production, replenishment and distribution related decisions in a supply chain. That is to say, extensive review of the IRP literature has also been performed inher-

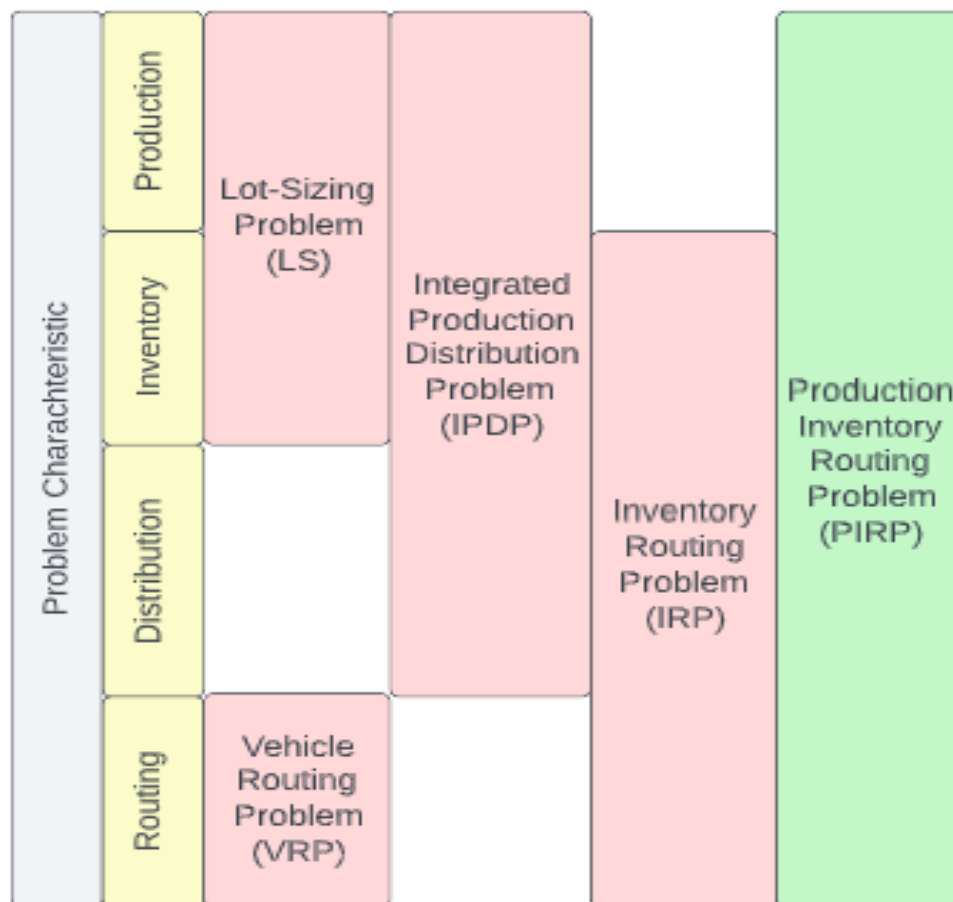


Figure 1.1: Production Inventory Routing Problem

ently due to closeness of the problem structures. Since PIRP is a very complex problem, IRP has been studied more, so solution algorithms designed for IRP have also been an inspiration for this study.

The problem presented in this study considers a PIRP with a single perishable product which gradually decays over time, until a predetermined fixed shelf life is reached. We refer to our problem as a perishable production inventory routing problem (PPIRP). Having said that PIRP is a very complex problem itself, perishability adds even more complexity. In this context, it is also relevant to mention that both in the literature of IRP and PIRP, perishable products are considerably less studied than non-perishable products. Therefore, this study contributes by enriching the solution approaches in this area. In our PPIRP, a single plant with a production and inventory capacity produces the single perishable product. All the production, replenishment and distribution decisions are jointly taken by a central decision maker where multiple retailers with limited storage capacities at different locations have deterministic demands over a planning horizon. Distribution of products are provided by a homogeneous fleet with a certain capacity.

1.1 Our Contribution

Since the decomposable structure of PIRP enables deriving variety of subproblems to be solved easily with mathematical programming, matheuristic algorithms take the major part for the PIRP studies last decades. On the other hand, metaheuristic approaches are studied more frequently for the IRP. However, studies for metaheuristic algorithms in the context of the perishable production inventory routing problem are considerably scarce, as it is shown in the following chapter. Therefore, we contribute by developing a strong metaheuristic algorithm that is able to compete and outperform one of the most commonly used matheuristic in this context. To the best of our knowledge, there is no metaheuristic algorithm studied for our variant of the perishable production routing problem.

While the differences between our problem and related examples from literature are discussed and revealed in Chapter 2, we summarize our contributions below:

- Considering the scarcity of metaheuristic algorithms to other solution approaches for perishable production inventory routing problem, we contribute by closing the research gap especially in evolutionary algorithms, as indicated in the literature.
- We develop a Hybrid Genetic Algorithm and adapted the existing and common matheuristic algorithm, Two-Phase Iterative Heuristic, to tackle with the problem.
- We propose a novel random instance set for our problem by modifying a benchmark instance for a similar perishable inventory routing problem in the literature.
- We show that our proposed Hybrid Genetic Algorithm returns high quality solutions and outperforms Two-Phase Iterative Heuristic for the medium and large size instances by an extensive computational study.

1.2 Thesis Structure

The remainder of this thesis study is organized as follows. In Chapter 2, literature of PIRP and IRP is reviewed. In Chapter 3, we provide a formal description of the problem. Then in Chapter 4, we discuss our main contribution, Hybrid Genetic Algorithm (HGA), in addition to the adaptation of a two-phase iterative algorithm which is inspired by an existing study in the literature and is used as a benchmark algorithm to compare with our proposed HGA. Finally, we present the computational results of the algorithms in Chapter 5 and compare the performance of two solution methods in various aspects, followed by conclusions in Chapter 6.

Chapter 2:

LITERATURE REVIEW

In this chapter, reviewed studies are categorized into three subsections. First, is a generalized overview of the studies that address PIRP. Note that PIRP is named as production–distribution problem in some research and studies without routing decisions are also reviewed due to the easily decomposable structure of the PIRP. Secondly, studies that have considered PIRP or IRP where products are perishable are taken into consideration. In the third and fourth subsections, a review of the matheuristic and metaheuristic algorithms to solve either PIRP or IRP is provided since we make our contribution based on these types of algorithms.

2.1 Studies Related to PIRP

First investigations to jointly optimize production, inventory, distribution and routing decisions have been made by *Chandra (1993)* and *Chandra and Fisher (1994)*. They showed that integrating the lot-sizing and routing problems leads to 3-20% cost saving for PIRP compared to solving the problems separately and sequentially. However, note that they considered multi-products, an unlimited fleet-size and split deliveries. *Lei et al. (2006)* addressed a PIRP formulation with a vehicle index where capacitated heterogeneous fleet was considered. *Bard and Nananukul (2009)* proposed a mixed integer linear model for PIRP. The objective was to minimize the total cost with a single product and a single production plant without capacity limitation. They proposed a tabu search (TS) algorithm combined with path-relinking in order to solve the problem. *Bard and Nananukul (2010)* considered a two-level supply chain and proposed a mixed-integer programming model in order to determine production, inventory and distribution related decisions. They also developed a hybrid method that utilizes column generation and heuristics within a branch and price framework. *Archetti et al. (2011)* investigated the production-distribution problem where supply chain was established on two-echelon and one single item is produced and distributed to multiple retailers. They analyzed two different replenishment policies, which are order-up to level (OU) and maximum level policies. They designed a hybrid heuristic for the maximum level (ML) policy and an

exact algorithm for the problem with one vehicle. *Yan et al. (2011)* studied two-level production-distribution problem and proposed a mixed-integer linear programming model in which routing decisions were eliminated. Another research without vehicle routing part was studied by *Melo and Wolsey (2012)*. They addressed a two-level supply chain which is consisted of multiple retailers and multiple production sites. Delivery phase of their model only included allocation of the products at the production level and the objective was to minimize the total cost incurred from storage, production and transportation.

Absi et al. (2014) studied the PIRP consisting of two echelons with a central production facility that produces a single product which is distributed to multiple retailers. They proposed a two-phase-iterative heuristic by iteratively solving the decomposed lot-sizing and vehicle routing models. In our study, we adapted their algorithm (i.e., Section 4.2) to solve our PPIRP then used the algorithm as a benchmark to our proposed HGA. *Adulyasak and Coerdau (2014a)* addressed two different PIRP models, which differentiated with vehicle indices, and developed a branch-and-cut algorithm to solve each model. *Adulyasak et al. (2015a)* provided a comprehensive review of the studies related with PIRP by considering various variants of the PIRP formulations and solution algorithms. *Adulyasak et al. (2015b)* considered the PIRP with stochastic demands, as they claimed that demands are not deterministic in many supply chain networks. They proposed two formulations with stochastic demand and solved the models with a branch-and-cut algorithm together with benders decomposition for different numbers of demand scenarios. *Moin and Yuliana (2015)* proposed a scatter search metaheuristic to solve traditional single item and single plant PIRP. *Brahimi and Aouam (2016)* studied PIRP with multi-product and back ordering policy. They suggested a hybrid relax and fix heuristic that includes local search procedures. *Kumar et al. (2016)* introduced a multi-objective mixed-integer model for PIRP including pollution routing consideration with time windows, then they proposed a self-learning particle swarm optimization approach. *Solyalı and Süral (2017)* considered the PIRP within a two-stage supply chain in which a plant produces a single product and distributes to multiple retailers. They developed a five-phase matheuristic approach to solve the problem.

Qiu et al. (2016) addressed a pollution-production routing problem where they aimed to minimize greenhouse gas emissions (GHG) caused by production and

distribution operations and suggested a branch-and-price heuristic. The multi-scale PIRP is studied by *Zhang et al. (2017)*. They proposed a mixed-integer model for a multi-commodity supply chain with continuous production facilities. Their model includes two different time grids that separately consider production scheduling and vehicle routing. They suggested an iterative MIP-based heuristic approach that solves the problem with a restricted set of feasible routes and updates the candidate routes dynamically. *Darvish et al. (2018)* studied an integrated production, location, inventory and distribution problem with multiple products, multiple plants and multiple echelons and developed several sequential heuristics and matheuristic which was then compared with exact methods. *Chitsaz et al. (2019)* developed a mathematical programming method for the extension of the PIRP with assembly decisions. Their model also included inbound transportation and inventory decisions. *Li et al. (2019)* investigated a two-level multi-product PIRP with outsourcing and proposed a three-level matheuristic algorithm to solve the problem. *Torkaman et al. (2020)* addressed a mixed-integer non-linear programming formulation for the PIRP where demand is a function of the price of a single product and suggested an outer approximation algorithm. *Schenekemberg et al. (2021)* suggested an exact algorithm which combined a branch-and-cut and a local search method to solve the classic two-echelon PIRP. *Qiu et al. (2021)* introduced a mixed-integer programming model for the two-echelon PIRP with cross-docking satellites. They proposed a branch-and-cut algorithm with valid inequalities in which a matheuristic algorithm generated initial feasible solutions. *Manousakis et al. (2023)* introduced the cyclic production routing problem (CPRP) which is an extension of the standard production routing problem (PRP). By generating repeatable cyclic production and delivery schedules. They proposed two commodity-flow formulation with valid inequalities and performed an extensive comparison between the standard and cyclic PRP.

2.2 Studies of PIRP and IRP with Perishable Products

In this section, a review of PIRP and IRP with perishable products is given. Perishable products add another dimension to the complexity of managing the operations within supply chain networks such that quality of the products is time dependent and may decrease throughout the supply chain. Furthermore, to avoid delivering non-fresh

products to customers is obviously crucial in order to conduct a healthy supply chain network. Hence, despite the high complexity of the problem, researchers' interest has significantly increased for developing solution approaches, however, note that studies with perishable products are still relatively few according to the other type of supply chains.

Due to the fact that the studies of PIRP that included perishable characteristic of products are limited, IRP with perishable products is also widely reviewed in addition to PIRP studies. Since IRP is obtained from PIRP by eliminating production lot-sizing decisions, solution algorithms and methods can be easily compared with PIRP solutions. *Geismar and Laporte (2008)* studied the production and transportation scheduling problem (PTSP) with capacity constraints that involves a short life product, thus, no inventory of the product is held during the process. PTSP also includes detailed production scheduling, that is to say, it differs from the classical PIRP in that aspect. Authors aimed to determine the minimum time required to complete producing and delivering the product to meet customers' demand in a two-echelon supply chain. They proposed a two-phase heuristic to solve the problem. *Chen et al. (2009)* proposed a non-linear mathematical model that considers production scheduling and vehicle routing under time windows and perishable food product restrictions within a continuous time horizon. They addressed the demands in a stochastic environment. Quality of the products deteriorates after production. They developed a solution algorithm for this high complexity problem including Nelder-Mead method and heuristic for vehicle routing. *Rong et al. (2011)* addressed a mixed-integer linear programming model for production and distribution within a food supply chain. They elaborated quality degradation modeling involving temperature effect. *Farahani et al. (2012)* investigated a production scheduling and distribution planning problem. They developed an integrated approach by first decomposing the production model into two phases and then integrating them with a distribution model. This approach resulted with a reduction in the problem complexity by a considerable amount and improved the quality of the products by shortening the time interval between production and delivery in an iterative scheme.

Coelho and Laporte (2014) addressed an IRP formulation where a single product with fixed-shelf-life is considered. Revenue and the holding costs are dependent to age

of the product which helps decision makers to determine to meet customer demands either with aged or fresh products. They solved the problem with an exact branch-and-cut algorithm. They implemented their solution approach under different selling priority policies. *Jia et al. (2014)* proposed a mixed-integer linear programming model for IRP with product quality deterioration and time windows. Their model's contribution lies in the consideration of loading costs and returning time intervals. They proposed a two-phase algorithm that includes metaheuristics. *Soysal et al. (2015)* considered a chance-constrained programming model for IRP within discrete planning horizon where products' quality decays gradually throughout the horizon. They also took fuel costs and wastage costs into consideration. *Belo-Filho et al. (2015)* tackled the integrated production and distribution planning problem with multiple perishable products with limited lifespan. In their study, the production phase involves detailed scheduling, line assignment and lot-sizing decisions which differs from the classical PIRP. They developed an adaptive large neighborhood search (ALNS) to solve this problem in a reasonable time. In fact, their study is an extension of the work presented by *Amorim et al (2012)*. Authors proposed a multi-objective mathematical model to consider PIRP in a two-level supply chain which includes multi-plant production with perishable products. *Le et al. (2013)* addressed an IRP with perishable product and introduced a mixed-integer formulation with cutting plane improvements. They proposed a column generation based heuristic. *Diabat et al. (2016)* considered a new arc-based formulation for IRP with a single perishable product and proposed a tabu search algorithm. *Li et al. (2016)* addressed a mixed-integer linear programming model for PIRP where the quality of the product is traced throughout the planning horizon. They performed an extensive analysis to demonstrate the impact of perishability within a three-echelon supply chain. *Hiassat et al. (2017)* studied a location – inventory routing problem (LIRP) which adds the number and location of required warehouse decisions to classical IRP. They claim that integrating strategic, tactical and operational decisions produces better results for supply chains. Their model features a single perishable product where incurred costs are age dependent. They proposed a genetic algorithm with local search heuristics to tackle the problem. Study of *Rahimi et al. (2017)* suggested a new IRP formulation considering economical aspect, customer satisfaction and environmental aspect for perishable product in a multi-objective framework. Contribution of this study lies in the integration of the customer satisfaction level concept by the definition of the number of delays in visiting the retailers

and the number of backordered products. In their work, they coped with this complex model with a well-known multi-objective metaheuristic non-dominated sorting genetic algorithm II (NSGA-II). *Azadeh et al. (2017)* introduced an IRP model with transshipment between customers of a single perishable product. Inventory replenishments and vehicle routes are decided in their model under maximum level policy. Products spoil with an exponential rate during the continuous planning horizon. Due to NP-hard nature of the problem, they proposed a genetic algorithm in which parameters were determined with a taguchi based approach to achieve the best solution. Another variant of IRP was introduced in the study of *Crama et al. (2018)* in which a deterministic shelf life for a single perishable product was considered and each vehicle could depart from the depot once in a predetermined maximum duration. Demands of the customers were uncertain with a known probability distribution in the study and several solution approaches with decomposition-based heuristics were proposed. *Neves-Moreira et al. (2019)* considered the rich multi-product PIRP with time windows. They presented a decomposition method based on a fix and optimize scheme.

Another extension of PIRP is proposed by *Qiu et al. (2019)*. They presented a mixed-integer programming formulation where products age with a specific rate within a period. They suggested a branch-and-cut algorithm based on three families of valid inequalities together with analysis of different delivery and selling priority policies. Study of *Absi et al. (2014)* extended with the study of *Chu et al. (2019)* by adding a single perishable product where the quality degrades by one unit per period. The authors tackled a bi-objective problem where the second objective maximizes the average food quality level at the end customer of two-stage supply chain. They addressed an ϵ -constrained based two-phase iterative heuristic to solve the problem. *Rohmer et al. (2019)* presented a two-echelon IRP with age-dependent inventory holding cost and developed a two-stage matheuristic that combines ALNS and mathematical programming. *Alvarez et al. (2020)* introduced four different mathematical model for the IRP, two of them considers vehicle index while others do not. A single product was assumed to have a fixed shelf-life which deteriorates one unit in each period. Exact branch-and-cut algorithm in addition to a hybrid heuristic based on iterative local search and mathematical programming was provided.

Temperature perspective was taken into account by *Manouchehri et al. (2020)*. Storage and vehicle temperatures added another complexity to PIRP as they have great impact on the quality in real life. Degradation rate of multiple products' quality depends on storage time and the environmental conditions. To solve the problem, a variable neighborhood search within a simulated annealing scheme is proposed. *Li et al. (2020)* developed a mixed-integer model for PIRP with multiple production facilities and different packaging considerations that determine different shelf lives for products. In their model, quality of a product does not decay during its shelf-life. Furthermore, profit gained by selling a product is dependent on the length of the duration of the shelf-life. As a solution approach, a hybrid matheuristic was developed by authors. *Mousavi et al. (2022)* considered the PIRP concerning the freshness of a single perishable product with a discrete and limited shelf life under pandemic scenarios where demands of multiple customers are stochastic. Authors aimed to minimize the total cost which consists of production, inventory, routing, wasted products and penalties for aged products. Authors suggested a five-phase matheuristic developed by decomposing the main model into sub problems based on the solution approach presented by *Solyalı and Süral (2017)*. To the best of our knowledge, one of the most recent studies in the field of PIRP with perishable products has been made by *Alvarez et al. (2022)*. They introduced a mathematical formulation for the PIRP with a single perishable product decaying gradually and where transshipments between retailers are allowed. They further analyzed decay rates for the products and transshipment effect on the overall cost. Finally, they proposed a branch-and-cut and a hybrid iterated local search algorithm. The problem presented in this thesis is similar to the problem of *Alvarez et al. (2022)* in a PIRP concept, but they analyzed different forms of perishability in an environment where transshipments between retailers are allowed where instance set they used is based on the standard PRP.

2.3 Matheuristic Algorithms for PIRP and IRP

Due to the decomposable structure of the PIRP and IRP, researchers have been interested in developing heuristics based on the mathematical programming components in recent years. Matheuristic algorithms consist of two or more phases. They are based on iteratively solving relatively smaller mathematical models which we call subproblems.

Furthermore, according to the problem's complexity, in some cases other heuristic methods may be employed within a phase of matheuristic algorithm.

Since both PIRP and IRP contains multiple problems as lot-sizing and vehicle routing, researchers have attempted more on designing matheuristic algorithms. Therefore, extensive review of this type of algorithms for both PIRP and IRP is made in this section. Moreover, one of the main contributions of this study is adapting the solution method proposed by *Absi et al. (2014)* as a benchmark algorithm which was originally designed to solve standard single echelon PIRP. After the adaptation of the algorithm, it is compared with our main algorithm named hybrid genetic algorithm (HGA). In their work, they suggested a two-phase iterative heuristic for PIRP in which they did not consider perishability. First phase of their algorithm corresponds to solving a capacitated and aggregate lot-sizing problem where decisions related to production, inventory and deliveries to retailers in each period are determined. In this phase, routing costs are approximated with visiting costs by considering the aggregate fleet capacity. After solving the first phase and having the distribution plan for each period, routing decisions are determined by iteratively solving VRP for each period. Solution of the second phase was utilized to update the visiting costs. This procedure is repeated until the maximum number of iteration or the maximum number of unimproved solutions is reached. The reason behind this algorithm being selected as a benchmark is that it is a commonly used matheuristic which can be easily implemented.

Chu et al. (2019) enhanced the work of *Absi et al. (2014)* into multi-objective frame. They proposed an ϵ -constrained two-phase iterative heuristic where average quality of the delivered products is limited by ϵ within the aggregate lot-sizing problem. The procedure is repeated in the range between nadir and ideal point for the quality objective. *Mousavi et al. (2022)* extended the five-phase heuristic with enhancements to tackle with perishability factor under uncertain pandemic situations. Phases two and four in the study of *Solyalı and Süral (2017)* were revised regarding to stochastic nature and perishability of the product in their problem. In addition, they suggested a new clustering algorithm for the third phase and a new repairing mechanism for the fourth phase. *Russel (2017)* introduced two mathematical programming-based heuristics which employ a relaxed mixed-integer model for an initial solution. First heuristic utilizes a set

partitioning approach whereas the other one uses the concept of seed routes to obtain an approximate solution for the addressed PIRP. The approximate solution does not involve routing decisions, thus a series of VRP for each period are solved to acquire a complete solution. As a last step, multi-iteration improvement procedure is performed.

Archetti et al. (2017) considered the IRP where a supplier makes replenishment decisions of the customers with limited storage capacity in addition to the distribution plan with capacitated fleet over a discrete planning horizon in order to minimize the total cost incurred from inventories and transportation. They presented a three-level matheuristic that integrates mathematical programming with a tabu search algorithm. Initial solution is constructed by mathematical programming by relaxing the routing variables. A feasible solution is then obtained by a tabu search in the second phase. Finally, using the information collected by the tabu search and fixing frequently observed variables, the feasible solution found with tabu search is improved by a smaller mixed-integer programming model. *Miranda et al. (2018)* addressed a rich production routing problem that contains multiple products, heterogeneous fleet and time windows together with a make-to-order policy. They developed a decomposition-based heuristic which splits the main problem into two subproblems that are solved iteratively. The first decomposition determines the production plan in detail, customer assignments and approximate routing costs. In the second phase, considering the customer assignments, routing decisions are obtained and improved by a local search heuristic. *Chitsaz et al. (2019)* proposed a flexible three-phase iterative matheuristic for the assembly routing problem, which can also be employed for PIRP and IRP with slight modifications. In the first phase, setup schedules are determined by solving a subproblem. In the second phase, production decisions, visiting schedule and delivery quantities are determined. In the last phase, as in the other matheuristics, a series of VRP for each period are solved. *Avci and Yildiz (2019)* considered visit spacing policy (VSP) concerning service quality level which imposes time intervals between two consecutive visits to the same customer for the PIRP. They suggested an iterative matheuristic algorithm that handles the distribution and routing part with a heuristic whereas production, inventory and delivery decisions were obtained by solving another subproblem. Later, *Avci and Yildiz (2020)* considered the PIRP with additional transshipment between retailers' decision this time and suggested a matheuristic algorithm.

Neves and Moreira (2019) introduced a novel formulation for PIRP with delivery time windows and multiple perishable products that should be sent and consumed in a predefined period. They utilized an improvement matheuristic based on a fix-and-optimize for large size instances and they used a decomposition approach to reduce the problem size for smaller instances. Another three-phase iterative heuristic was introduced by *Li et al. (2019)*. Authors constructed the initial solution by employing a two-phase iterative heuristic in the first phase of the algorithm by iteratively solving special lot-sizing problem and a series of VRP decompositions. This initial solution may be infeasible, hence in the second phase, infeasibility was repaired, if necessary, by solving the limited version of the main model and also series of VRP. A fix and optimize (FO) method was performed to improve the solution at the final phase. *Manousakis et al. (2022)* presented a two-phase infeasible space matheuristic for the generic PIRP. The first phase of the algorithm determined the production and distribution plan by relaxing and solving the problem. In the second phase, a local search that oscillates between the infeasible and feasible search space was executed. *Vadseth et al. (2021)* studied an iterative matheuristic algorithm for the IRP under maximum level policy. In the algorithm they proposed, giant tour was first created, and simple heuristics were applied to create feasible routes that are part of the path-flow formulation. Algorithm iterates over solving a path-flow model with a small set of routes then improves the routes based on the optimal solution in the prior iterations. Algorithm then terminates after reaching the maximum number of iterations. *Fortes et al. (2022)* presented a rich production-routing problem which differs from the standard PRP. In the rich PRP, they included a heterogeneous fleet with maximum riding time, and they allowed back-orders to meet unfulfilled demands at penalty, in addition to limited production and inventory capacities. Then they proposed a three hybrid two-level decomposition approaches where top-down strategy devised. *Wang and Chu (2022)* studied the classical IRP. To tackle the IRP, they proposed a two-phase decomposition-based heuristic where they combine logic based on benders decomposition and restricted mixed integer linear programming (MILP) approach in different phases. *Chekoubi et al. (2022)* proposed a two-phase decomposition heuristic to solve the integrated production-inventory-routing problem with remanufacturing (IPIRP-R) of end-of-life products especially in closed loop supply chains. Their approach achieved a good performance regarding small and medium instances. *Torkaman et al.*

(2022) addressed a PRP with price dependent demand and they formulated the problem as a mixed-integer nonlinear program (MINLP). They tackled the problem with an outer approximation (OA) approach where they generated and solved primal and master problems iteratively. They proposed four different rolling horizon-based heuristics for the computationally expensive master problem. *Ben Ahmed et al. (2023)* performed an extensive computational study to evaluate three different mathematical formulations for the standard PRP on the basic matheuristic they proposed. Their matheuristic consists of solving a sequence of constructive heuristics and improving the results by solving a mixed integer programming formulation iteratively. They underlined the importance of the initial solutions for the success of their matheuristic, in addition to the formulations they used.

2.4 Metaheuristic Algorithms for PIRP and IRP

In this section, metaheuristic algorithms for PIRP and IRP are extensively reviewed. Since metaheuristic solution methods to PIRP have received relatively less attention and not many studies exist in this context, studies addressed IRP are also comprehensively discussed.

Gen and Syarif (2005) investigated the production-distribution problem that integrates facility location, inventory and distribution decisions for multiple products where routing decisions are neglected. They developed a technique called hybridized spanning tree-based genetic algorithm (hst-GA). What distinguishes this approach from the traditional GA is that researchers utilize the Prüfer number representation for chromosomes based on a spanning tree which is capable of uniquely representing all possible trees in the network. At each iteration, they use a fuzzy logic controller (*Wang et al. 1997*) that works for fine-tuning of the crossover and mutation ratio parameters within a GA framework. *Boudia et al. (2007)* presented a mixed-integer linear model for production-distribution problem and developed a reactive greedy randomized adaptive search (GRASP) to solve the model. A memetic algorithm with population management (MA-PM) was investigated by *Bouida and Prins (2009)* for the production-distribution problem. Memetic algorithm was basically an enhanced version of the GA by some local

search procedures (*Moscata et al. 1999*). In their algorithm, production plan was created by a simple heuristic that equalizes the production quantity in the plant to total demand in each period. Then, distribution plan was generated by a saving heuristic and production plan was modified with a *Wagner-Whitin* algorithm. Offsprings were then generated with the crossover operator. Local search was performed in the final step which improves the existing offsprings with consideration of the population management that applied a selection criterion with a predetermined threshold value. *Abdelmaguid and Dessouky (2006)* proposed a GA for the integrated inventory distribution problem where production decisions were omitted, and transportation costs were determined with a non-linear function. Construction of the initial solution was made by an approximated transportation costs heuristic which mostly focused on optimal inventory allocations. Improvement procedure was applied to all chromosomes by genetic operators. Vehicle capacity violation was allowed during crossover and mutation parts, thus fixing procedures were applied once infeasible solutions arise. Mutation aimed to achieve partial deliveries by exchanging moves which help improving the overall cost. *Bard and Nananukul (2009)* designed a tabu search, introduced by *Glover (1989)* for PIRP. Reactive tabu list was employed with a variable size. Initial solution was constructed by solving the model without the routing variables. They calculated the approximated cost of transportation by assuming round trips, then they improved the routing phase by a subroutine with a tabu search with following simple local search procedure based on swap or transfer moves. Obtained solutions were stored in the reactive tabu list where infeasible solutions were prohibited. *Armentano et al. (2011)* investigated a multi-product PIRP and presented a mixed-integer programming model. They also developed a tabu search combined with path-relinking as a solution approach. *Fahimnia et al. (2012)* addressed a mixed-integer non-linear formulation for a two-echelon production-distribution problem and their objective was to maximize the total profit. They employed a genetic algorithm to solve the developed model. *Diabat et al. (2014)* addressed a tabu search algorithm that is more suitable for their proposed arc-based formulation for the periodic IRP with perishable products. Their algorithm starts with construction of an initial solution for the periodic distribution inventory problem, neighbor generation and evaluation procedures are executed iteratively in the tabu search framework. *Adulyasak and Coerdau (2014b)* introduced an optimization-based adaptive large neighborhood search (ALNS) heuristic for the PIRP in which binary variables are handled in the upper-level enumeration scheme

while continuous variables are solved within the network flow sub model. *Mirzaei and Sefi (2015)* considered an IRP involving perishable items that may end up in a lost sale within discrete time horizon. Age of the items determines the lost sale cost with both linear and exponential functions in their study. *Qiu et al. (2018)* studied the classic two-stage PIRP and developed a variable neighborhood search as a metaheuristic algorithm.

The pollution location routing problem (PLIRP) is introduced first time by *Karakostas et al. (2020)*, they addressed a mixed-integer linear programming that determines the fuel consumption over transportation phase besides decision of opening depots. They solved the problem with general variable neighborhood search (GVNS) based metaheuristic. IRP with perishable products where greenhouse gas emissions occurred in the routing phase was tackled by *Alkaabneh et al. (2020)*. They focused on minimizing the total cost which also considers the age-dependent price of the product and the inventory holding costs. Quality decays a unit within a discrete planning horizon. *Luo et al. (2023)* studied a bi-objective production-inventory-distribution problem minimizing the total tardiness and energy consumption in which production decisions are tackled in a homogeneous flowshop environment and distribution decisions are made by a three-party logistics provider. They proposed a modified NSGA-II including different local search techniques.

Table 2.1 summarizes reviewed papers in terms of problem characteristics of both PIRP and IRP studies, perishability consideration, demand type, planning horizon being discrete or continuous and solution method proposed.

Table 2.1: Literature Review

Study	Problem Characteristic										Perishability	Demand	Planning Horizon	Algorithm	
	SC	Production			Inventory			Distribution							
	Multi-echelon	N. Plant	Product	Cap.	Planning	Scheduling	Waste/Trans./Backorder	Cap.	Location	Policy	Routing	Fluct.	N. Yrns.	Cap.	Time Windows
Chandra (1993)	-	Single	Multiple	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Chandra and Fisher (1994)	-	Single	Multiple	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Fumero and Vercellis (1999)	-	Single	Multiple	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Gen and Syarif (2005)	-	Multiple	Multiple	+	+	-	-	-	-	ML	-	Unlimited	+	-	-
Lei et al. (2006)	-	Multiple	Single	+	+	-	-	-	-	ML	+	Het.	Unlimited	+	-
Bouda et al. (2006)	-	Single	Single	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Bouda and Pons (2007)	-	Single	Single	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Abdelmaguid and Dessouky (2007)	-	Single	Single	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Gelman and Laporte (2008)	-	Single	Single	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Chen et al. (2009)	-	Single	Multiple	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Bard and Nananukul (2009)	-	Single	Single	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Bard and Nananukul (2010)	-	Single	Single	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Archetti et al. (2011)	+	Single	Single	-	+	-	-	-	-	ML/OU	+	Hom.	Single	+	-
Farahani et al. (2011)	-	Single	Multiple	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Armentano et al. (2011)	-	Single	Multiple	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Rong et al. (2011)	+	Multiple	Multiple	+	+	-	-	-	-	ML	-	Unlimited	+	-	-
Mele and Wotley et al. (2012)	+	Multiple	Multiple	+	+	-	-	-	-	ML	-	Hom.	Unlimited	+	-
Fahimnia et al. (2012)	-	Single	Single	-	+	-	-	-	-	ML/OU	+	Hom.	Unlimited	+	-
Lee et al. (2013)	+	Multiple	Multiple	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Abbi et al. (2014)	-	Single	Single	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Adulyasak and Cordeau (2014a)	-	Single	Single	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Adulyasak and Cordeau (2014b)	-	Single	Single	+	+	-	-	-	-	ML/OU	+	Hom.	Unlimited	+	-
Diabat and Salem (2014)	+	Multiple	Single	+	+	-	-	-	-	ML	-	Hom.	Unlimited	+	-
Jia et al. (2014)	+	Single	Single	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Coelho and Laporte (2014)	+	Single	Single	+	+	-	-	-	-	FFOFOP/ML	+	Hom.	Unlimited	+	-
Moin and Willaen (2015)	+	Single	Single	+	+	-	-	-	-	ML/OU	+	Hom.	Multiple	+	-
Mirzazadeh and Seifi (2015)	-	Single	Single	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Bedi-Filho et al. (2015)	-	Single	Multiple	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Adulyasak et al. (2015a)	-	Single	Single	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Adulyasak and Cordeau (2015b)	+	Single	Single	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Diabat and Deshpande (2015)	+	Multiple	Single	-	+	-	-	-	-	ML	-	Hom.	Unlimited	+	-
Soyal et al. (2015)	-	Single	Single	-	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Kumar et al. (2016)	-	Single	Single	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Lu et al. (2016)	-	Single	Single	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Gu et al. (2016)	+	Single	Single	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Brahimi and Aouam (2016)	+	Single	Single	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Hasan et al. (2017)	+	Single	Multiple	-	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Soyal and Sural (2017)	+	Single	Single	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Rahimi et al. (2017)	-	Single	Multiple	-	+	-	-	-	-	ML	+	Het.	Unlimited	+	-
Azadeh et al. (2017)	-	Single	Single	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Gu et al. (2018)	-	Single	Single	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Crama et al. (2018)	-	Single	Single	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Miranda et al. (2018)	-	Single	Single	+	+	-	-	-	-	ML	+	Het.	Unlimited	+	-
Darvish et al. (2018)	+	Multiple	Multiple	-	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Qu et al. (2019)	+	Single	Single	+	+	-	-	-	-	8 different	+	Hom.	Unlimited	+	-
Naves-Moreira et al. (2019)	+	Single	Multiple	+	+	-	-	-	-	ML	+	Het.	Unlimited	+	-
Rohmer et al. (2019)	+	Single	Single	+	+	-	-	-	-	ML	+	Het.	Unlimited	+	-
Chizaz et al. (2019)	+	Single	Single	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Aci and Yildiz (2019)	+	Single	Single	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Chu et al. (2019)	+	Single	Single	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Aci and Yildiz (2020)	-	Single	Single	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Torres et al. (2020)	-	Single	Single	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Karakostas et al. (2020)	+	Multiple	Single	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Alkhabbeh et al. (2020)	-	Single	Single	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Alvarez et al. (2020)	+	Single	Single	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Alvarez et al. (2020)	+	Single	Single	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Li et al. (2020)	+	Single	Single	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Schnekenberg et al. (2021)	+	Multiple	Multiple	+	+	-	-	-	-	OU/ML	+	Hom.	Unlimited	+	-
Manousaki et al. (2021)	-	Single	Single	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Nasr et al. (2021)	+	Single	Multiple	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Qu et al. (2021)	+	Single	Single	+	+	-	-	-	-	ML	+	Het.	Unlimited	+	-
Vasanth et al. (2021)	+	Single	Single	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Mousavi et al. (2022)	+	Single	Single	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Alvarez et al. (2022)	+	Single	Single	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Forbes et al. (2022)	-	Single	Single	+	+	-	-	-	-	ML	+	Het.	Unlimited	+	-
Wang and Chu (2022)	-	Single	Single	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Cheikouh et al. (2022)	-	Single	Multiple	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Torres et al. (2022)	-	Single	Single	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Manousaki et al. (2023)	+	Single	Single	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Ben Ahmed et al. (2023)	-	Single	Single	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Lu et al. (2023)	+	Single	Single	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-
Mini-Study	+	Single	Single	+	+	-	-	-	-	ML	+	Hom.	Unlimited	+	-

Chapter 3:

PROBLEM DEFINITION

In this chapter, Perishable Production Inventory Routing Problem is formally introduced and illustrated. Then a mathematical model for the problem is presented.

3.1 Problem Statement

Consider a complete graph $G = \{N, E\}$ with a set of vertices $N = \{0, 1, 2, \dots, n\}$ and a set of edges $E = \{(i, j): i, j \in N, i \neq j\}$. A single plant with limited production and storage capacities is located at vertex 0. It has a fleet of homogeneous vehicles $K = \{1, 2, \dots, |K|\}$, each having a capacity V . A set of retailers $R = \{1, 2, \dots, n\}$ with a limited storage capacity are geographically dispersed at vertices $\{1, \dots, n\}$ and each retailer has a demand quantity through a time horizon $T = \{1, 2, \dots, |T|\}$. The demand at each retailer is assumed to be deterministic and time varying. In this setting, a perishable production inventory routing problem (PPIRP) consists of simultaneously determining production, storage, delivery and route planning to satisfy retailer demand while considering perishability quality of the products. Perishability quality of the products can be defined by a set $L = \{0, 1, 2, \dots, |L|\}$ where 0 represents the freshest level. Objective is to minimize the total cost consisting of all the stages. For each period, following decisions are to be made: 1) how much to produce at the plant; 2) how much to replenish at each retailer; 3) how to determine the routes of the planned deliveries to retailers; 4) how to determine the quality of the fulfillments of the customer demands at retailers.

The study is conducted under the following assumptions: 1) Each identical vehicle is allowed to have only one tour which starts and ends at the plant during a period. 2) Split delivery is not allowed in a period. 3) Product quality decreases one unit after each period. Once the quality goes beyond $|L|$, product can no longer be used to fulfill customer demand. 4) The total demand in the same route cannot exceed vehicle capacity. 5) Total inventory at each node cannot exceed maximum inventory level in each period. 6) Plant act as a central decision maker. 7) Retailer demands in each period are known at the beginning of the planning horizon. 8) Only one type of product is produced and distributed. The first assumption is characteristic for the vehicle routing problems, and it

is commonplace in PIRP. Second assumption is supported by the fact that daily operations of the retailers are disturbed as the delivery number increases. Third assumption is realistic in real life, and it is related with product's characteristic especially in perishable food logistics. Sixth assumption remarks that PPIRP in this thesis consists of a two-echelon problem where plant decides delivery qualities for final customers who receive products from retailers that plant distributes. Eighth assumption specifies that the model presented here is applicable for a single product, however, note that it can be easily converted into multi-product. The rest of the assumptions are substantially similar with traditional inventory routing problem. Network representation of PPIRP is shown in Figure 3.1.

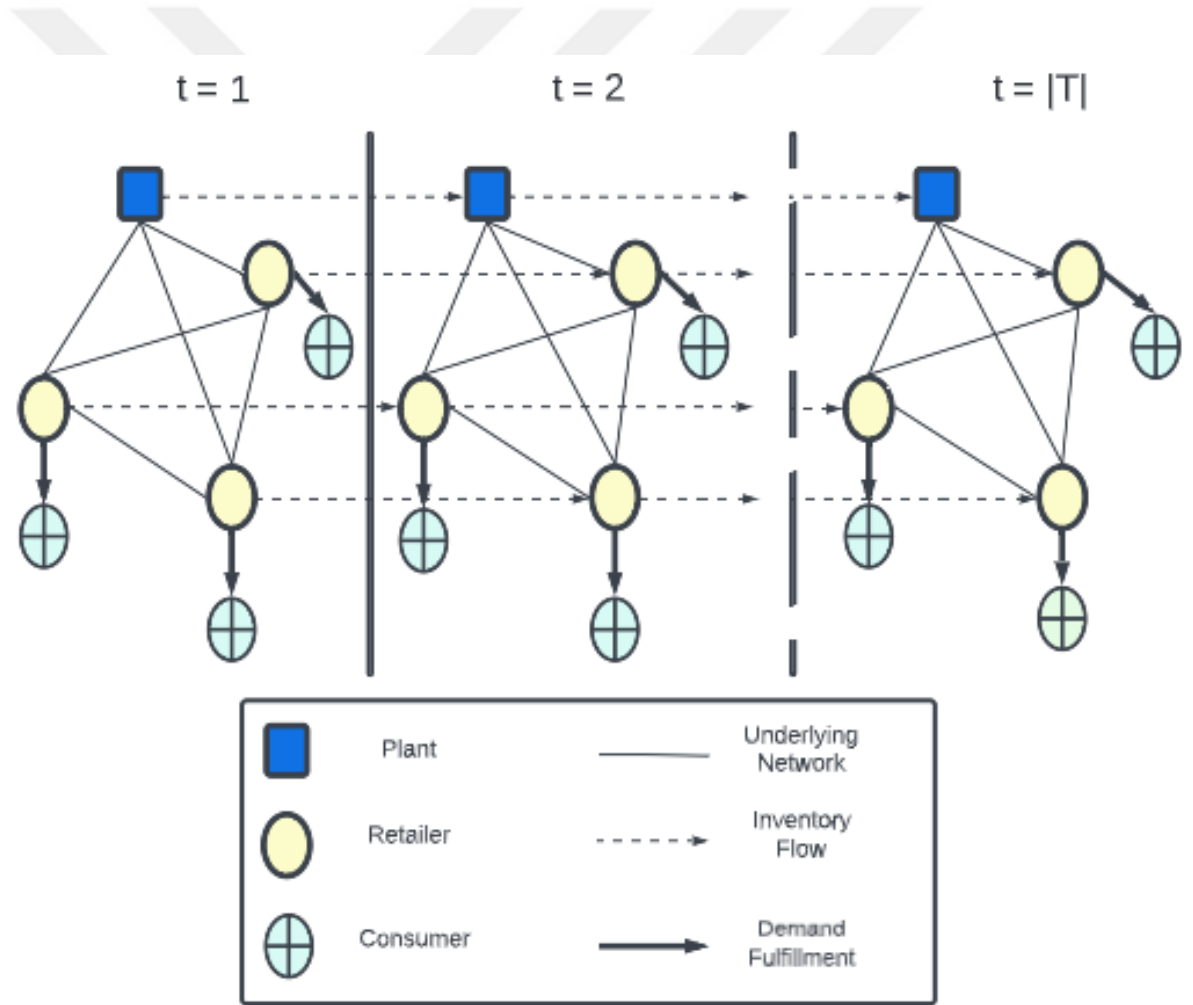


Figure 3. 1: Network Representation of PPIRP

3.2 Mathematical Model

In this section, parameters and decision variables used in the mathematical model are defined. Then a mixed integer linear programming (MILP) model for PPIRP is presented.

Sets

N :	Set of nodes $\{0, 1, 2, \dots, n\}$
R :	Set of retailers $\{1, 2, \dots, n\}$
E :	Set of edges $\{(i, j): i, j \in N, i \neq j\}$
T :	Set of time periods $\{1, 2, \dots, T \}$
L :	Set of quality levels of the product $\{0, 1, 2, \dots, L \}$
K :	Set of vehicles $\{1, 2, \dots, K \}$

Parameters

e_t :	Unit production cost in period $t \in T$
f_t :	Fixed set-up cost in period $t \in T$
h_i^l :	Holding cost per unit with quality level $l \in L$ at node $i \in N$
c_{ij} :	Travel cost between nodes $i \in N$ and $j \in N$
D_i^t :	Demand quantity at retailer $i \in R$ in period $t \in T$
g_i^l :	Initial inventory at node $i \in N$ with quality level $l \in L$
C :	Production capacity of the plant
U_i :	Maximum inventory level at node $i \in N$
V :	Vehicle capacity

Decision Variables

p_t :	Production quantity in period $t \in T$
I_i^{lt} :	Inventory level with quality level $l \in L$ at node $i \in N$ in period $t \in T$
q_i^{lkt} :	Quantity delivered to retailer $i \in R$ by vehicle $k \in K$ with quality level $l \in L$ in period $t \in T$
d_i^{lt} :	Quantity fulfilling demand of retailer $i \in R$ with quality level $l \in L$ in period $t \in T$

$$\begin{aligned}
w_t: & \begin{cases} 1, \text{ if there is a production set-up performed at the plant in period } t \in T, \\ 0 \text{ otherwise} \end{cases} \\
x_{ij}^{kt}: & \begin{cases} 1, \text{ if edge } (i, j) \in E \text{ is traversed by vehicle } k \in K \text{ in period } t \in T, \\ 0 \text{ otherwise} \end{cases} \\
z_i^{kt}: & \begin{cases} 1, \text{ if vehicle } k \in K \text{ visits node } i \in N \text{ in period } t \in T, \\ 0 \text{ otherwise} \end{cases}
\end{aligned}$$

Formulation of the PPIRP with the above notations as a mixed integer linear mathematical model is as follows:

$$\text{Min } \sum_{t \in T} (e_t p_t + f_t w_t) + \sum_{i \in N} \sum_{l \in L} \sum_{t \in T} h_i^l I_i^{lt} + \sum_{(i,j) \in E} \sum_{k \in K} \sum_{t \in T} c_{ij} x_{ij}^{kt} \quad (1)$$

$$\text{s.t. } I_i^{l0} = g_i^l, \quad \forall i \in N, l \in L \quad (2)$$

$$I_0^{lt} = I_0^{l-1, t-1} - \sum_{i \in R} \sum_{k \in K} q_i^{lkt}, \quad \forall l \in L \setminus \{0\}, t \in T \quad (3)$$

$$I_0^{0t} = p_t - \sum_{i \in R} \sum_{k \in K} q_i^{0kt}, \quad \forall t \in T \quad (4)$$

$$I_i^{lt} = I_i^{l-1, t-1} + \sum_{k \in K} q_i^{lkt} - d_i^{lt}, \quad \forall i \in R, l \in L \setminus \{0\}, t \in T \quad (5)$$

$$I_i^{0t} = \sum_{k \in K} q_i^{0kt} - d_i^{0t}, \quad \forall i \in R, t \in T \quad (6)$$

$$p_t \leq C w_t, \quad \forall t \in T \quad (7)$$

$$\sum_{l \in L} I_i^{l, t-1} + \sum_{l \in L} \sum_{k \in K} q_i^{lkt} \leq U_i, \quad \forall i \in R, t \in T \quad (8)$$

$$\sum_{l \in L} I_0^{lt} \leq U_0, \quad \forall t \in T \quad (9)$$

$$\sum_{l \in L} d_i^{lt} = D_i^t, \quad \forall i \in R, t \in T \quad (10)$$

$$\sum_{i \in R} \sum_{l \in L} q_i^{lkt} \leq V z_0^{kt}, \quad \forall k \in K, t \in T \quad (11)$$

$$\sum_{l \in L} q_i^{lkt} \leq V z_i^{kt}, \quad \forall i \in R, k \in K, t \in T \quad (12)$$

$$\sum_{k \in K} z_i^{kt} \leq 1, \quad \forall i \in R, t \in T \quad (13)$$

$$\sum_{j \in N, j \neq i} x_{ij}^{kt} + \sum_{j \in N, j \neq i} x_{ji}^{kt} = 2 z_i^{kt}, \quad \forall i \in N, k \in K, t \in T \quad (14)$$

$$\sum_{j \in R} x_{0j}^{kt} \leq 1, \quad \forall k \in K, t \in T \quad (15)$$

$$\sum_{i \in S} \sum_{j \in S: i \neq j} x_{ij}^{kt} \leq \sum_{i \in S} z_i^{kt} - z_s^{kt}, \quad \forall S \in R, |S| \geq 2, k \in K, t \in T, s \in S \quad (16)$$

$$(p_t, I_i^{lt}, q_i^{lkt}, d_i^{lt}) \geq 0, \quad \forall (i, j) \in E, k \in K, t \in T, l \in L \quad (17)$$

$$(w_t, x_{ij}^{kt}, z_i^{kt}) \in \{0, 1\}, \quad \forall (i, j) \in E, k \in K, t \in T, l \in L \quad (18)$$

Objective function (1) minimizes the total operational cost that comprises of production cost, set-up cost, inventory holding cost and routing cost respectively. Inventory levels are initialized by Constraint (2). Constraints (3)-(4) are the inventory balance equations for the plant. Similarly, Constraints (5)-(6) impose the inventory balance constraints for the retailers. Production and inventory capacity restrictions are enforced by Constraints (7)-(9). Constraint (10) ensures that customer demands at each retailer must be exactly satisfied. Constraint (11) defines the vehicle capacity constraint. Constraints (12) ensure that each retailer can get the delivery only if it is visited. Split delivery is forbidden to a retailer in a period by Constraint (13). Flow conversations are imposed by Constraint (14). Constraint (15) enforces that each vehicle can have at most one tour in a period. Constraint (16) eliminates subtours that can occur in a tour. Constraints (17)-(18) define the variables and their domains.

In addition to the constraints above, following strengthening valid inequalities are useful to cut off infeasible solutions and break symmetric situations. Even though inequalities are redundant for our problem definition, they can help the solver to generate new cuts:

$$z_i^{kt} \leq z_0^{kt}, \quad \forall i \in R, k \in K, t \in T \quad (19)$$

$$x_{ij}^{kt} \leq z_i^{kt}, \quad \forall (i, j) \in E, k \in K, t \in T \quad (20)$$

$$x_{ji}^{kt} \leq z_i^{kt}, \quad \forall (i, j) \in E, k \in K, t \in T \quad (21)$$

Constraint (19) enforces the situation that if any customer i is visited by vehicle k in period t , then the plant must be visited with the same vehicle. Constraints (20)-(21) ensure that if customer i is the successor or the predecessor of customer j in the route of vehicle k in period t , then customer i must be visited by the same vehicle.

It is relevant to note that similar modelling approaches are taken by *Coelho and Laporte (2014)* and *Alvarez et al. (2022)*, when perishability of products is considered within a two-echelon supply chain. In their studies, the way perishability modelled is the same with this study as the quality of the products are age-dependent which spoils after a certain number of periods. *Coelho and Laporte (2014)* studied a perishable inventory

routing problem where production decisions are omitted. From this perspective, Constraints (2), (4) and (7) are the main extensions to the study of *Coelho and Laporte (2014)*, by adding the production dimension. In addition, the objective was to maximize the total net profit. For the rest, our model is very similar except for minor details. On the other hand, model used in the study of *Alvarez et al. (2022)* is also similar apart from the fact that they modelled the perishable production inventory routing problem with transshipments between retailers were allowed with a penalty in the objective function. Also, those transshipments quantities are involved in all the inventory constraints which differs the model's tendency on determining production and inventory levels clearly from ours. In addition to that, their model also maximizes the overall net profit.

Considering the age-dependent perishability modelling, our model can be adapted to study other aspects occurring in the environment of age-dependent quality. Unlike the single perishable product in our case, other products which has a higher quality over time, can easily be adapted. Also, it is common in the industry to price each quality level of the products. Highest quality products may be considered as the highest priced and reflected in the objective function easily, whereas in our model, it does not matter for the final consumer as long as products are delivered without being perished.

Chapter 4:

SOLUTION METHODOLOGY

Since considered PPIRP is NP-Hard, medium and large size instances cannot be solved to optimality by commercial solvers. Also, other exact methods are not effective for this variant of the PPIRP in terms of run times and practicality. Furthermore, when we use CPLEX to solve the mathematical model presented in Chapter 3, we were not able to get any further feasible solutions for instances larger than those that include 10 nodes and 10 periods. It is also relevant to note that there has not been any study performed for designing a metaheuristic for this variant of the PPIRP to the best of our knowledge.

In this chapter, our Hybrid Genetic Algorithm (HGA) is explained elaborately. Then, adaptation of the two-phase iterative heuristic (TIH) approach of *Absi et. al (2014)* is introduced as a benchmark algorithm for our HGA to be able to compare performances of the algorithms in addition to model results.

4.1 Hybrid Genetic Algorithm

Evolutionary algorithms have gained popularity for solving NP-Hard combinatorial optimization problems in the last decades. These algorithms are inspired from the biological processes in the nature. Genetic algorithm (GA) is one of the most popular metaheuristics in the class of evolutionary algorithms. It is commonly used to get high quality solutions in a reasonable time. As GA is a population-based algorithm, a population comprising chromosomes with various types of solution characteristics carries good characteristics into next generations. Fitter individuals have higher chances to be evolved through genetic operators such as crossover and mutation. Until the stopping criterion is met, this process repeats itself. Considered PPIRP has so many complexities such that it is not viable to solve with a simple and traditional GA. A hybrid GA approach (HGA), which is combined with an additional mathematical programming components and local search techniques, is designed for PPIRP.

In the following, general structure of the proposed HGA, chromosome representation, encoding and decoding mechanisms together with local search techniques and mathematical programming, repairing mechanisms for infeasible chromosomes, fitness function, initial population construction, genetic operators, selection processes and stopping criteria are discussed.

4.1.1 General structure of the HGA

In this section, the general overview of HGA is discussed briefly. Since PPIRP includes many decisions regarding production, inventory replenishment and transportation, it is not a trivial task to represent the solution in a simple way, hence representation must carry a key information that can easily be converted into a complete solution. The basic idea is to reduce the complexity of the problem by decomposing these different layers of decisions into several sub problems. Thus, chromosome representation for the proposed HGA displays the visit and set-up decisions simultaneously. Once visit and set-up plans are obtained, determining rest of the decisions, which are production, inventory, delivery in addition to routing decisions, is much easier. Details of this decoding process is explained in the following section. We underline that since this decoding process contains mathematical programming model and a tabu search algorithm, we call our algorithm as a hybrid heuristic over a genetic algorithm framework. In the remainder of this section, first the general flow of the algorithm is discussed briefly, then in the following section, all the details are elaborately explained. The general framework of the HGA is shown in Figure 4.1.

In the first stage of the proposed HGA, the initial population is generated by a random greedy construction heuristic which randomly determines a feasible and random different visit and set up configurations. The purpose of randomly generating different solutions as many as population size P is to explore different parts of the search space and to avoid being trapped in a local optima within the starting phase. Since infeasible individuals are not allowed within the population, this initialization process is performed until a full feasible population is obtained. In the next stage of the algorithm, the evolutionary process of the GA is continued, incorporating a cyclical process of generating new populations until the termination condition is satisfied. Moreover, after the initial populat-

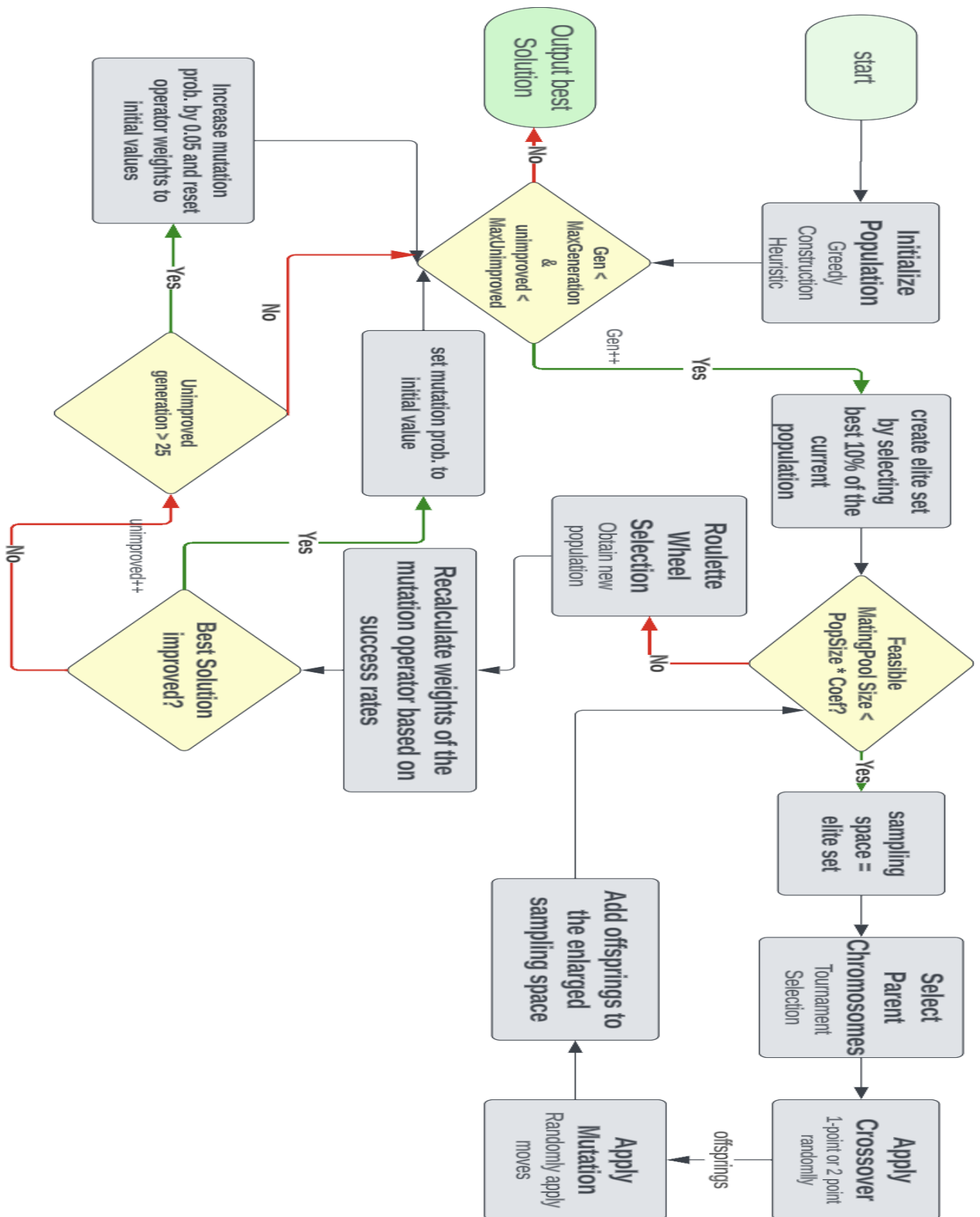


Figure 4. 1: Framework of HGA

ion is created, new offsprings with different characteristics of the existing chromosomes are produced by successive selection operators, crossover, mutation and an improvement phase. An elitist approach is followed when generating a new population, consisting of directly adding the m best chromosomes in the current population. The rest of the population, $P-m$ chromosomes, is generated by genetic operations. These $P-m$ chromosomes are selected out of the enlarged sampling space which has a size of $\beta \times P$ newly generated offsprings. After having the enlarged sampling space, $P-m$ chromosomes are selected by a roulette wheel selection which is a stochastic selection process in which fitter individuals have a higher chance to be selected. Another selection approach is made for the parents that produces the offsprings together with the genetic operators, crossover and mutation, respectively. In each generation, to apply the genetic operators, two parents are selected with a binary tournament selection which randomly selects two chromosomes from the current population and finally the fitter one is chosen as a parent. Crossover and mutation are extremely significant for both satisfying intensification and diversification mechanisms in the algorithm, as explained in detail in Sections 4.1.8 and 4.1.9. Another significant element of HGA to ensure diversification mechanism and to avoid fast convergence within the cyclical process is to increase the mutation probability by 0.05 if there have been 25 iterations without improvement. Based on our computational experiences, it was observed that this methodology helps finding different parts of the search space to overcome local optima. Then mutation rates are reestablished to the initial value as the best solution is improved again. The process repeats itself until either the maximum number of generations or the maximum number of generations without improvement is reached. Algorithm 1 describes the detailed procedure of the HGA.

4.1.2 Chromosome Representation

Encoding the chromosomes is a crucial part of any genetic algorithm since it is critical to find an appropriate solution representation that can easily be decoded to complete feasible solution. Each representation should carry all the necessary information of a solution. Furthermore, finding an appropriate and efficient representation for the chromosome is not a trivial task in such a complex problem as PPIRP. Encoding and decoding schemes are explained in the following subsections.

Algorithm 1 – Hybrid Genetic Algorithm

```

1. input: PopSize,  $P_c$ ,  $P_m$ , eliteSize,  $\beta$ , MaxGeneration, MaxUnimproved,  $\alpha$ ,  $\Omega$ 
2. Initial Population Construction:
3. Solve RandomGreedyConstruction() until PopSize feasible chromosomes obtained
4. Main Loop:
5. Initialize  $f_{best}$  = min. fitness in the initial population; gen = 0; unimproved = 0; unimproved2 = 0; BestSol = chromosome with  $f_{best}$ ;
6. while (gen < MaxGeneration and unimproved < MaxUnimproved) do
7.   S = {Best eliteSize chromosomes in the current population}
8.   while (|S| < PopSize x  $\beta$ ) do
9.     Select two parents with tournament selection;
10.    if ( rand[0,1] <  $P_c$ ) Apply Crossover to parents and get offsprings;
11.    for (each offspring or parent)
12.      if ( rand[0,1] <  $P_m$ )
13.        Select the mutation operator by weights ( $\Omega$ ) and update the performances;
14.        Apply Mutation to the offspring or parent;
15.        if (current fitness < previous fitness)
16.          Increment improvement number of the operator
17.        end if
18.      end if
19.    end for
20.    S = S U {Offsprings};
21.  end while
22. Obtain new population by roulette wheel mechanism (S) and update  $f_{best}$ 
23. gen=gen+1;
24. if ( gen > 5 and gen mod 5 = 0) update operator weights based on success rates;
25. if ( $f_{best}$  improved) set unimproved, unimproved2 = 0; set  $P_c$ ,  $P_m$ ,  $\alpha$  to initial values;
26. else unimproved = unimproved + 1; unimproved2 = unimproved2 + 1;
27. if (unimproved2>25)  $P_m$  = min{0.9,  $P_m$  + 0.05},  $\alpha$  =  $\alpha$  - 0.3; reset  $\Omega$  to initial values;
28. end while
29. output: BestSol

```

4.1.3 Encoding

It is already discussed in the previous sections that delivery decisions have an important role for the PPIRP, because delivery decisions are the key variables providing the transition between two stages of the problem. In other words, pre-determining delivery decisions significantly helps obtaining a full solution. Once we acquire a feasible delivery plan, we have a promising starting point to get all production, inventory and distribution related decision variables by utilizing a linear programming mathematical model and fast running local search techniques.

To obtain such a feasible delivery plan we have a chromosome representation which consists of two layers: each gene in the first layer represents the visit variables, z_i^t denoting a binary decision whether node $i \in N$ in period $t \in T$ is visited or not, respectively. Different than the original visit variable stated in the main model, vehicle index k is dropped instead of including those decisions in the genetic representation, because a simple and fast running tabu search is able to generate higher-quality routing decisions which makes the process less challenging and complicated. So, the first layer called visit layer has a size of $(n + 1) \times |T|$ matrix. The second layer of the genetic representation incorporates set-up variables, w_t , which states whether a production takes place in period t . This layer is represented by a single row which has a length of $|T|$. Together with the second layer, genetic search is conducted comprehensively where visit and production decisions are evaluated in a simultaneous approach. Finally, a matrix of size $(n + 2) \times |T|$ is used for representing the solution, as shown in Figure 4.2.

4.1.4 Decoding

In addition to gene values, z_i^t and w_t , to obtain a complete feasible solution, rest of the continuous variables, $p_t, I_i^{lt}, q_i^{lt}, d_i^{lt}$ that denote production, inventory, delivery and consumption decisions, in addition to routing decisions, x_{ij}^{kt} , need to be determined, respectively. To determine the values of production, inventory, delivery and consumption variables, $p_t, I_i^{lt}, q_i^{lt}, d_i^{lt}$, a linear programming (LP) model is solved, which takes, z_i^t and w_t as its input, in the first stage of decoding. Then, having, z_i^t , and delivery quantities, q_i^{lt} , as the output of the LP, they are fed into a tabu search algorithm for the VRP as the

second stage of decoding. The details of the tabu search are explained in *Section 4.2.2.1*. In the remainder of this section, LP model, denoted as $LP(z, w, \lambda)$, is introduced. Then, according to the LP results, some of the gene values, z_i^t and w_t , might need to be adjusted (i.e., when $w_t = 1$ and $p_t = 0$), hence, this adjustment process, named as trimming process, is discussed. Finally, obtaining the routing variables as a last stage is explained. Overall decoding framework is presented in Figure 4.3.

		periods		
		1	2	3
z_i^t	0	0	1	1
	1	0	0	1
	2	0	1	0
	3	0	1	0
	4	0	1	1
	5	0	1	0
	6	0	0	1
	7	0	1	0
	8	0	1	0
	9	0	1	1
	10	0	1	0
w^t	11	0	1	0

Figure 4. 2: Chromosome representation of the instance with retailer size of 10 and period of 3. First 11 row indicates the visit decisions for the plant and each retailer whereas the last row shows whether any production takes place for each period.

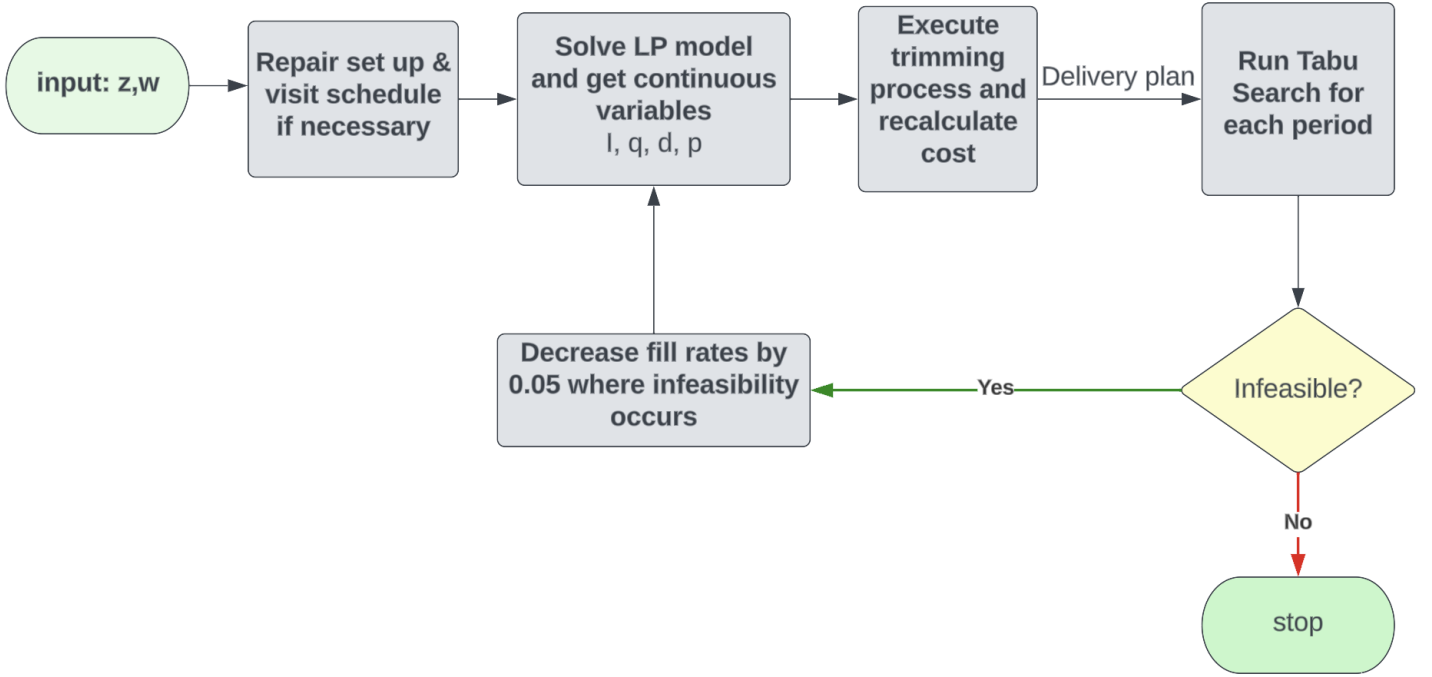


Figure 4. 3: Decoding Framework of a Chromosome

4.1.4.1 LP Model

Given the values of the visit and set up variables (z_i^t and w_t , respectively) for a given chromosome, rest of the continuous variable $p_t, I_i^{lt}, q_i^{lt}, d_i^{lt}$ (production quantities, inventory levels, delivery and consumption amounts without vehicle specification) can be found easily. To achieve this, the following LP formulation, $LP(z, w, \lambda)$, is developed, which is similar to one given in *Alvarez et al. (2022)*. Note that delivery quantities are aggregated over all fleet since routing or vehicle related decisions are determined in the next phase.

$$\text{Min } \sum_{t \in T} (p_t e_t + \sum_{i \in N} \sum_{l \in L} h_i^l I_i^{lt}) \quad (22)$$

$$\text{s.t. } (2), (9)-(10),$$

$$I_0^{lt} = I_0^{l-1, t-1} - \sum_{i \in R} q_i^{lt}, \quad \forall l \in L \setminus \{0\}, t \in T \quad (23)$$

$$I_0^{0t} = p_t - \sum_{i \in R} q_i^{0t}, \quad \forall t \in T \quad (24)$$

$$I_i^{lt} = I_i^{l-1, t-1} + q_i^{lt} - d_i^{lt}, \quad \forall i \in R, l \in L \setminus \{0\}, t \in T \quad (25)$$

$$I_i^{0t} = q_i^{0t} - d_i^{0t}, \quad \forall i \in R, t \in T \quad (26)$$

$$\sum_{l \in L} I_i^{l,t-1} + \sum_{l \in L} q_i^{lt} \leq U_i, \quad \forall i \in R, t \in T \quad (27)$$

$$p_t \leq C w^{*t}, \quad \forall t \in T \quad (28)$$

$$\sum_{i \in R} \sum_{l \in L} q_i^{lt} \leq \lambda^t V |K| z_0^{*t}, \quad \forall t \in T \quad (29)$$

$$\sum_{l \in L} q_i^{lt} \leq \min\{V, U_i\} z_i^{*t}, \quad \forall i \in R, t \in T \quad (30)$$

$$q_i^{lt}, I_i^{lt}, d_i^{lt}, p^t \geq 0, \quad \forall i \in R, l \in L, t \in T \quad (31)$$

Note that in this LP, given set-up and visit variables, w^t and z_i^t , are known and thus replaced by parameters indicated as w^{*t} and z_i^{*t} . To elaborate, objective function (22) minimizes the total unit production and overall inventory cost. Constraints (2), (9) and (10), are directly taken from the main model ensuring initial inventory levels, inventory capacities at plant, and satisfying retailer consumptions according to given demands, respectively. Constraints (23) – (27) impose inventory balance equations for both plant and retailers, as well as inventory capacity restriction for each retailer. The only difference between these constraints and the constraints in the original model is that vehicle index k is dropped from delivery quantity variable, q_i^{lkt} , since vehicle related decisions are disregarded in this LP. Yet, aggregated delivery amounts are used as an input in the next decoding phase. Constraint (28) enforces the maximum production amount for each period, while the total capacity is restricted by parameter w^{*t} . Constraint (29) dictates the total delivery amount in each period where the total fleet capacity is controlled by parameter z_0^{*t} which indicates whether there is any tour taken place in that period. In addition, the same approach used in TIH algorithm is utilized to avoid split delivery infeasibility in the VRP phase, which is to use fill rate parameter for each period, λ^t . Value of λ^t for each period $t \in T$ is initialized as 1.0. If tabu search cannot find a feasible routing plan afterwards, infeasible periods' fill rate will be decremented by 0.05 to achieve feasibility. Details are discussed in *Section 4.2.2* Concerning long computational time in decoding mechanism, this fill rate reduction is done only once. Constraint (30) has the same function with (12) of the main model where delivery amount to each retailer and in each period is restricted, since we know beforehand whether the retailer is visited or not at that period by parameter z_i^{*t} .

4.1.4.2 Trimming Process

After solving $LP(z, w, \lambda)$, with a commercial solver CPLEX, we remove or trim all empty visits and set ups that may remain after. This trimming process occurs for the empty set ups where $p_t = 0$ and $w^{*t} = 1$ and visits where $\sum_{l \in L} q_i^{lt} = 0$ and $z_i^{*t} = 1$. After trimming, decision variables z_i^t and w^t are revised and set up cost is recalculated for deducted set up values, accordingly. At the end, all the production and inventory cost in the objective function (1) is obtained.

4.1.4.3 Obtaining Routing Variables

The only remaining cost component and missing variables consist of routing part, which is not yet determined in the decoding mechanism, are calculated with the tabu search. Tabu search is executed exactly in the same way where we have aggregate delivery (i.e., $q_i^t = \sum_{l \in L} q_i^{lt}$) and visit plans (i.e., z_i^t) after solving $LP(z, w, \lambda)$. With the same approach in TIH, each period t with delivery is considered as a VRP sub problem to attain a complete solution of the original problem as a final step of decoding.

4.1.5 Fitness Function

Fitness value is the main indicator which specifies the survival evaluation of the chromosome in the population. Fitness function of each chromosome is calculated by the value of the main objective function (1). If the decoding mechanism leads to an infeasible solution, which is very rare, since infeasible solutions are also not allowed in the next generations of the algorithm, fitness is an extremely big number.

4.1.6 Infeasibility Repairing Mechanisms

In the proposed HGA, since decoding process comprises of solving an LP and a tabu search, repetitively decoding the same chromosome over and over may lead to large computational times. Hence, only obvious and easily detectable infeasibilities are repaired by simple moves on both set-up and visit schedule without obtaining the complete solution. Even though that does not necessarily mean there cannot be any type

of infeasibility in the next stage of decoding, our computational experiments have also shown that this greedy repairing approaches on visit and set-up schedules achieve feasibility with a great extent. In the following, these two types of procedures are discussed.

4.1.6.1 Repairing Infeasible Production Set Up Schedule

Greeditly repairing the set-up schedule is only done by a simple move, which checks whether the first period of production (i.e. $\arg \min_{t \in T} \{w_t = 1\}$) is performed after the minimum of these following two expressions: the first period of delivery, (i.e. $\arg \min_{t \in T} \{z_0^t = 1\}$) and the minimum period of required latest deliveries of retailers. First period of delivery is known by the representation, and the latest needed delivery period of each retailer, $latest_i$, can easily be calculated by the given parameters of demands and initial inventory levels, as formulated in the following:

$$latest_i = \arg \min_{t'} \{t' \in T \mid \sum_{l \in L} g_i^l \leq \sum_{t < t'} D_i^t\}, \quad \forall i \in R \quad (32)$$

Accordingly, the minimum required first production period is formulated as below:

$$MRP = \min\{\arg \min_{t \in T} \{z_0^t = 1\}, \min\{latest\}\} \quad (33)$$

Then, the following condition is checked to insert a new set up before the current first period of production: $\arg \min_{t \in T} \{w_t = 1\} < \min\{\arg \min_{t \in T} \{z_0^t = 1\}, latest\}$. If it holds then we add a new production set up into MRP (i.e. $w_{MRP} = 1$). Figure 4.4 (a) demonstrates an example of the production set up repairing procedure. The downside of this greedy approach is that it is only valid for instances where it is assumed that no initial inventory in the plant exists, yet, as it will be discussed in Section 5, Since the plant has no initial inventory in our instances, this greedy approach works well.

4.1.6.2 Repairing Infeasible Visit Schedule

Infeasible visit schedule is observed in the representation when there is no delivery for retailer i before the latest needed delivery period, $latest_i$ (i.e., Figure 4.4 (b)) or when there are more than $|L|$ consecutive periods without any delivery (i.e., Figure 4.4 (c)). More than $|L|$ consecutive periods without any delivery lead to a certain infeasibility

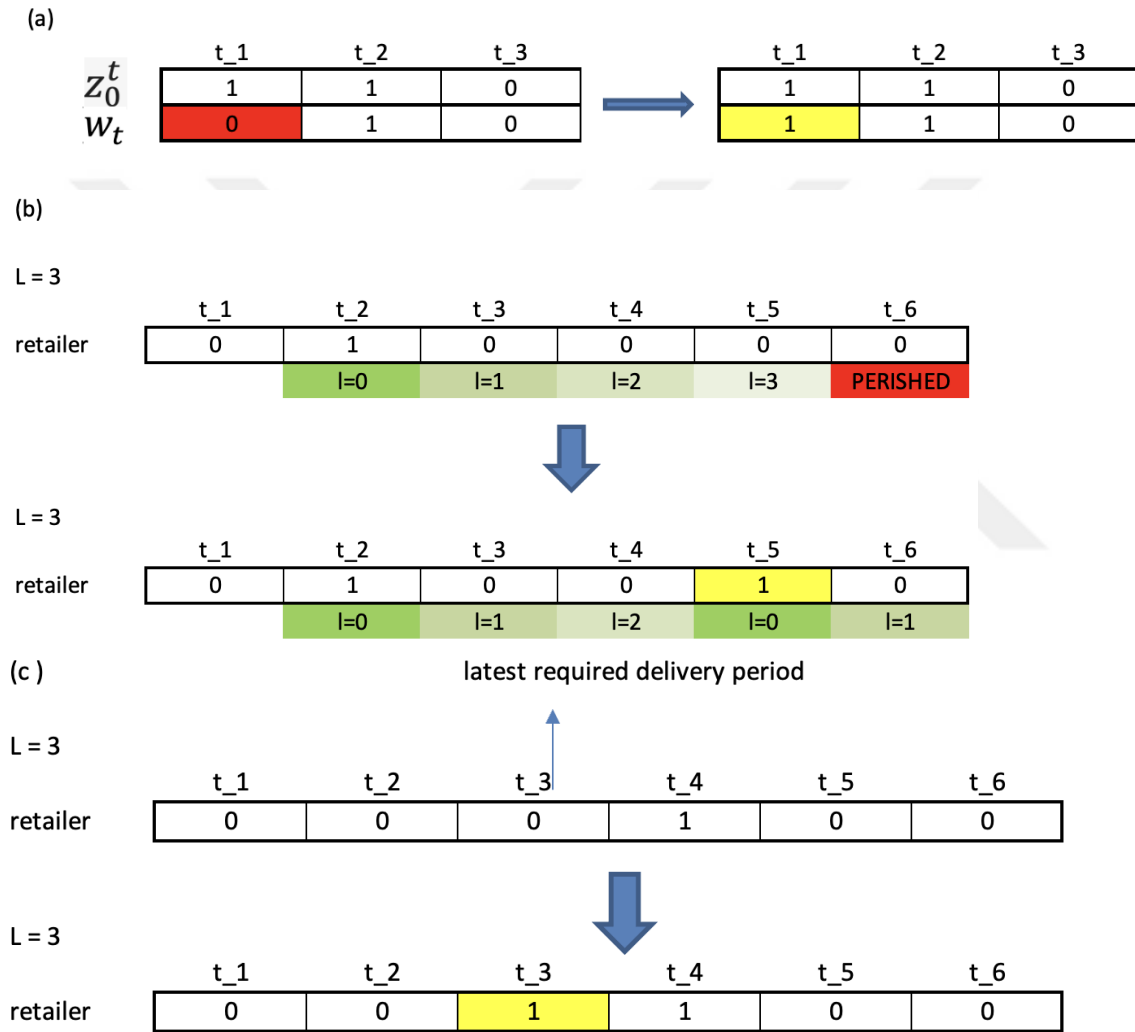


Figure 4.4: (a) Repairing of the production schedule where there is a tour before the first production period. (b) Repairing of the visit schedule where there are more than $|L|$ consecutive periods without delivery, (c) Repairing of the visit schedule where there is no single visit until the latest required period

considering there are demand quantities that must be fulfilled in each period. To elaborate, even though a retailer has highest quality products in the inventory, having more than $|L|$ consecutive periods without delivery, later causes a stock-out situation at the retailer due to having all products perished regardless of the quality level at the end of the $|L|^{th}$ consecutive period.

If any of the above two conditions holds, then new visits are randomly inserted. As a greedy algorithm, it does not guarantee to avoid infeasibility. However, based on our computational experience, there are only occasional cases seen after the decoding that repaired visit schedule ends up remaining infeasible.

4.1.7 Initial Population Construction

Initialization of the HGA is an important process where a set of randomized solutions are generated. Since solution space of PPIRP is extremely large, starting population of the HGA requires to contain diversified solutions, such that replicated or similar chromosomes must be avoided while generating the initial population. At the same time, fully random solutions may lead to difficulties in finding near optimal solutions in the future generations. Thus, to start with a quality and still randomized set of solutions, random greedy construction heuristic is implemented as shown in Algorithm 2.

Construction heuristic starts with building the set-up schedule first. The basic idea is that once the set-up schedule is constructed, visit schedule of each retailer can be constructed accordingly. On the other hand, predetermining the production periods helps giving replenishment decisions, avoiding excessive inventory costs and creating efficient partial delivery decisions that possibly reduces transportation costs more easily. Hence, the construction is initialized by random set-up moves within the time horizon. There is only one criterion that must be satisfied to keep away from infeasible set-up schedule, which is to have at least one period with production until the minimum required production period, MRP (line 7). Once the randomly designed schedule does not have any production before MRP , then a set-up is inserted directly into the period of

MRP (line 8). In addition, this process repeats itself until at least $|T|/3$ different periods of set-ups are obtained as a precaution to avoid production capacity issues.

Algorithm 2 – Random Greedy Construction

```

1. Set Up Construction:
2. Initialize setUpCount = 0;
3. while (setUpCount <  $|T|/3$ ) do
4.   for ( $t \in T$ )
5.      $w^t = \text{rand}[0, 1]$ ;
6.     setUpCount = setUpCount +  $w^t$ ;
7.     if ( $t = \text{MRP}$  and setUpCount = 0)
8.        $w^{\text{MRP}} = 1$ ;
9.       setUpCount = setUpCount + 1;
10.    end if
11.  end for
12. end while
13. Visit Construction:
14. Initialize visitCount, totalVisit, isFeasible;
15. for ( $i \in R$ )
16.   visitCount = 0;
17.   totalVisit =  $\text{rand} \left[ \left\lfloor \frac{|T|}{|L|} \right\rfloor, |T| - \text{argmin}_{t \in T} \{w_t = 1\} + 1 \right]$ ;
18.   isFeasible = false;
19.   while ( visitCount < totalVisit or !isFeasible) do
20.      $z_i^{\text{rand}[\text{argmin}_{t \in T} \{w_t = 1\}, |T|]} = 1$ ;
21.     visitCount = visitCount + 1;
22.     if (firstVisitPeriod > latest $_i$  OR
23.       more than  $|L|$  consecutive periods without delivery exists)
24.       isFeasible = false;
25.     else isFeasible = true;
26.     end if
27.   end while
28. end for
29. Assign plant visits  $z_0^{t \in T}$  according to  $z_{i \in R}^{t \in T}$  retailer visits;
30. output:  $z_i^t, w^t$ 

```

After obtaining a set up schedule, w^t , construction of visit schedule, z_i^t is executed retailer by retailer (line 15). For each retailer, a total visit number is predetermined randomly in between range $\left[\lfloor \frac{|T|}{|L|} \rfloor, |T| - \argmin_{t \in T} \{w_t = 1\} + 1\right]$ (line 17). The lower bound of the total visit number is the minimum visit number to handle perishability infeasibility. However, in some of the cases, especially when $|T|$ increases, it is possible that $\lfloor \frac{|T|}{|L|} \rfloor - 1$ total visits might end up with a feasible schedule. On the other hand, as explained in subsection 4.1.4.2., trimming process could also remove these additional visits if production and inventory costs can be minimized by means of doing it. The rest of the procedure is continued with randomly assigning visits in between the first period of production and the last period until a predetermined total visit number is reached (line 18-26). Also, feasibility of each retailer's visit schedule is checked by the following conditions: the first visit period being later than the latest required delivery period for the retailer (i.e., $latest_i$) and the second is having more than $|L|$ consecutive periods without delivery.

To obtain a complete initial population, random greedy construction is executed until the total number of feasible chromosomes reaches to population size, $PopSize$. To finally check the feasibility of the chromosome, each solution is decoded. If there are infeasible solutions, the construction heuristic is executed again until a feasible solution is found.

4.1.8 Crossover

Crossover operation is significant due to its ability to explore the search space. In other words, it is the key factor in terms of being a major part of the intensification mechanism in genetic algorithms. Genetic algorithms traditionally provide diversification by their population-based nature and mutation operator. Yet, intensification is mainly provided by the crossover operator which helps inheriting the beneficial characteristics of the individuals in order to obtain fitter offsprings further. In other words, crossover is the process whereby two chromosomes (parents) partially contribute to a new child.

Two types of crossovers used in this study are single-point and two-point crossovers that applies with the crossover rate P_c . Selection of the crossover type is chosen randomly. Figure 4.5 shows an example of both types of crossovers. In both types, two parents produce two different children. Each child inherits all genes from one parent within the boundaries that crossover cutoff point or points determine. Then, the exchange operation is executed horizontally in between common rows (i.e., visit schedules of retailers). Basic idea behind horizontal exchange is that it has much less probability to result in infeasible children. On the contrary, vertical exchanges are more likely to result in insufficient visit schedule due to possibility of ending up with overloaded capacities and unsatisfied demands. In the single-point crossover, the crossover point is determined randomly in between 1 and $(n - 1)$ due to concerns of getting the same chromosomes. At the same time, which side of the point will be exchanged, either up or down, is determined randomly. Next to that, in two-point crossover, two cuts are made to divide parent chromosomes into three pieces. Then, only the middle parts are exchanged. At the end of the crossover operations, it is important to keep the offsprings feasible, therefore, right after the crossover, repairing mechanism is applied if needed.

4.1.9 Mutation

Even though population-based structure itself is known to be helpful to diversification mechanism, conventionally, performing mutations at a rare frequency also helps exploring various parts of the solution space which is also important to avoid trapping in the local optima. The idea of the mutation is to randomly mutate or alter the gene values and thus produce a new individual which is not so far away from the previous individual.

Unlike traditional genetic algorithms where mutation rates are very small, in the proposed HGA, mutation is applied more frequently. It is because PPIRP's search space is extremely large and the crossover operator does not achieve efficient exploration intensity. Hence, the mutation operator also plays a role to widen the exploration capability by a self-learning approach based on the success rates of the operators, while

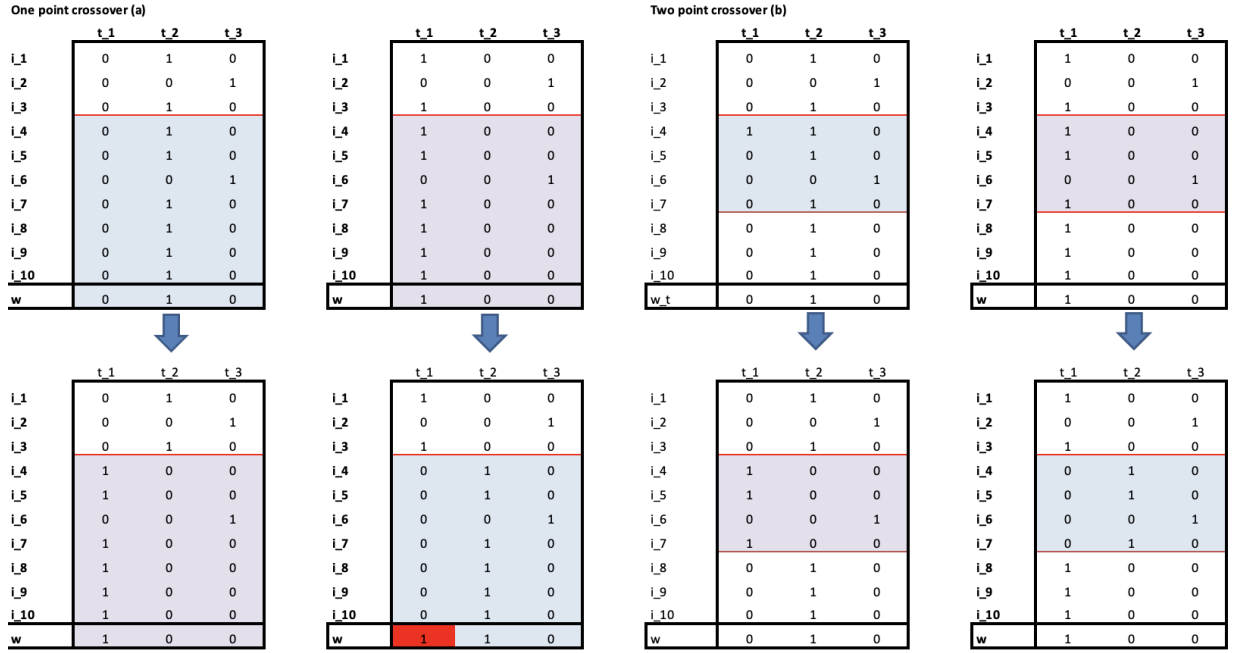


Figure 4. 5: Crossover operation for 10 retailers and 3 periods, (a) one-point crossover with the red cell is repaired, (b) two-point crossover

simultaneously conducting the variety of the population. Furthermore, note that the mutation is applied to parent chromosomes, with the mutation rate p_m , regardless of the crossover execution. In the remainder of this section, mutation operators and the mechanism of the operator selections are discussed.

4.1.9.1 Mutation Operators

Eight different types of operators are used to mutate an individual. Seven of the eight operators aimed at mutating the visit decisions (i.e., z_i^t), whereas only a single move is aimed for set-up (i.e., w^t) decisions. For the reason that the set-up decisions correspond to only a single row in the solution representation, there is a sufficient variety for the set-up schedule considering the population-based search. Thus, having a simple move for set-ups in the mutation mechanism, facilitates enough diversification already. On the contrary, visit decisions take up a relatively higher area in the search space which

motivates us to use more operators. The operators applied are listed below and illustrated in Figure 4.6.

- *One-to-one exchange*: It removes a randomly selected visit (i.e., $(i, t) \mid z_i^t = 1$) of a retailer $i \in R$, then reinserts the visit into another period without a visit for the selected retailer.
- *One-to-two exchange*: It removes a randomly selected visit (i.e., $(i, t) \mid z_i^t = 1$) of a retailer $i \in R$, then reinserts two visits into other two empty periods without visits for the selected retailer.
- *One-to-three exchange*: A randomly selected visit (i.e., $(i, t) \mid z_i^t = 1$) of a retailer $i \in R$ is removed, then three visits are reinserted into other three empty periods without visits for the selected retailer.
- *Exchange visit*: For a random retailer $i \in R$, two different periods, $t_1, t_2 \in T$, with different visit values (i.e., $z_i^{t_1} = 1$ and $z_i^{t_2} = 0$) are chosen, then visits are exchanged to $z_i^{t_1} = 0$ and $z_i^{t_2} = 1$.
- *Insert visit*: A visit is inserted into a random period $t \in T$ where there is no visit currently for a randomly chosen retailer i . (i.e., $z_i^t = 0$ changed to $z_i^t = 1$).
- *Move visits*: A period with a delivery (i.e., $z_0^{t_1} = 1$) is randomly chosen and visit schedule of that period vertically copied to another randomly chosen period, t_2 , without removing the visits in period t_2 .
- *Swap periods*: Two different periods, t_1 and t_2 , such that at least one of them having a visit (i.e., $z_0^{t_1} + z_0^{t_2} > 1$) are randomly chosen and they are vertically swapped.
- *Change set-up*: A random time period t is chosen and its production set-up status is flipped (i.e., if $w^t = 1$ then $w^t = 0$, otherwise, if $w^t = 0$ then $w^t = 1$).

Note that, these operators may lead to infeasible solutions, therefore, repairing procedure can apply after the mutation, if necessary. Selection strategy also plays an important role, which is discussed in the following section.

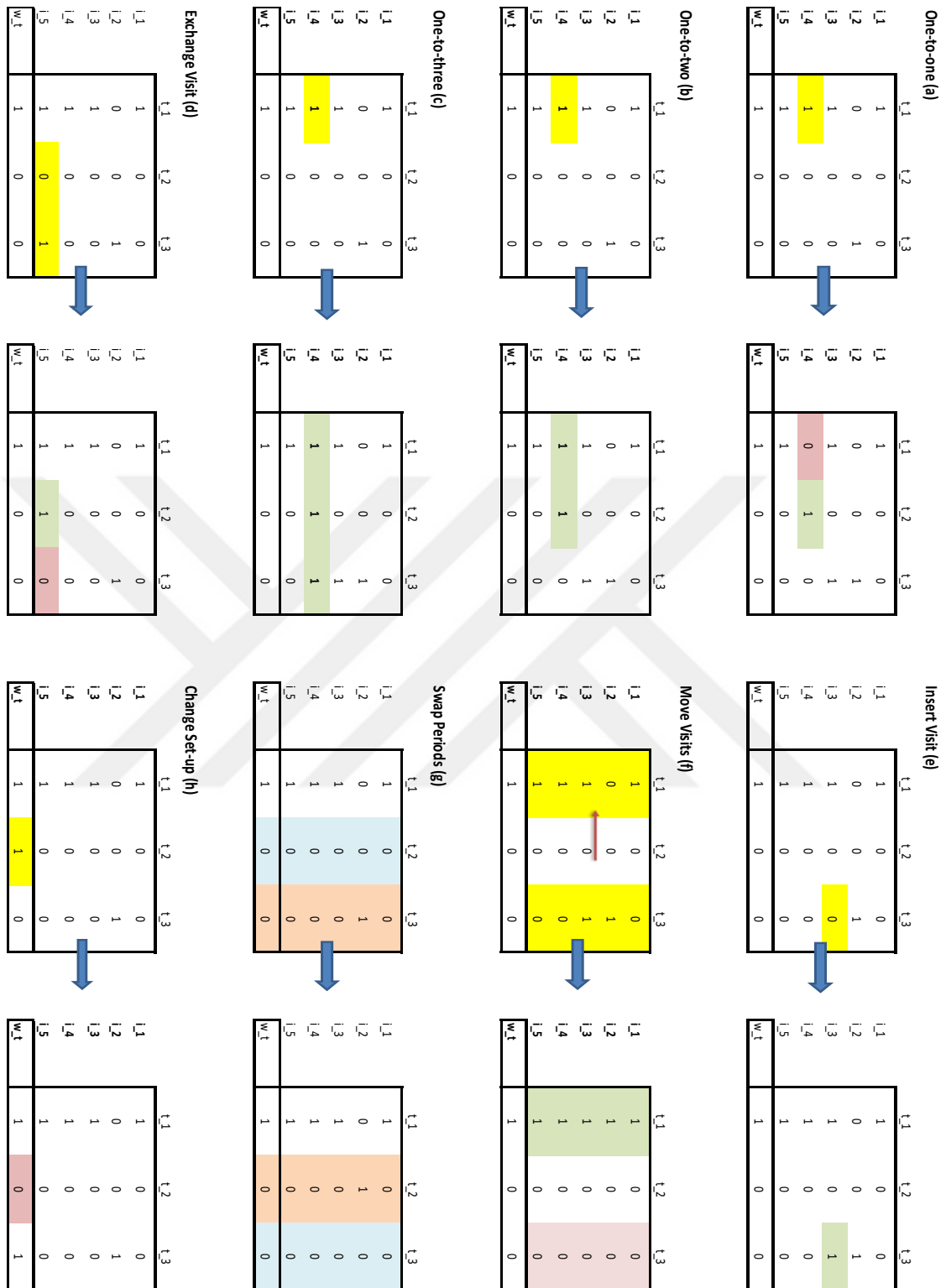


Figure 4.6: Mutation operators: (a) One-to-one exchange, (b) One-to-two exchange, (c) One-to-three exchange, (d) Exchange visit, (e) Insert visit, (f) Move visits, (g) Swap periods, (h) Change set-up.

4.1.9.2 Operator Selections

Selection of the mutation operator to be performed is controlled by the roulette-wheel mechanism. Let O be the set of mutation operators and each operator is assigned a weight denoted by $\Omega_{i \in O}$. An operator is selected at each mutation operation with probability, $p_i = \Omega_i / \sum_{j \in O} \Omega_j$. Weights are correlated with the success rates of the historical performances where operators that improve fitness of the chromosome are defined as successful. Therefore, operators with higher success rates are more likely to be selected. The weight of an operator i is calculated based on the percentage of calls that the operator i improves the current chromosome's fitness. Initial weights of the chromosomes are fixed at the beginning of the algorithm until the 10th generation. They are then recalculated at every 5 iterations. We use 25, 25, 25, 12, 10, 12, 6 and 15 as initial weights for the operators based on preliminary testing, respectively. The update mechanism of the weight of operator i is formulated as in Eq. (34), where μ_i and τ_i denotes the number of times that operator i is able to improve the fitness and is performed, respectively. Weights are also adjusted by parameter α according to the importance given to the successful operators. Higher α values encourage the algorithm to focus on repetitive successful operators that could improve the fitness frequently, hence promoting intensification but reducing diversification. So, $\alpha = 2$ is used in this study.

$$\Omega_i = 10^{1+\alpha\left(\frac{\mu_i}{\tau_i}\right)}, \quad \forall i \in O \quad (34)$$

This selection methodology based on success rates, balances the exploration mechanism of the search space in terms of avoiding unnecessarily high level of diversification. Since each operator brings out different characteristics of the solution and some of the operator might get less productive after generations, being selective and still stochastic about the mutation operators brings robustness to our proposed HGA.

4.1.10 Enlarged Sampling Space Formation and Selection Process

In each generation in the algorithm, new individuals produced by genetic operators are accumulated in a solution pool which is known as enlarged sampling space

containing the elite set with the fixed number (given parameter *eliteSize*) of the fittest chromosomes from the current population and acquired offsprings by the genetic operators. Size of this sampling space is equal to $PopSize \times \beta$, where we use $\beta = 1.8$ in our study. After sampling space is acquired, number of fittest *eliteSize* chromosomes are directly added to the next generation. Rest of the population is selected with roulette-wheel method which is a stochastic process as used in operator selection (i.e., subsection 4.1.9.2). When selecting a chromosome from the enlarged sampling space, fitness value is taken into account. According to fitness value of chromosomes, selection probabilities are assigned to each chromosome. Lower the fitness value of a chromosome, higher the chance of being a survivor for the next generation. After probability assignments, a random number is generated between 0 and 1 and then the corresponding number is checked in terms of the corresponding chromosome within cumulative distribution function intervals. This operation proceeds until the selected chromosome number for the next generation reaches the population size.

4.1.11 Stopping Criteria

Stopping criteria is selected as the combination of the maximum iteration number without improvement, namely *MaxUnimproved* and maximum generation number, *MaxGeneration*. Maximum unimproved iteration number is effective when generations are converged, and no progress is observed over long times. In that case it is pointless to wait until the maximum generation number is achieved. We use 500 as the parameter *MaxGeneration*, while parameter *MaxUnimproved* is determined as 250.

4.1.12 Elite Set Improvement

At the end of each generation, fittest chromosomes included in the elite set of the sampling space are tried to be further improved by running a travelling salesman problem (TSP) subroutine on each route individually. The basic idea is that in our proposed HGA, routing decisions are not explored as extensively as other decisions (i.e., production, inventory, and visit decisions). Even though tabu search is a strong local search technique especially for routing problems, the iteration number of tabu search algorithm is kept at

much lower values than the iteration number used for the tabu search of TIH in order to keep the total run time for the decoding part of HGA as short as possible. However, in terms of route-retailer assignments, current tabu search with low iteration number is still successful. Hence, TSP subroutine is likely to further extend the quality of routes. Current routes in the elite set are taken as initial solutions, then they are improved by the 2-opt procedure (*Lin 1965*) that basically looks for possibilities of a move that removes nonconsecutive arcs from a route, reverses the arcs in between them and adds two more arcs such that a new route is generated. That does not guarantee an exact improvement but since it is a fast local search and the current fitness cannot get any worse, it is used in this study. If a better solution is found, it is stored as the new incumbent solution.

4.2 *Adaptation of the Two-Phase Iterative Heuristic for PIRP*

The PPIRP formulation is very close to the combined formulations of the lot sizing problem and the vehicle routing problem. Apart from these two well-known and hard combinatorial problems, PPIRP needs one more decision which is considered in neither lot sizing problem nor vehicle routing problem, namely, delivery quantities at each quality level to each customer at each period. In order to decompose PPIRP, a variant of the lot sizing problem is taken into consideration. In this variant, delivery quantity decisions are also taken for each period. Regarding the delivery feasibility, variant of the lot sizing problem is considered with an aggregated fleet capacity without determining routing decisions. Main logic behind this TIH method is solving these two subproblems sequentially and iteratively until having near optimal solution.

Taking advantage of the decomposable structure that would emerge in possible real-world applications, PPIRP is decomposed into a variant of the lot-sizing subproblem and vehicle routing problems for each period. Decomposition is also necessary because of the problem size in order to deal with the complexity of the problem. Finally, PPIRP can be considered as two discrete subproblems. First is a variant of the lot sizing problem ($VLS(vc, \lambda)$) and second deals with the classical vehicle routing problem ($VRP(t)$) for each period $t \in T$. Here, we have new parameters vc_i^t and λ_t where vc_i^t is the visiting

cost of each retailer $i \in N$ and λ_t denotes the fill rate of the aggregate transportation capacity of the fleet in each period $t \in T$. These parameters have an important role in order to have feasible and high-quality solutions for PPIRP. Even though parameter vc_i^t is not included in the cost objective of the original problem, this visiting cost parameter is crucial for $VLS(vc, \lambda)$, since visiting and delivery decisions are taken in the $VLS(vc, \lambda)$ without specifying which vehicle to utilize. This visiting decisions and delivery amounts are then disaggregated into vehicles in the $VRP(t)$ decomposition which we called second phase. Thus, visiting cost parameter vc_i^t has a significant role in terms of connecting the first and the second phases. Note that the initial value of the vc_i^t is set to $c_{i0} + c_{0i}$. After solving the second phase in which routing decisions are taken, the value of the vc_i^t is updated. Main reason to update the visiting cost is to increase the quality of the delivery plan obtained by $VLS(vc, \lambda)$ at each iteration. Considering the initial values of parameter vc_i^t , solutions tend to visit customers far from the plant less frequently than the ones near the plant. So, vehicle routes are characterized with clusters without considering the proximity of the customers which leads to higher transportation cost. Hence, iteratively updating the parameter vc_i^t is an efficient way to have less costly solutions in which visited customers in each tour are efficiently selected. Otherwise, visiting widely scattered customers in the same tour would cost more. Framework of updating the visiting costs can be seen in Algorithm 3. Updating procedure improves the clustering of the retailers for further iterations by setting the parameter to the auxiliary difference in the related period, independent of whether the considered retailer existing in a tour or not the in the related period. Also, diversification of the value of parameter vc_i^t is also necessary especially when the algorithm is stuck in the local optima. If solution is not improved in pre-determined U iterations, vc_i^t is set to $\theta(c_{i0} + c_{0i})$, where θ is a random number within a range $[0.5, 1.5]$.

To ensure a feasible decomposition and to have a good quality solution, the lot-sizing problem must take the delivery cost and the transportation capacity into account in

Algorithm 3 – Updating Visiting Costs

```

1. for( $t \in T$ )
2.   for( $i \in R$ )
3.     if ( $i \in R_t$ )
4.        $vc_i^t = c_{i-i} + c_{i+i} - c_{i-i+}$ 
5.     else if ( $R^t \neq \emptyset$ )
6.        $vc_i^t$  is set to minimum insertion cost of retailer  $i \in R$  to any tour in  $t \in T$ 
7.     else
8.        $vc_i^t = c_{i0} + c_{0i}$ 
9.   end if
10. end for
11. end for

```

an aggregated perspective by parameters vc_i^t and λ_t . Additionally, fill rate of the aggregate capacity of the fleet, λ_t , in period $t \in T$, is significant for restricting the overall capacity usage. If routing decomposition gives an infeasible solution, fill rate λ_t should be reduced to get a feasible solution for the periods that generate infeasibility. λ_t does not guarantee feasible routing solutions for the second phase since split delivery is forbidden in the same period. Even though aggregate delivery quantities in period $t \in T$ is less than or equal to the aggregate capacity with the fill rate λ_t , different deliveries might not fit into the available vehicles, hence in case of infeasibility λ_t is set to $\lambda_t - \tau$ for the next iteration of TIH.

$VLS(vc, \lambda)$ determines the following decisions: the quantity of the product at each quality level for delivering to retailers in each period; quantity for fulfilling the demands at each retailer for each quality level in each period; and inventory levels for each retailer in each period. Delivery quantities are crucial for PPIRP. After having delivery quantity decisions, we can obtain the values for other variables easily. This idea has been widely used for the algorithms developed for IRP. Once $VLS(vc, \lambda)$ is solved, production and inventory costs, which are defined by objective function (1) without routing costs is obtained. $VLS(vc, \lambda)$ and $VRP(t)$ are solved sequentially and iteratively in order that the aggregate perspective of $VLS(vc, \lambda)$ leads to a feasible solution. Vehicle routing problems are solved for each period after lot sizing sub problem obtains the delivery quantities at

each period. The variant of the lot sizing problem with aggregated vehicle capacity is addressed in the following section.

4.2.1 Variant of the Lot Sizing Problem

VLS (vc, λ) is obtained by eliminating routing related variables x_{ij}^{kt} and routing related constraints (14)-(16). Vehicle index $k \in K$ is removed from all other variables because of the aggregated approach for the capacity of the distribution. Introduced new parameters and the formulation are as follows:

vc_i^t : visiting cost for retailer $i \in R$ in period $t \in T$,

λ^t : fill rate of the fleet capacity in period $t \in T$.

$$\text{Min } f_{VLS} = \sum_{t \in T} (e_t p_t + f_t w_t) + \sum_{i \in N} \sum_{l \in L} \sum_{t \in T} h_i^l I_i^{lt} + \sum_{i \in N} \sum_{t \in T} vc_i^t z_i^t \quad (35)$$

$$\text{s.t. } I_i^{l0} = g_i^l, \quad \forall i \in N, l \in L \quad (36)$$

$$p_t \leq C w_t, \quad \forall t \in T \quad (37)$$

$$\sum_{l \in L} I_i^{l,t-1} + \sum_{l \in L} q_i^{lt} \leq U_i, \quad \forall i \in R, t \in T \quad (38)$$

$$\sum_{l \in L} I_0^{lt} \leq U_0, \quad \forall t \in T \quad (39)$$

$$\sum_{l \in L} d_i^{lt} = D_i^t, \quad \forall i \in R, t \in T \quad (40)$$

$$I_0^{lt} = I_0^{l-1,t-1} - \sum_{i \in R} q_i^{lt}, \quad \forall l \in L \setminus \{0\}, t \in T \quad (41)$$

$$I_0^{0t} = p_t - \sum_{i \in R} q_i^{0t}, \quad \forall t \in T \quad (42)$$

$$I_i^{lt} = I_i^{l-1,t-1} + q_i^{lt} - d_i^{lt}, \quad \forall i \in R, l \in L \setminus \{0\}, t \in T \quad (43)$$

$$I_i^{0t} = q_i^{0t} - d_i^{0t}, \quad \forall i \in R, t \in T \quad (44)$$

$$\sum_{i \in R} \sum_{l \in L} q_i^{lt} \leq \lambda^t |K| V, \quad \forall t \in T \quad (45)$$

$$\sum_{l \in L} q_i^{lt} \leq V z_i^t, \quad \forall i \in R, t \in T \quad (46)$$

$$q_i^{lt}, I_i^{lt} d_i^{lt}, p_t \geq 0, \quad \forall i \in R, l \in L, t \in T \quad (47)$$

$$z_i^t, w_t \in \{0, 1\}, \quad \forall i \in R, k \in K, t \in T \quad (48)$$

Objective function (35) minimizes the total production cost, set-up cost, inventory holding cost and visiting cost. Constraints (36)-(44) and (46) are the same constraints

with the constraints (2)-(12) in the main PPIRP model without the vehicle specific decisions indexed by $k \in K$. Constraints (45) ensure that total delivery quantity to all retailers cannot exceed the total fleet capacity with the given fill rate, λ^t , for each period. Constraints (47)-(48) are the integrality constraints.

In addition to the constraints above, the following strengthening valid inequalities are useful to cut off infeasible solutions and break symmetric situations:

$$\sum_{t \in T} z_i^t \geq \lceil \frac{(\sum_t D_i^t - \sum_l g_i^l)}{\min\{V, U_i\}} \rceil, \quad \forall i \in R \quad (49)$$

$$\sum_{t \in T} z_0^t \geq \lceil \frac{\sum_i \sum_t D_i^t - \sum_i \sum_l g_i^l}{(Vx|K|)} \rceil \quad (50)$$

$$\sum_{t \in T} z_i^t \geq \lfloor \frac{|T|}{|L|} \rfloor - \lfloor \frac{g_i^0}{\sum_{t=1}^{|L|+1} D_i^t} \rfloor, \quad \forall i \in R \quad (51)$$

Constraint (49) enforces that the minimum visit number to customer i is greater than or equal to the rounded-up number which equals to the quantity needed to satisfy the future demands divided by the maximum amount that can be carried to the customer. A similar reasoning is implemented to the plant in valid inequality (50). Constraint (51) ensures that the total visit number to customer i is greater than the minimum required visit number that is determined by the perishable characteristic of the product and the initial inventory regardless of the capacity restrictions.

Lot sizing decomposition has a special structure such that once we have the solution of this decomposition, we have the inventory and fulfillment decisions before going to the next step for routing decisions. Hence, getting high quality solutions from the lot sizing decomposition is a significant step. In order to have near optimal solutions, sets of parameters vc_i^t and λ^t are updated at each iteration by the solution of $VRP(t)$. $\lambda^t \in (0, 1]$ is decreased by $\tau \in (0, 1]$ when the solution of $VRP(t)$ leads to an infeasible solution as the determined delivery quantities by $VLS(vc, \lambda)$ cannot be satisfied by the available fleet. Since vc_i^t is only used in the objective function of the decomposition, it does not affect the feasibility. However, vc_i^t is crucial for iteratively improving the

solution by being updated based on the solution in the previous iteration and by being diversified to reach the global optimum.

In conclusion, once we have a feasible solution to the variant of the lot sizing problem, $VLS(vc, \lambda)$, set of retailers R^t to be visited in each period $t \in T$ are determined, where $R^t = \{i \in R | z_i^t = 1\}$. In terms of the delivery plan, we obtain Q_i^t which denotes the corresponding delivery quantities for retailer $i \in R^t$ in period $t \in T$, where $Q_i^t = \sum_l q_i^{lt}$. Also, quality of the deliveries is demonstrated in a quicker way, since we already obtain q_i^{lt} variables.

4.2.2 Vehicle Routing Problem

A solution obtained from $VLS(vc, \lambda)$ provides all the required information to reduce the rest of the problem to a traditional vehicle routing problem for each period $t \in T$ independently. Thus, solving $VRP(t)$ for all periods independently such that $R^t \neq \emptyset$ leads us to obtain the values of all the decision variables for the PPIRP. Remember that in the first phase, we found the total operational cost except routing part together with the average quality levels of deliveries. Summation of independent routing costs of all periods and production-inventory cost which are obtained in $VLS(vc, \lambda)$ establishes the main objective function (1) of PPIRP. General framework of TIH algorithm is given in Figure 4.7, and we can observe the following characteristics of this solution method:

The final solution is obtained by iteratively solving lot sizing and vehicle routing sub problems. In addition, the problem is directly decomposed into T sub problems due to time independency, whereas lot sizing sub problem has only one solution in the same iteration. Also, lot sizing phase outputs are the main inputs for VRP phase. Finally, $VRP(t)$ can be formulated as follows:

$$\text{Min } f_{RC} = \sum_{(i,j) \in A} \sum_{k \in K} c_{ij} x_{ij}^{kt} \quad (52)$$

$$\text{s.t } \sum_{i \in R^t} Q_i^t z_i^{kt} \leq V, \quad \forall k \in K \quad (53)$$

$$\sum_{k \in K} z_i^{kt} = 1, \quad \forall i \in R^t \quad (54)$$

$$\sum_{j \in R^t \cup \{0\}, j \neq i} x_{ij}^{kt} + \sum_{j \in R^t \cup \{0\}, j \neq i} x_{ji}^{kt} = 2z_i^{kt}, \quad \forall i \in R^t, k \in K \quad (55)$$

$$\sum_{j \in R^t} x_{0j}^{kt} \leq 1, \quad \forall k \in K \quad (56)$$

$$\sum_{i \in S} \sum_{j \in S: i \neq j} x_{ij}^{kt} \leq \sum_{i \in S} z_i^{kt} - z_l^{kt}, \forall S \in R^t, |S| \geq 2, k \in K, t \in T, l \in S \quad (57)$$

$$z_i^{kt}, x_{ij}^{kt} \in \{0, 1\}, \quad \forall i \in R^t, k \in K, r \in V \quad (58)$$

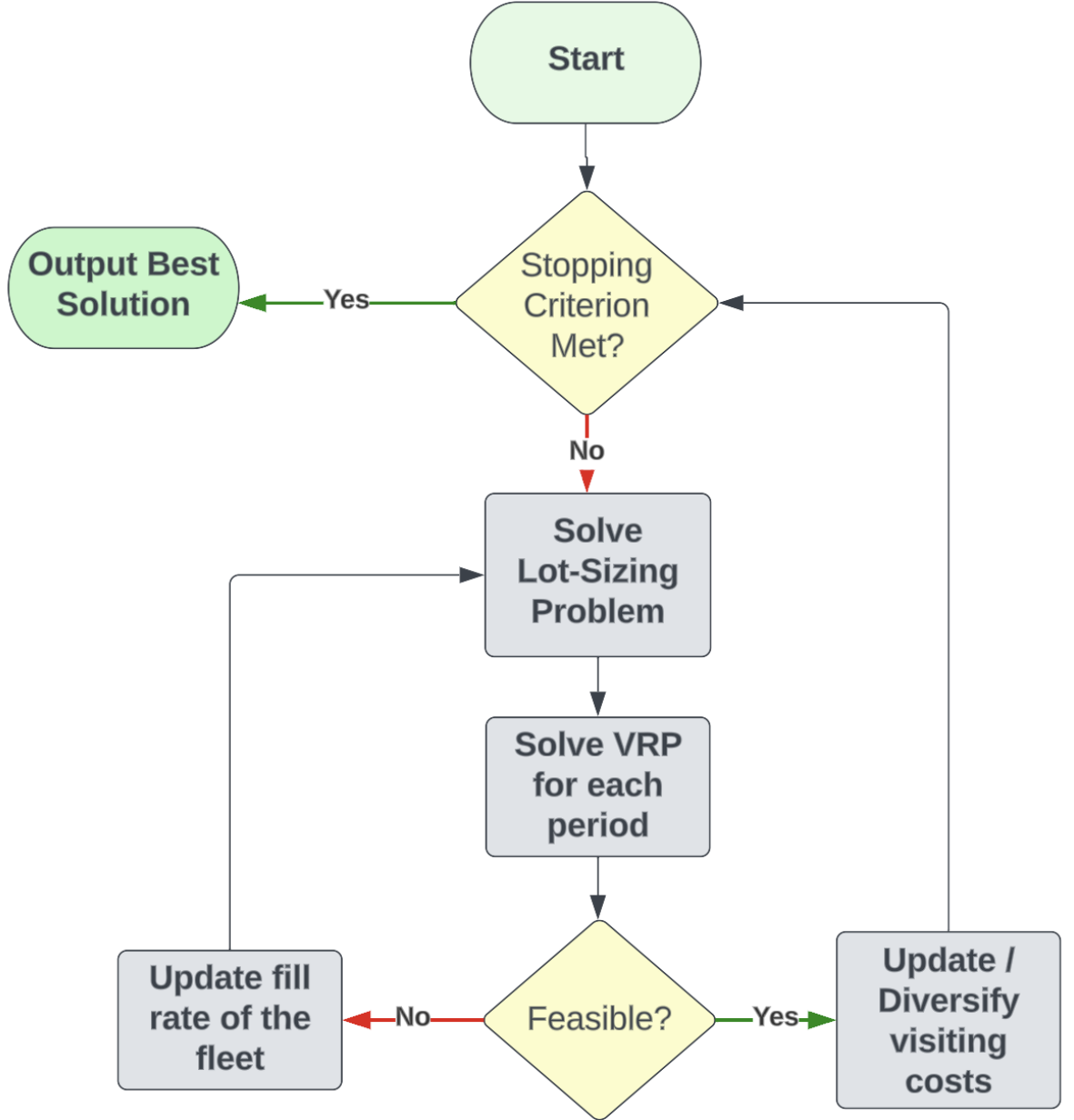


Figure 4. 7: Framework of the TIH Algorithm

Modifications over the main model are done as follows: instead of set R , set R^t is used for the retailers to be visited and instead of the expression $\sum_{l \in L} q_j^{lkt}$, Q_i^t is used as a parameter that is obtained from the lot sizing phase that indicates specific quantity to be delivered to retailer $i \in R^t$ in period $t \in T$. The crucial role of the parameter vc_i^t is not shown in the framework of TIH in Figure 4.7, since it is elaborately discussed in the previous section and main reason for showing the framework is to understand how a feasible solution for PPIRP is generated. Detailed description of the TIH is introduced in Algorithm 4.

At the end of each iteration, a new solution is obtained with different parameter setting of vc_i^t and λ_t . Parameter vc_i^t is updated and diversified (line 12 – 13) with the previously explained methods at the end of each iteration. For diversification, we have a counter l for the unimproved solutions. Once we have more than the maximum number of unimproved solutions, U , a diversification mechanism is executed (line 13). Fill rate, λ_t , is reduced if second phase gives an infeasible solution (line 8 – 9). At each iteration, after the lot sizing phase is solved, VRP phase is solved iteratively for each period $t \in T$ (line 5 – 7). TIH algorithm terminates after I iteration. The rest of the lines in the algorithm are self-explanatory.

4.2.2.1 Tabu Search

Vehicle routing problem is a well-known NP-hard problem, hence solving VRP(t) for medium and larger instances with commercial solvers cannot be expected because of time considerations. Even small instances can take considerable CPU times. Hence a Tabu Search is used for solving VRP (t). Tabu search is proven to be very successful in the VRP literature and it is widely used for its strength to converge near optimal solutions in a short amount of time. Neighborhood structure of the algorithm consists of an exchange move which is based on relocating an existing node into a new position in either another route or the same route. The exchange move is illustrated in Figure 4.8. Stopping criterion is selected as the maximum number of iterations.

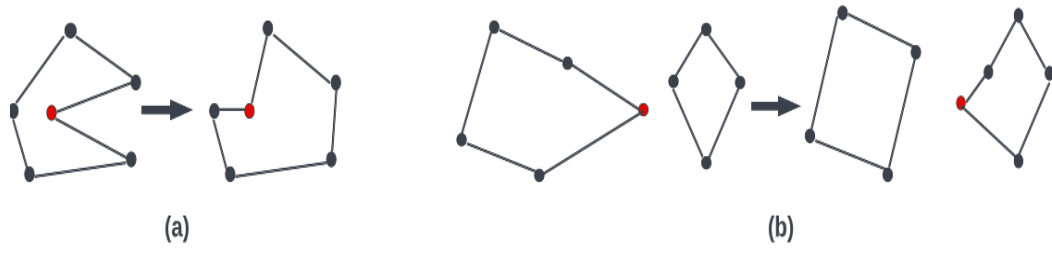


Figure 4.8: Exchange move, (a) intra-route and (b) inter-route

Algorithm 4 – TIH Algorithm for PPIRP

1. **Initialize:** $f_1 = +\infty$, $iter = 0$, $l = 0$;
 2. **input:** I , U , λ , τ , vc
 3. **while** ($iter < I$) **do**
 4. Solve VLS (λ, vc) $\rightarrow$ Get production and inventory cost (PIC), R^t and Q_i^t
 5. **for** ($t \in T$)
 6. Solve VRP (t) for $t \in T \rightarrow$ Update routing cost (RC)
 7. **end for**
 8. **if** (Number of routes at any period $> |K|$) $\lambda_t = \lambda_t - \tau \rightarrow$ Go to line 4
 9. **else if** ($PIC + RC < f_1$) $vc^* = vc$, $\lambda^* = \lambda$, $f_1 = PIC + RC$
 10. **else** $l = l + 1$
 11. **end if**
 12. Update_vc()
 13. **if** ($l \geq U$) Diversify_vc(), $l = 0$
 14. $iter = iter + 1$
 15. **end while**
 16. **output:** f_1
-

Chapter 5:

COMPUTATIONAL EXPERIMENTS

In this chapter, computational experiments of the proposed algorithms described in Chapter 4 are discussed. The performance of the algorithms are evaluated comparatively from different aspects. Algorithms have been implemented in *Java* using the *Eclipse IDE for Java Developers version 4.27.0*. We call the *IBM ILOG CPLEX 22.1.1* as a commercial solver with the default settings for the mathematical models of PPIRP and its sub problems. Runs are taken on a personal computer with *Apple M2 chip* and *16 GB RAM* using the *MacOS* operating system. The goals of this computational experiment are to assess the effectiveness of our HGA against the adapted and well-known TIH algorithm, as well as to identify the maximum sizes of the instances that can be solved exactly using both algorithms by comparing with the results of the mathematical models. Note that TIH is an effective and common matheuristic in both production routing and inventory routing literature as a benchmark. In the remainder of this chapter, first the instances we use to evaluate the performances of our algorithms are explained. In the second part, selection of the parameters for the proposed algorithms is discussed. Finally, extensive numerical results are evaluated with different measurement techniques in order to compare the abilities of the proposed solution methodologies.

5.1 Instance Generation

With the aim of evaluating the performance of the proposed algorithms, a new instance set is generated. Instance set proposed by *Coelho and Laporte (2014)* for the perishable inventory routing problem is utilized as a main resource for our instance set. *Coelho and Laporte (2014)* randomly generated instances on a wide range of situations. There are 70 different randomly generated instances which vary in terms of the number of retailers, periods, vehicles and maximum age of the perishable product. The problem they have studied is close to our problem, as discussed in Chapter 2, with the only difference of lacking production decisions. Also, in their problem, they do not consider a storage capacity at the plant. Hence, we add these parameters to be able to evaluate our algorithms and to create a novel instance set for the perishable production routing problem literature. The tests have been carried out on four different classes of instance sets with a

similar approach employed in *Archetti et al. (2011)*. The first class, base class alternatively, instances have been generated with the following logics:

- Number of retailers N : 10, 20, 30, 40, 50
- Time horizon T : 3 and 6 for up to $N = 50$ retailers; 10 for up to $N = 40$ retailers.
- Number of vehicles in the homogeneous fleet K : 1 for $N = 10$ retailers; 2 for $N = 20$ and $N = 30$ retailers; 3 for $N = 40$ and $N = 50$ retailers.
- Maximum age of the products $|L|$: 2 for $T = 3$ periods; 3 for $T = 6$ periods and 5 for $T = 10$ periods.
- Demand D_i^t for retailer $i \in R$ in period $t \in T$: selected randomly from the interval $[30, 210]$ (i.e., directly taken from *Coelho and Laporte (2014)*).
- Distance matrix d_{ij} : formed via the euclidian distances of the coordinates (x, y) of plant and retailers that are randomly selected from the interval $[0, 1000]$ (i.e., positions are directly taken from *Coelho and Laporte (2014)*).
- Inventory capacities U_i for retailer $i \in R$: $r \times \max_t \{d_i^t\}$, where r is randomly selected from the set $\{2, 3\}$ (i.e., directly taken from *Coelho and Laporte (2014)*).
- Initial inventory of the freshest product g_i^0 : $U_i - D_i^1$.
- Inventory holding cost h_i^l for node $i \in N$ and age $l \in L$: equals to $(r_1 + l \times r_2 / (1 + l)) / 100$ where r_1 and r_2 are randomly selected within intervals $[0, 100]$ and $[0, 70]$, respectively (i.e., directly taken from *Coelho and Laporte (2014)*).
- Vehicle capacity V : equals to $[1.25 \times \sum_{i \in R} \sum_{t \in T} D_i^t / (TK)]$.
- Production capacity at the plant C : equals to $(\sum_{i \in R} \sum_{t \in T} D_i^t / T) \times r$, where r is randomly selected within the set $\{3, 4\}$.
- Inventory capacity for the plant U_0 : equals to $C \times 2$.
- Unit production cost e^t : equals to $5 \times h_0^0$.
- Fixed set-up cost f^t : equals to $500 \times e^t$.

For each combination of N and T , total of 5 different instances were generated. Thus, for each class of instances we generated 70 instances. Considering we have four different class, each indicating different cost structures, a total of 280 instances were generated.

The second class of instances is identical to the base class but differs with only one parameter change. The coordinates (x, y) of all plant and retailers are multiplied by 5. This class of instances simulates situations with higher transportation costs. The third class of instances omits the inventory costs occurring in retailers, i.e., $h_i^l = 0, \forall i \in R, l \in L$, whereas inventory costs in the plant is still relevant to the main decision maker. Finally, the fourth class only differs from the first class by the multiplication of the unit cost parameter e^t by 5. This class simulates the situation where production costs have a greater impact than the inventory costs. Having these four types of instance classes, algorithm performances are evaluated from different aspects that might emerge in real life situations. Therefore, results of different classes may reveal useful insights for the decision makers.

5.2 Parameter Settings

Tuning the parameters of the heuristic algorithms plays a significant role when implementing heuristics. In the adapted version of TIH, a variety of the parameters is less than that of the proposed HGA, hence fine-tuning operation is less complex. The best performing parameter setting for TIH is determined via preliminary computational experiment by using different parameter settings and randomly selecting different sizes of instances. TIH parameters are as follows: U , indicating the maximum number of iterations without improvement, is set to 40 for instances with $N = 10$ retailers, 50 for 20 retailers, 60 for 30 and 40 retailers and 70 for 50 retailers. I , indicating the maximum number of iterations is set to 100 for instances with $N = 10$ retailers, 130 for 20 retailers, 140 for 30 retailers, 150 for 40 retailers and 200 for 50 retailers, respectively. It is observed that for greater values for U and I , the solution quality does not improve. Then, θ value that helps diversifying the visiting cost is randomly selected within the range $[0.5, 1.5]$. Initial fill rate λ_t values are set to 1.0 for each period and τ value is chosen as 0.05 for all instances. The tabu tenure parameter for the tabu search which is executed for solving the vehicle routing problem for each period is randomly set within the ranges $[0, 5]$ for 10 retailers, $[5, 15]$ for 20 and 30 retailers, and $[20, 35]$ for 40 and 50 retailers, respectively.

For the tuning of the HGA parameters, a more comprehensive study has been carried out. A test set consisting of the total of 80 instances with 10 periods from different classes has been used. The tuning on this test set was made by using different parameter settings within random intervals. Parameter settings that output smallest average of the best solution are considered. Following assumption is considered during the test: since instances with 10 periods represent the largest instances with larger search spaces, parameter settings resulting in quality solutions are likely to succeed in smaller instances as well. Finally, the stopping criteria for all the experiments in the tuning process were set to a maximum of 1 hour time limit. Resulting parameter values are shown in Table 5.1 and Table 5.2. Tested intervals and sets for the parameters are as follows: crossover rate (P_c) is determined as 0.9 based on a test performed within interval [0.6, 0.95]; mutation rate (P_m) is determined as 0.5 from interval [0.1, 0.7]; β , which sets the enlarged sampling space size by being multiplied with PopSize, is determined as 1.5 from interval [1.2, 2.5]; maximum generations without improvement is set to 250 where the tested interval is [100, 400]; elite set size is tested within the set {PopSize/20, PopSize/10, PopSize/5} and determined as PopSize/10; finally, the maximum generation number was set to 1000 during the test but determined as 500 after observing that there has not been an improvement within 1 hour time limit after 500 generations. On the other hand, during the tuning test for the fixed parameters, population size is taken as 100. However, for our HGA, it is a value that may take unnecessarily long run times or may be insufficient for the search ability. Hence, for each of the instance size characterized by periods, T , and number of retailers, N , sequential runs are taken to determine the best population sizes. Results are shown in Table 5.2.

Table 5. 1: Fixed Parameters of HGA

Crossover Rate	Mutation Rate	Max. Generations	Max. Generations w/o Improvement	Elite Set Size	β
0.9	0.5	500	250	Population Size / 10	1.5

Table 5. 2: Population Size per Instance Size

T	N	Population Size
3	10	30
	20	60
	30	70
	40	80
	50	90
6	10	40
	20	70
	30	80
	40	100
	50	110
10	10	50
	20	80
	30	90
	40	110
	50	120

5.3 Numerical Results

In this section, experimental results for the TIH and proposed HGA are presented. HGA and TIH are compared in terms of various aspects. Since each algorithm contains randomness in various parts, each objective function value and run time reported represents the average value obtained out of 5 replicated runs. In 5.3.1, Hybrid Genetic Algorithm and adapted Two-Phase Heuristic are compared with MILP model. In this manner, it is examined whether our algorithms achieve optimality or negligible optimality gaps. Also, relative differences between both algorithms are discussed. In 5.3.2, medium and large size instances are computationally experimented for our algorithms in which model could not achieve to output feasible solutions under the limit of one hour run time. Finally, production, inventory and transportation costs featured in the main objective function of PPIRP are separately inspected to discuss managerial insights in terms of using each algorithm.

5.3.1 Performance Analysis of TIH and HGA for Small Instances

This section presents the results for the instances with 10 retailers which corresponds to small instances. In other words, these are the instances that mathematical

model achieves to return feasible solutions within one hour time limit. We focus on the performance of the model and the algorithms, TIH and HGA. The analysis with regards to TIH and HGA is based on the results from five replicated runs carried out for each of the instances from four different classes. A comparison among the three solution approaches with regard to different instance characteristics is shown in Tables 5.3 – 5.4. Columns represent the number of periods (T), number of retailers (N), instance number (Instance), resulting optimality gap found by MILP (Gap (%)) and as well as the run time (Time), in seconds. For the proposed algorithms, “Rel. Diff. (%) *” refers to the relative difference between the average objective value of the algorithms and the objective value found by MILP within the given time limit. Average run time of the algorithms are indicated by “Time” column. Finally, the last column heading (Rel. Diff. (%)) presents the relative difference between TIH and HGA. For the relative difference values less than 0.00, we see that HGA outperforms TIH. In the following, algorithm performances on each instance class that emphasizes different cost structure are discussed.

As Table 5.3 shows a comparison of the average performance for the small instances of class I instances, MILP is able to find optimal solutions in reasonable run times with an average of 6.2 seconds and 73.2 seconds for the instances with 3 and 6 periods, respectively. For the instances with 10 periods, MILP could achieve optimality in 3 out of 5 instances. There are only 2 instances in class I that MILP returned optimality gaps higher than 0% after 1 hour run time, which are 9.18% and 11.73%, respectively. On the other hand, TIH and HGA were able to solve the instances with less than 10 periods with an average optimality gap of less than 0.1%. Average run times of both algorithms are approximately 12 and 25 seconds, respectively. Looking at the instances with 10 periods, relative differences between TIH and MILP hit under 1% for 3 out of 5 instances for which MILP was able to find optimal solutions. For the remaining 2 instances, TIH averages -2.08% relative difference against MILP results with 215.1 seconds run time while MILP terminates in an hour. HGA outperforms both MILP and TIH in terms of relative differences by averages of -1.76% and -1.21% respectively. HGA averaged 0.09% relative difference against MILP where optimality is achieved for 3 out of 5 instances. For the rest, HGA outperforms MILP by the average of -4.55% relative difference. Comparing the HGA for 10 period instances with TIH, HGA succeeds better for all instances in terms of the objective function value. On the other hand, TIH slightly

did better in terms of run times with 189.1 seconds average while HGA averages 221.0 seconds.

Table 5. 3: TIH and HGA performances on class I-small instances

T	N	Instance	CPLEX		TIH		HGA		
			Gap(%)	Time	Rel. Diff. (%)*	Time	Rel. Diff. (%)*	Time	Rel. Diff (%)
3	10	1	0.00	4.0	0.00	5.0	0.00	4.1	0.00
		2	0.00	6.0	0.00	4.1	0.00	6.3	0.00
		3	0.00	6.0	0.00	6.1	0.00	7.2	0.00
		4	0.00	9.0	0.00	5.1	0.00	8.0	0.00
		5	0.00	6.0	0.00	5.1	0.00	2.8	0.00
6	10	6	0.00	97.0	0.00	20.6	0.00	33.3	0.00
		7	0.00	49.0	0.00	13.1	0.00	45.7	0.00
		8	0.00	131.0	0.00	23.8	0.19	78.0	0.19
		9	0.00	47.0	0.00	16.7	0.00	25.3	0.00
		10	0.00	41.0	0.00	16.3	0.13	45.7	0.13
10	10	11	0.00	1737.0	0.78	240.4	0.00	200.3	-0.78
		12	0.00	1286.0	0.39	134.6	0.28	318.3	-0.10
		13	9.18	3596.0	-2.66	313.1	-4.10	219.5	-1.58
		14	0.00	812.0	0.08	140.4	0.00	139.3	-0.08
		15	11.73	3601.0	-1.50	116.8	-5.00	227.3	-3.53
3			0.00	6.2	0.00	5.0	0.00	5.7	0.00
6			0.00	73.0	0.00	18.1	0.06	45.6	0.06
10			4.19	2206.4	-0.58	189.1	-1.76	221.0	-1.21

For the second class of instances, finding optimal solutions with MILP seems to be particularly challenging. As in the first-class instances, MILP found optimal solutions for all 3- and 6-period instances within an average of 152 seconds run time, as presented in Table 5.4. However, for the 10-period instances, all the experiments hit the time limit of 1 hour with an average of 18.66% optimality gap. The reason can be explained with the cost increase in transportation part for this class. Since the routing cost (i.e., $\sum_{(i,j) \in E} \sum_{k \in K} \sum_{t \in T} c_{ij} x_{ij}^{kt}$) within the overall cost has the majority, the way to obtain a high-quality solution is to perform well in the vehicle routing problems included in PPIRP. Having said that, due to vehicle routing problem being an NP-Hard problem, MILP cannot find quality solutions with an exhaustive computational effort in an hour. TIH and HGA yield 0% optimality gaps for all instances with 3 periods as well. They also average 0.44% and 0.18% average gaps for the 6-period instances, respectively. Even though both run time averages are reasonable, TIH outperforms HGA in terms of overall run times. Besides, HGA results with higher quality solutions than TIH by achieving -0.25% and -2.50% relative differences for 6 and 10 periods respectively. There is only

one instance that TIH was able to outperform HGA in this class, which is with 0.1% relative difference only. Accordingly, this class has the most gap between HGA and TIH in terms of the solution quality.

Table 5. 4: TIH and HGA performances on class II-small instances

T	N	Instance	CPLEX		TIH		HGA		
			Gap(%)	Time	Rel. Diff. (%)*	Time	Rel. Diff. (%)*	Time	Rel. Diff (%)
3	10	1	0.00	7.0	0.00	5.0	0.00	8.9	0.00
		2	0.00	12.0	0.00	5.1	0.00	4.1	0.00
		3	0.00	14.0	0.00	5.1	0.00	5.1	0.00
		4	0.00	16.0	0.00	5.1	0.00	8.1	0.00
		5	0.00	10.0	0.00	6.1	0.00	6.1	0.00
6	10	6	0.00	271.0	0.00	20.6	0.00	37.2	0.00
		7	0.00	124.0	0.82	14.1	0.92	59.6	0.10
		8	0.00	505.0	0.53	25.7	0.00	106.9	-0.52
		9	0.00	246.0	0.85	17.6	0.00	31.4	-0.84
		10	0.00	315.0	0.00	16.3	0.00	62.2	0.00
10	10	11	15.19	3598.0	-2.22	119.2	-3.90	231.3	-1.73
		12	14.17	3600.0	-0.68	97.0	-1.47	316.8	-0.79
		13	15.85	3602.0	-2.57	182.8	-5.39	300.0	-2.89
		14	23.90	3599.0	-0.67	84.8	-5.44	147.5	-4.80
		15	24.19	3589.0	-6.48	100.0	-8.64	310.9	-2.31
3			0.00	11.8	0.00	5.2	0.00	6.4	0.00
6			0.00	292.2	0.44	18.9	0.18	59.5	-0.25
10			18.66	3597.6	-2.52	116.8	-4.97	261.3	-2.50

Table 5.5 represents the results for the third-class instances where the inventory cost at retailers is neglected. Similar to the first two classes of instances, MILP achieves optimality for all instances with 3 and 6 periods. On the other hand, it averages 9.26% optimality gap for the instances with 10 periods in which optimality could not be obtained in any single instance. Relative differences between TIH and HGA on 3- and 6-period instances are negligible, with an average of 0.0% and 0.17%, again as in the previous experiments. We observe that HGA outperforms TIH in all instances for 10 periods, except the one that is tied with 0.0% relative gap. However, relative difference is decreased according to the first two classes with an average of -0.68% relative difference. In general, both TIH and HGA succeed in lower optimality gaps than MILP in 10-period instances with -1.64% and -2.32%, respectively. In terms of the run times, HGA is more efficient for the instances with 3 periods, whereas TIH outperforms other solution approaches with 18.7 seconds and 117.8 seconds for 6 and 10 periods, respectively.

Table 5. 5: TIH and HGA performances on class III-small instances

T	N	Instance	CPLEX		TIH		HGA		
			Gap(%)	Time	Rel. Diff. (%)*	Time	Rel. Diff. (%)*	Time	Rel. Diff (%)
3	10	1	0.00	7.0	0.00	5.0	0.00	2.0	0.00
		2	0.00	9.0	0.00	5.1	0.00	2.0	0.00
		3	0.00	8.0	0.00	5.1	0.00	2.0	0.00
		4	0.00	9.0	0.00	5.1	0.00	6.0	0.00
		5	0.00	11.0	0.00	5.1	0.00	2.0	0.00
6	10	6	0.00	274.0	0.00	21.6	0.00	33.3	0.00
		7	0.00	282.0	0.00	14.1	0.01	45.7	0.01
		8	0.00	351.0	0.00	23.8	0.00	78.0	0.00
		9	0.00	227.0	0.00	17.6	0.00	25.3	0.00
		10	0.00	218.0	0.00	16.3	0.84	45.7	0.84
10	10	11	2.66	3598.0	0.29	115.1	-0.23	269.0	-0.52
		12	5.39	3600.0	-0.86	89.1	-1.52	341.7	-0.66
		13	12.11	3602.0	-2.44	186.9	-4.23	306.0	-1.83
		14	6.83	3599.0	-0.17	88.9	-0.57	167.7	-0.40
		15	19.30	3589.0	-5.04	108.9	-5.04	277.7	0.00
3			0.00	9.2	0.00	5.0	0.00	2.8	0.00
6			0.00	270.4	0.00	18.7	0.17	51.0	0.17
10			9.26	3597.6	-1.64	117.8	-2.32	272.4	-0.68

Fourth class of instances, with a greater impact of the production costs, are the ones that MILP achieves the lowest optimality gap and run time. As presented in Table 5.6, 12 out of 15 instances, MILP found the optimal solution, whereas the rest 3 instances are solved with less than 3% optimality gap. Next to that, relative differences of algorithms are the lowest according to the other experiments as well. This success rate of the algorithms can be explained with the easiness of the lot-sizing problem with higher impact than inventory and routing parts. TIH performs 0.0% and 0.15% average optimality gaps for the 3- and 6-period instances, while HGA finds 0.0% and 0.06% gaps. For the 10 periods, algorithms slightly outperform the MILP by -0.09% and -0.31% average relative differences. Both algorithms are more efficient in terms of the run times comparing with MILP. Looking at the relative differences between TIH and HGA, HGA is moderately more efficient compared to TIH. They both achieve 0.0% optimality gaps for 3-period instances, but HGA finds results with average deviations of -0.04 and -0.28% from the average TIH for higher periods. Also, HGA consistently finds better solutions than TIH even though the deviations are negligible. Finally, TIH outperforms other solution methods in terms of the run time. Note that HGA run times are still reasonably short for the small size instances with an average of 89 seconds overall.

Table 5. 6: TIH and HGA performances on class IV-small instances

T	N	Instance	CPLEX		TIH		HGA		
			Gap(%)	Time	Rel. Diff. (%)*	Time	Rel. Diff. (%)*	Time	Rel. Diff (%)
3	10	1	0.00	6.0	0.00	5.0	0.00	2.0	0.00
		2	0.00	6.0	0.00	5.1	0.00	2.1	0.00
		3	0.00	8.0	0.00	5.1	0.00	2.0	0.00
		4	0.00	11.0	0.00	5.1	0.00	6.0	0.00
		5	0.00	7.0	0.00	5.1	0.00	2.2	0.00
6	10	6	0.00	120.0	0.00	22.5	0.00	32.7	0.00
		7	0.00	56.0	0.00	13.1	0.00	48.3	0.00
		8	0.00	61.0	0.31	25.7	0.19	71.3	-0.04
		9	0.00	48.0	0.00	16.7	0.00	26.3	0.00
		10	0.00	35.0	0.42	16.3	0.13	46.7	-0.18
10	10	11	0.00	2080.0	0.18	113.1	0.00	199.7	-0.18
		12	0.00	994.0	0.17	96.0	0.28	302.0	-0.10
		13	0.90	3600.0	0.28	193.9	-0.04	227.0	-0.31
		14	1.44	3601.0	-0.65	80.8	-0.90	136.3	-0.24
		15	2.86	3597.0	-0.41	101.0	-0.91	230.3	-0.60
3			0.00	7.6	0.00	5.0	0.00	2.8	0.00
6			0.00	64.0	0.15	18.9	0.06	45.1	-0.04
10			1.04	2774.4	-0.09	117.0	-0.31	219.1	-0.28

5.3.2 Performance Analysis of TIH and HGA for Medium and Large Instances

After having shown that both of TIH and our HGA are computationally efficient and outputs high quality solutions for the small instances, we compare the performances for the medium and large size instances. In this context, a general overview of the comparison, in terms of the relative differences of average objective values, for the adapted Two-Phase Iterative Heuristic and our Hybrid Genetic Algorithm is presented in Table 5.7. In addition, average run time for each period parameter (T) is depicted in Figure 5.1. The details of the runs are shown in Tables 5.8-5.13.

It can be seen from Table 5.7 that HGA outperforms TIH in all the instance sets without exception. Amount of the relative difference depends on the instance characteristics such as the class of the instances, periods and number of retailers. Overall, average relative difference among all instances equals to -4.14%. However, if we neglect

Table 5. 7: Average relative differences (%) of the objective values of TIH and HGA runs on medium and large instances.

Class	T	N				Avg
		20	30	40	50	
I	3	-2.36	-1.30	-1.28	-1.67	-1.65
	6	-4.14	-6.73	-6.89	-3.55	-5.33
	10	-2.58	-5.03	-2.56	-	-3.39
	Avg	-3.03	-4.35	-3.58	-2.61	-3.46
II	3	-1.97	-1.53	-1.60	-6.26	-2.84
	6	-9.21	-7.44	-12.88	-10.38	-9.98
	10	-9.58	-7.66	-8.60	-	-8.61
	Avg	-6.92	-5.54	-7.69	-8.32	-7.14
III	3	-0.59	-0.90	-1.00	-3.36	-1.47
	6	-9.12	-9.41	-6.43	-5.94	-7.72
	10	-7.93	-4.56	-5.54	-	-6.01
	Avg	-5.88	-4.96	-4.32	-4.65	-5.07
IV	3	-0.99	-0.96	-0.55	-0.42	-0.73
	6	-1.81	-0.85	-1.23	-0.75	-1.16
	10	-0.32	-0.90	-1.05	-	-0.76
	Avg	-1.04	-0.90	-0.94	-0.58	-0.88
	3	-1.48	-1.18	-1.11	-2.93	
	6	-6.07	-6.10	-6.86	-5.16	
	10	-5.10	-4.54	-4.44	-	
	Total	-4.22	-3.94	-4.13	-4.04	-4.14

the fourth class, with greater impact of production costs, relative difference is widened to -5.22%. Similar to the results of the small instances on the fourth class, experiments on the medium and large fourth-class instances yield nearly the same average relative difference values due to the relative easiness of determining production decisions. Therefore, we can conclude that HGA makes the least difference in terms of the objective value for the fourth class with the overall relative difference of -0.88%. Note that this difference may differ depending on the production impact level, in our case the base scenario production parameters were multiplied by 5. Furthermore, results differ mainly over time horizon lengths. The largest difference in class IV occurred for the instances

with 6 periods whereas the smallest difference is seen for 3 periods with -0.73%. We see the largest difference with -7.14% overall average relative gap within the second class of instances. This can be partially explained by the relatively large searching effort of our HGA over visit decisions (i.e., section 4.1.2) as the population-based structure in addition to the genetic operators of the algorithm enables it. The second largest overall difference is obtained, by the relative difference of -5.07% for the third class of instances where inventory costs of the retailers are removed from the objective function. The value of the relative difference is ranging between -1.47% (refers to 3 periods) and -7.72% (refers to 6 periods) over parameter T . Then, for the base class in which more balanced cost structure is considered, HGA outperforms TIH with -3.46% average relative difference. Before going on with details of the experiments, in terms of the solution quality indicated by the relative objective value differences, we can see in general that as the impact of the production cost with respect to the inventory and transportation costs rises, effectiveness of the HGA against TIH decreases. Moreover, as the transportation costs increases, HGA becomes more predominant. In the following, detailed computational experiments are discussed as presented in Tables 5.8-5.13. Evaluation of the medium and large instance results are made separately for each period, since period parameter T is the main indicator for run times as can be seen in Figure 5.1. It is shown that HGA clearly outperforms TIH for each T value. Regardless of the number of retailers, as the T increases, computational effort for each algorithm increases as well. For the instances with 3, 6 and 10 periods TIH averages 653.1, 997.58 and 1774.8 seconds, whereas HGA performs with 277.1, 491.8 and 688.1 seconds, respectively.

Tables 5.8-5.13 show the performances of the algorithms for each instance. Since the structure of each table is same, we introduce each column generically. The first column (Class) represents the instance class, the second and third column denotes the number of time periods and the number of retailers ($T - N$), respectively. Fourth column represents the instance number (Instance). Fifth and sixth columns show the average objective function value out of 5 runs. Seventh column (Rel. Diff (%)) represents the relative difference that compare column fifth and sixth, as in the previous tables. If the value of the column is less than 0.0, HGA is considered to outperform TIH. Last two columns show the average run time of each algorithm.

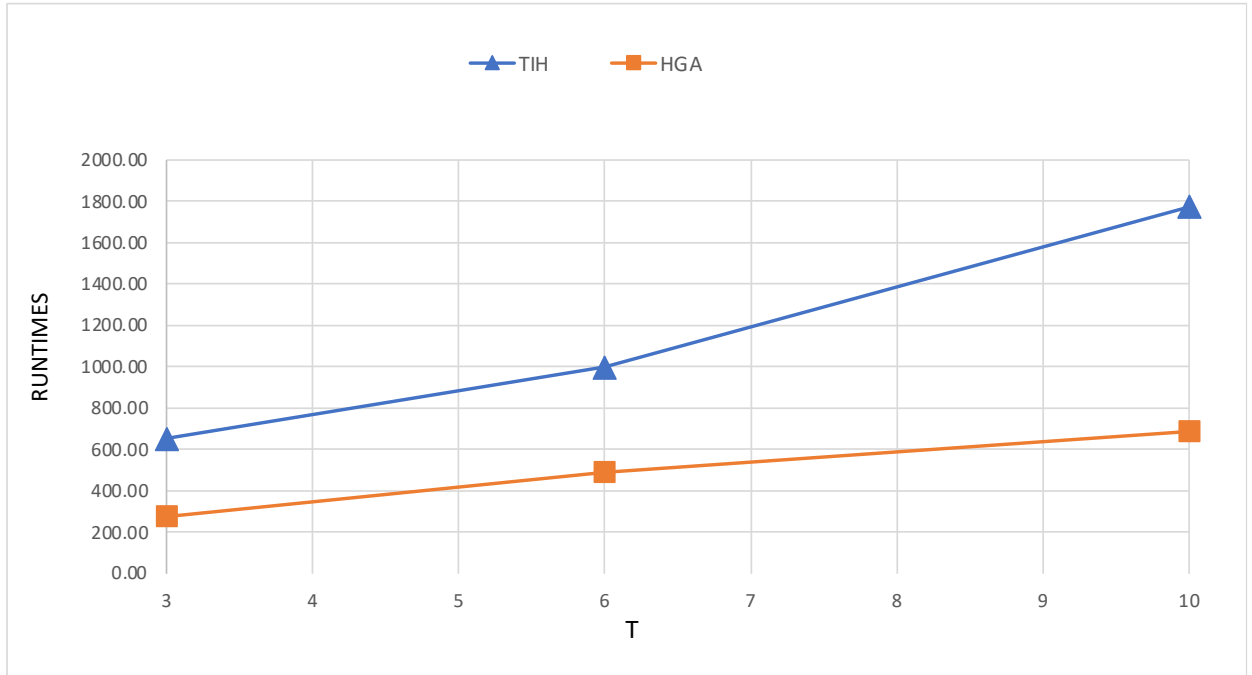


Figure 5. 1: Average Run times (seconds) of TIH and HGA for 3, 6 and 10 periods

In Tables 5.8-5.9, performances of the algorithms in all instances with 3 periods are shown. We see that for the retailer numbers greater than or equal to 20, HGA surpassed TIH in 65 out of 80 instances overall. In 12 instances, TIH yielded the same average objective function value as HGA. In only 3 instances, TIH was able to perform better than HGA. The most dominant performance of HGA is seen for the instances with 50 retailers with a gap of -2.93%. However, there is no linear correlation between the retailer size and the amount of the relative gap, as the second highest relative difference is observed with 20 retailers with a gap of -1.48%. The relative difference between 20 and 40 retailer instances varies within the interval $[-1.48, -1.11]$. Meaning that, TIH starts to perform worse specifically after 50 retailers due to its less successful searching ability, especially for the second class of instances. Instances with 20, 30 and 40 retailers can be considered as medium which may result in similar performances regardless of retailer size for 3 periods. Furthermore, run time of each algorithm gets higher as the number of retailers is increased, as expected. Overall, TIH underperforms HGA in terms of average run time where the most difference occurred in second class with 666.6 seconds average difference.

Table 5. 8: TIH and HGA performances on class I-II instances with 3 periods

Class	T	N	Instance	Objective Value			Time	
				TIH	HGA	Rel. Diff (%)	TIH	HGA
I	3	10	1	13911.32	13911.32	0.00	5.0	4.0
			2	7864.54	7864.54	0.00	4.1	6.0
			3	14686.69	14686.68	0.00	6.1	7.0
			4	10675.37	10675.35	0.00	5.1	8.0
			5	9638.71	9638.71	0.00	5.1	2.0
		20	16	15073.36	14820.18	-1.68	233.8	118.0
			17	19780.15	18508.74	-6.43	161.6	100.0
			18	20062.68	19636.68	-2.12	112.3	108.0
			19	26577.12	26166.00	-1.55	256.5	103.7
			20	30147.65	30147.65	0.00	166.7	188.0
		30	31	33886.57	33886.57	0.00	695.9	157.0
			32	36935.71	36548.61	-1.05	795.2	197.3
			33	40041.94	39956.74	-0.21	686.0	207.0
			34	21598.73	21227.26	-1.72	825.0	234.0
			35	16877.68	16281.96	-3.53	759.9	211.0
		40	46	28618.96	28686.37	0.24	840.4	111.7
			47	39598.02	38300.15	-3.28	859.6	205.0
			48	25598.26	24955.11	-2.51	835.3	127.3
			49	48028.15	47772.01	-0.53	822.9	305.0
			50	32114.52	32014.73	-0.31	871.8	106.3
		50	61	43798.99	42239.50	-3.56	1072.2	797.3
			62	23843.78	23770.20	-0.31	1094.5	942.3
			63	58560.15	57918.56	-1.10	1234.2	975.7
			64	56035.34	54566.57	-2.62	1574.9	958.0
			65	30088.5	29851.60	-0.79	1414.0	1008.0
		Avg				-1.65	613.5	287.5
II	3	10	76	25315.32	25315.32	0.00	5.0	6.0
			77	19356.54	19356.54	0.00	5.1	4.0
			78	27360.28	27360.28	0.00	5.1	5.0
			79	27232.94	27232.94	0.00	5.1	8.0
			80	22177.71	22177.71	0.00	6.1	6.0
		20	91	34303.37	34303.37	0.00	261.6	120.0
			92	35166.06	35166.06	0.00	211.2	126.0
			93	41704.57	39285.68	-5.80	159.6	142.3
			94	53056.00	50994.62	-3.89	407.9	137.0
			95	53114.574	53031.87	-0.16	163.6	195.3
		30	106	54748.57	54748.57	0.00	751.4	163.0
			107	58774.71	58726.71	-0.08	1161.5	230.3
			108	62046.54	62046.54	0.00	960.0	226.3
			109	48473.67	47982.29	-1.01	893.3	257.7
			110	41997.79	39237.95	-6.57	1181.4	279.0
		40	121	60192.66	59749.28	-0.74	1145.5	127.3
			122	66596.99	66924.99	0.49	1085.6	219.0
			123	54164.11	52781.11	-2.55	992.5	152.3
			124	75876.15	74902.76	-1.28	2049.7	364.3
			125	60023.85	57671.03	-3.92	1703.4	127.7
		50	136	79379.32	74464.76	-6.19	1465.7	394.0
			137	58375.73	53131.97	-8.98	1205.1	330.9
			138	93249.44	88051.58	-5.57	1733.0	295.0
			139	86834.59	84938.26	-2.18	1886.7	293.9
			140	66063.63	60551.41	-8.34	1770.7	340.9
		Avg				-2.84	848.6	182.1

Table 5. 9: TIH and HGA performances on class III-IV instances with 3 periods

Class	T	N	Instance	Objective Value		Rel. Diff (%)	Time	
				TIH	HGA		TIH	HGA
III	3	10	151	11117.60	11117.60	0.00	5.0	2.0
			152	4575.60	4575.60	0.00	5.1	2.0
			153	11083.52	11083.52	0.00	5.1	2.0
			154	7347.70	7347.70	0.00	5.1	6.0
			155	4554.40	4554.40	0.00	5.1	2.0
		20	166	7074.50	7074.50	0.00	293.9	163.7
			167	11750.40	11750.40	0.00	208.1	137.7
			168	12611.30	12611.30	0.00	141.4	106.3
			169	18104.50	17600.65	-2.78	269.9	107.7
			170	22560.956	22522.69	-0.17	220.2	225.3
		30	181	24590.70	24590.70	0.00	639.5	178.0
			182	25499.40	25215.77	-1.11	688.2	240.3
			183	26368.00	26368.00	0.00	710.2	220.7
			184	8912.55	8594.60	-3.57	698.5	271.7
			185	6963.90	6974.90	0.16	602.3	289.3
		40	196	13651.80	13650.80	-0.01	842.2	215.0
			197	23114.00	22932.92	-0.78	901.6	207.7
			198	11796.50	11679.50	-0.99	815.7	158.3
			199	33523.05	33288.73	-0.70	893.9	327.7
			200	18051.2	17594.20	-2.53	995.8	110.0
		50	211	25083.1	25036.10	-0.19	1348.4	1140.7
			212	8650.8	7875.48	-8.96	1182.4	1340.7
			213	40298.75	39096.25	-2.98	909.3	1027.3
			214	35002.5	34796.40	-0.59	981.0	1142.3
			215	11830.5	11345.60	-4.10	1770.7	1188.0
		Avg				-1.47	605.5	352.5
IV	3	10	226	47417.72	47417.72	0.00	5.0	2.0
			227	14514.94	14514.94	0.00	5.0	2.0
			228	45766.69	45766.69	0.00	5.0	2.0
			229	25102.37	25102.37	0.00	5.0	6.0
			230	15520.31	15520.31	0.00	5.0	2.0
		20	241	25135.36	24882.18	-1.01	243.4	127.0
			242	49405.75	48134.34	-2.57	154.5	99.7
			243	51039.88	50613.88	-0.83	115.1	108.0
			244	74066.12	73666.67	-0.54	269.9	95.0
			245	99823.25	99823.25	0.00	159.6	199.0
		30	256	113099.94	111785.37	-1.16	598.6	161.3
			257	116663.01	116420.01	-0.21	554.4	182.7
			258	125665.94	125580.74	-0.07	713.8	212.3
			259	28598.50	28360.70	-0.83	792.1	213.0
			260	23533.28	22937.56	-2.53	604.7	194.7
		40	271	54256.76	53941.61	-0.58	692.5	98.3
			272	98680.23	97846.63	-0.84	739.8	211.0
			273	47877.88	47337.11	-1.13	819.1	120.3
			274	154640.35	154410.21	-0.15	776.0	296.0
			275	80678.8	80627.53	-0.06	659.9	108.0
		50	286	112578.19	112532.28	-0.04	1008.8	727.3
			287	29700.89	29601.71	-0.33	1182.4	975.3
			288	186195.15	185389.56	-0.43	1234.2	1034.3
			289	164013.61	163941.61	-0.04	1197.6	908.0
			290	45472.97	45042.49	-0.95	1071.8	1071.7
		Avg				-0.72	544.5	286.3

Results of the instances with 6 periods are presented in Tables 5.10-5.11. TIH underperforms in all instances comparing with HGA. In 6-period instances, with retailer size greater than 20, HGA clearly outperforms TIH in terms of objective value and run time. This time, HGA was the most dominant to TIH for instances with 40 retailers with an average of -6.86% relative difference. This time, unlike the 3-period instances, the second highest relative difference belongs to the instances with 30 retailers with -6.10% while the least difference is seen at 50 retailer instances with -5.16% average. Finally, difference in terms of run times becomes 505.7 seconds, as also depicted in Figure 5.1.

Finally, performances of the algorithms for the largest instances are shown in Tables 5.12-5.13. HGA performs better than TIH in 59 out of 60 instances. A single large instance, with instance number 251, that TIH slightly outputs better average relative difference belongs to the fourth instance class with 0.58%. Nevertheless, HGA still clearly surpasses TIH in this set of instances too. It is not unexpected to see that underperforming HGA results came out of the least successful class so far as discussed in earlier section. In addition, the largest relative gap between algorithms is observed for the instances with 20 retailers by -5.1% average. Even though the instance size increases as the number of retailers grows, relative difference trend slightly goes in the opposite way. For the 30 and 40 retailers, HGA performs still considerably better than TIH but decreases in terms of the relative difference with -4.54% and -4.44%, respectively. Considering that the least effective class for using HGA over TIH is the class four, if we only focus on the large instances of the first three classes, the relative differences rise to -6.70%, -5.75% and -5.57%, respectively. This situation leaves an open door for the future research to improve the searching ability of the algorithm for the instances with 10 periods, together with the production cost weighted instances in general. On the other hand, regarding run times, gap is at its highest level where HGA obviously outperform TIH with averages of 1774.3 and 688.1 seconds, respectively.

Table 5. 10: TIH and HGA performances on class I-II instances with 6 periods

Class	T	N	Instance	Objective Value			Time	
				TIH	HGA	Rel. Diff (%)	TIH	HGA
I	6	10	6	14970.30	14970.30	0.00	21.6	33.3
			7	30294.60	30294.60	0.00	14.1	45.7
			8	20064.01	20064.01	0.00	23.8	78.0
			9	21093.71	21093.71	0.00	17.6	25.3
			10	20855.22	20855.22	0.00	16.3	45.7
		20	21	68181.357	66842.67	-1.96	571.7	399.3
			22	59850.548	56670.82	-5.31	488.8	228.7
			23	36735.926	35644.35	-2.97	459.4	280.3
			24	29496.11	28253.37	-4.21	506.0	460.7
			25	25534.976	23941.26	-6.24	338.4	368.3
		30	36	44090.23	40497.18	-8.15	973.6	462.3
			37	98189.73	91521.55	-6.79	918.8	458.7
			38	74791.91	71537.20	-4.35	1053.7	483.0
			39	48671.57	46926.15	-3.59	994.0	361.7
			40	58672.34	52365.32	-10.75	901.5	321.3
		40	51	54598.37	51715.07	-5.28	2066.0	890.3
			52	139527.42	130802.06	-6.25	1432.4	868.3
			53	76103.16	71063.20	-6.62	1501.4	850.7
			54	100901.403	91841.85	-8.98	1863.1	666.3
			55	96996.4525	89915.89	-7.30	1396.7	868.7
		50	66	161881.68	158631.94	-2.01	2303.8	890.0
			67	58056.317	56627.28	-2.46	2041.6	887.0
			68	131800.605	125136.10	-5.06	2133.1	1001.0
			69	70654.4485	67138.19	-4.98	1998.1	967.0
			70	149464.425	144632.87	-3.23	2112.0	992.0
		Avg				-5.33	1094.6	409.8
II	6	10	81	41112.01	41112.01	0.00	20.6	38.7
			82	49996.5	50046.50	0.10	14.1	58.0
			83	46396.1	46153.73	-0.52	25.7	107.3
			84	50241.06	49818.55	-0.84	17.6	31.7
			85	46581.33	46581.33	0.00	16.3	61.7
		20	96	117770.453	113150.14	-3.92	639.3	245.7
			97	105739.292	96246.91	-8.98	558.3	321.7
			98	85059.836	76275.64	-10.33	589.8	367.0
			99	72117.37	63771.03	-11.57	670.5	294.0
			100	71386.43	63367.75	-11.23	496.9	183.0
		30	111	97196.24	91697.12	-5.66	992.8	484.0
			112	148780.30	135100.77	-9.19	1071.6	528.7
			113	128164.15	116850.29	-8.83	990.7	616.7
			114	103893.40	95531.87	-8.05	1141.2	399.0
			115	108589.68	102671.20	-5.45	998.9	348.7
		40	126	142558.48	122895.24	-13.79	1439.0	878.0
			127	251762.34	217036.50	-13.79	1604.5	874.3
			128	160459.48	139529.99	-13.04	1472.0	851.0
			129	184647.23	164863.60	-10.71	1799.6	894.3
			130	190874.07	165977.45	-13.04	1798.6	866.0
		50	141	256971.10	227408.05	-11.50	2254.3	891.7
			142	148361.40	133159.42	-10.25	2051.0	968.3
			143	219676.77	198479.15	-9.65	2099.8	876.2
			144	150599.13	136899.83	-9.10	2041.1	951.3
			145	245335.15	217352.13	-11.41	2028.6	831.8
		Avg				-9.98	1073.3	518.7

Table 5. 11: TIH and HGA performances on class III-IV instances with 6 periods

Class	T	N	Instance	Objective Value		Rel. Diff (%)	Time	
				TIH	HGA		TIH	HGA
III	6	10	156	7554.56	7554.56	0.00	21.6	43.3
			157	22535.54	22537.54	0.01	14.1	47.0
			158	11735.64	11735.64	0.00	23.8	79.3
			159	12765.00	12765.00	0.00	17.6	30.7
			160	14099.60	14218.32	0.84	16.3	54.7
		20	171	54391.46	52178.73	-4.07	729.2	451.7
			172	41373.7	39026.69	-5.67	514.0	288.0
			173	22639.25	20415.58	-9.82	463.6	339.0
			174	17319.44	15188.34	-12.30	634.5	557.3
			175	11031.8	9518.04	-13.72	419.2	409.7
		30	186	27943.30	25075.70	-10.26	981.6	464.0
			187	77212.36	71654.80	-7.20	862.5	579.7
			188	55937.95	50543.95	-9.64	995.9	671.7
			189	26490.10	23250.10	-12.23	918.6	432.3
			190	40691.18	37559.60	-7.70	906.0	410.0
		40	201	30642.816	27779.6745	-9.34	1354.2	618.2
			202	116385.889	111205.718	-4.45	1260.4	356.6
			203	56163.675	52107.9023	-7.22	966.2	508.1
			204	75069.6275	71817.7872	-4.33	1294.7	616.4
			205	73598.5075	68594.8265	-6.80	1320.8	563.9
		50	216	127123.75	120951.09	-4.86	2175.2	919.0
			217	26994.7	25485.08	-5.59	1911.7	948.2
			218	100286.3	94777.03	-5.49	2059.8	884.8
			219	39981.176	37243.77	-6.85	1985.5	960.5
			220	118261.75	110063.00	-6.93	2018.0	847.6
		Avg				-7.72	954.6	483.3
IV	6	10	231	19506.01	19506.01	0.00	22.5	32.7
			232	98748.13	98748.13	0.00	13.1	48.3
			233	44986.69	45006.36	0.04	25.7	71.3
			234	47723.71	47723.71	0.00	16.7	26.3
			235	54981.94	54882.52	-0.18	16.3	46.7
		20	246	234714.374	234084.31	-0.27	448.4	422.3
			247	170736.327	168674.79	-1.21	453.2	240.3
			248	83745.818	82489.16	-1.50	432.3	263.3
			249	53843.09	52159.47	-3.13	516.0	432.3
			250	33370.688	32385.40	-2.95	365.6	339.7
		30	261	96766.54	95771.99	-1.03	885.8	445.0
			262	352441.41	351169.05	-0.36	919.1	458.0
			263	251506.42	250320.52	-0.47	879.0	502.0
			264	115393.70	113975.73	-1.23	893.4	360.3
			265	167744.60	166414.78	-0.79	935.7	333.3
		40	276	114051.57	111162.68	-2.53	1562.7	581.4
			277	523981.44	520323.04	-0.70	1002.5	556.6
			278	241875.12	238784.38	-1.28	1039.8	465.6
			279	337888.18	334908.75	-0.88	981.9	598.0
			280	315219.05	312840.09	-0.75	1482.0	556.2
		50	291	593021.74	590160.88	-0.48	1998.1	873.1
			292	102000.18	100513.50	-1.46	2003.8	948.2
			293	470157.16	467888.57	-0.48	1967.8	840.5
			294	168612.77	166925.13	-1.00	1980.2	941.3
			295	545137.64	543394.62	-0.32	2016.6	822.2
		Avg				-0.96	767.9	391.0

Table 5. 12: TIH and HGA performances on class I-II instances with 10 periods

Class	T	N	Instance	Objective Value			Time	
				TIH	HGA	Rel. Diff (%)	TIH	HGA
I	10	10	11	70836.20	70428.86	-0.58	240.4	200.3
			12	64017.29	64178.47	0.25	134.6	318.3
			13	45910.63	45678.24	-0.51	313.1	219.5
			14	65463.01	65364.14	-0.15	140.4	139.3
			15	24852.18	24597.41	-1.03	116.8	227.3
		20	26	127445.78	125416.86	-1.59	1593.4	562.7
			27	104729.04	102266.96	-2.35	1515.0	585.0
			28	137686.29	134297.59	-2.46	1167.0	629.7
			29	133073.69	130447.69	-1.97	1930.1	488.0
			30	41668.27	39778.76	-4.53	2022.0	471.3
		30	41	191622.41	184463.58	-3.74	2021.0	666.3
			42	99509.89	94358.45	-5.18	2389.6	760.3
			43	66662.64	62205.49	-6.69	2218.6	713.3
			44	58881.69	56662.56	-3.77	2033.2	782.0
			45	97700.32	92056.34	-5.78	2393.9	910.0
		40	56	216753.28	213043.68	-1.71	2481.0	922.0
			57	95467.42	91903.92	-3.73	2002.0	967.7
			58	202994.14	200185.53	-1.38	2002.0	940.0
			59	245902.19	238599.66	-2.97	2002.0	992.0
			60	174635.35	169372.33	-3.01	2002.0	872.7
Avg						-3.39	1453.7	562.5
II	10	10	86	111546.82	110118.53	-1.28	119.2	231.3
			87	98803.07	98192.44	-0.62	97.0	316.8
			88	89389.92	88001.41	-1.55	182.8	300.0
			89	120228.75	116045.75	-3.48	84.8	147.5
			90	64822.26	63398.02	-2.20	100.0	310.9
		20	101	198798.486	191391.86	-3.73	1671.6	779.0
			102	196361.992	169398.23	-13.73	2039.4	859.0
			103	238933.659	227176.22	-4.92	1545.3	1303.7
			104	211906.842	205901.44	-2.83	2637.8	560.3
			105	126233.09	97598.56	-22.68	2405.8	507.7
		30	116	282212.22	251722.80	-10.80	2950.2	1039.7
			117	202986.96	193074.04	-4.88	2413.9	793.0
			118	160974.368	151495.32	-5.89	2618.3	678.7
			119	158099.51	152152.46	-3.76	2765.6	1043.0
			120	189833.541	175231.50	-7.69	2417.4	1379.7
		40	131	368746.54	349861.79	-5.12	2632.1	921.0
			132	238971.72	228011.43	-4.59	2386.1	919.0
			133	329639.32	303669.66	-7.88	2930.8	1002.0
			134	371272.43	319560.11	-13.93	3076.5	994.0
			135	307605.44	272304.38	-11.48	2758.9	939.0
Avg						-8.26	1891.7	751.3

Table 5. 13: TIH and HGA performances on class III-IV instances with 10 periods

Class	T	N	Instance	Objective Value			Time	
				TIH	HGA	Rel. Diff (%)	TIH	HGA
III	10	10	161	56759.32	56462.32	-0.52	119.2	269.0
			162	50258.40	49924.60	-0.66	97.0	341.7
			163	33865.40	33244.24	-1.83	182.8	306.0
			164	52165.40	51955.40	-0.40	84.8	167.7
			165	13634.37	13654.04	0.14	100.0	277.7
		20	176	110064.24	105620.92	-4.04	2140.2	679.3
			177	84154.80	78611.12	-6.59	1606.8	775.3
			178	119276.61	111701.04	-6.35	1430.2	823.3
			179	114677.04	109754.48	-4.29	2027.0	621.7
			180	22814.98	18626.07	-18.36	2345.2	614.0
		30	191	164099.63	161042.90	-1.86	2485.6	937.0
			192	72444.45	69625.45	-3.89	2296.7	1065.7
			193	38398.59	35998.60	-6.25	2462.7	791.3
			194	28458.00	26416.87	-7.17	2460.7	1064.0
			195	66449.20	64054.61	-3.60	3215.7	1489.0
		40	206	179032.36	170648.117	-4.68	2982.7	939.4
			207	59321.55	53362.5917	-10.05	2756.1	965.0
			208	163208.20	155521.644	-4.71	2955.3	1012.0
			209	198590.35	192140.463	-3.25	2952.8	1003.9
			210	138271.42	131335.464	-5.02	3858.8	957.8
		Avg				-6.01	1928.0	755.0
IV	10	10	236	238094.99	238026.91	-0.03	115.1	199.7
			237	218011.77	217784.68	-0.10	89.1	302.0
			238	128093.494	127911.18	-0.14	186.9	227.0
			239	218764.646	218317.48	-0.20	88.9	136.3
			240	41367.34	41870.55	1.22	108.9	230.3
		20	251	460439.214	463127.01	0.58	1687.7	572.0
			252	347775.272	345348.98	-0.70	1498.7	579.0
			253	490368.445	487547.57	-0.58	1213.0	790.0
			254	485863.048	483416.48	-0.50	2106.4	506.3
			255	68298.276	68022.29	-0.40	1820.0	474.7
		30	266	743075.93	741031.37	-0.28	2021.0	627.7
			267	300155.712	296089.49	-1.35	2155.3	808.3
			268	147812.007	146273.35	-1.04	2018.8	670.3
			269	98025.91	96679.54	-1.37	2033.2	766.7
			270	268582.776	267292.41	-0.48	2417.4	988.3
		40	281	803623.00	802158.31	-0.18	3031.5	892.4
			282	223782.16	222974.84	-0.36	3233.0	965.0
			283	735968.76	731522.53	-0.60	3028.2	961.4
			284	909054.74	905336.62	-0.41	3049.8	983.9
			285	595056.12	573202.54	-3.67	2926.1	929.0
		Avg				-0.76	1304.0	525.2

5.3.3 Managerial Insights

In this section, after demonstrating that our HGA outperformed the adapted TIH algorithm in each class of instances in terms of an integrated approach of simultaneously optimizing overall cost, we compare again the performances of the algorithms regarding the cost components that belong to the main objective function of PPIRP, which are production (including set-up costs), inventory and routing. We discuss the advantages and disadvantages of using each algorithm in different management approaches under different conditions of the cost structures represented by the class categorization of instances. All three types of cost features are examined with their relative differences between average algorithm results for each class of instances while the ultimate goal is still minimizing the overall cost. We assume that each class corresponds to different real-life situations that can emerge. Hence, this section presents valuable managerial insights for the decision makers on selecting the solution approach they can use to deal with different operational aspects in supply chains.

5.3.3.1 Insights into the Production-Oriented Decisions

To evaluate the performances of our algorithms under the production-oriented approach, experiments on four different instance classes are examined. Table 5.14 shows the average relative differences of production costs incurred in our algorithms for each instance group in different classes. Similar to earlier computational discussion, negative values indicate the superiority of HGA.

For a balanced distribution of each cost component (class I), TIH outperforms HGA by an average of 1.94% for 3-period instances. For the number of 6 and 10 periods, HGA outputs slightly better average with -0.19% and -0.16%. Overall, in a balanced environment, TIH is able to beat the HGA with a relative difference of 0.53%. Hence, the decision maker can be more comfortable with selecting TIH over HGA for the small periods and can select HGA with a slight difference for larger periods concerning production costs. For class II, in which transportation costs have larger impact, it seems that algorithms cannot beat each other clearly. Overall average of the relative difference is a negligible amount, -0.02%. Hence, the significant factor would be the run time of the

algorithms which may lead the decision maker to select HGA over TIH. Moreover, considering that the decision maker is not concerned about the inventory costs in the retailer storages, as in the class III, HGA is likely to be chosen. For class III, almost in all the instance groups, HGA makes a considerable difference by being defeated only one time out of 15 instance groups. Relative difference varies from -19.89% to 0.88%. For the last condition that represents the higher impact of the production cost, TIH outperforms HGA with a slight difference. Most of the instance group still averages less than 1% differences. We see that the biggest average difference value is obtained in the instance group with 3 periods. Hence, for the 3-period instances, the decision maker may tend to select TIH whereas for the longer periods, although there is no clear superiority, HGA can be selected over TIH concerning the computational effort.

Table 5. 14: Average relative gaps (%) of production costs between TIH and HGA for each instance class

Production Cost					
T	N	Class I	Class II	Class III	Class IV
3	10	0.00	0.00	0.00	0.00
	20	0.00	0.00	0.00	0.00
	30	0.69	0.00	-1.79	0.00
	40	7.07	0.00	0.00	3.62
	50	0.00	0.00	0.00	0.00
Avg		1.94	0.00	-0.45	0.90
6	10	1.25	0.00	-19.89	0.00
	20	2.03	-1.13	-1.33	-1.29
	30	-0.73	0.01	0.00	0.00
	40	-2.06	0.00	0.00	0.00
	50	-0.01	-0.40	0.00	0.00
Avg		-0.19	-0.38	-0.33	-0.32
10	10	1.14	0.41	0.88	0.37
	20	-1.16	1.52	-3.02	1.08
	30	-0.68	-0.68	-0.72	-0.37
	40	0.07	0.00	-0.01	0.00
Avg		-0.16	0.32	-0.72	0.27
Total		0.53	-0.02	-0.50	0.28

5.3.3.2. Insights into the Inventory-Oriented Decisions

Three different classes of instances are evaluated regarding the inventory-oriented consideration. Class III, which simulates a situation with no inventory costs incurred at retailers, is omitted. As can be seen in Table 5.15, HGA outperforms TIH in classes I with -1.12% for an equally distributed cost structures and almost ties with TIH in an environment where large transportation costs are incurred. However, for class II, there is no consistency as the relative difference varies within the interval [-5.26, 9.23]. But in general, HGA can be said to give better results, including the fact that it requires less computational effort. Alternatively, when production costs are larger, in class IV, HGA outperforms TIH, especially for periods 6 and 10, with an overall average of -0.66%. So, for the classes I and IV, it would not be a hard decision to use HGA for achieving lower inventory costs. For the second class of instances, where long distances of transportation take place, HGA does not clearly beat TIH, but it may be useful regarding the computational effort.

Table 5. 15: Average relative gaps (%) of inventory costs between TIH and HGA for each instance class

Inventory Cost					
T	N	Class I	Class II	Class III	Class IV
3	10	0.00	0.00	-	0.00
	20	0.87	-0.70	-	0.87
	30	0.35	2.45	-	-0.20
	40	-2.75	0.10	-	1.08
	50	0.00	0.00	-	-1.55
Avg		-0.38	0.46	-	0.05
6	10	-2.49	-1.79	-	-3.28
	20	-2.86	9.23	-	-0.67
	30	-2.67	-5.26	-	-0.84
	40	0.58	-4.20	-	-0.05
	50	0.00	-0.07	-	-0.03
Avg		-1.24	-0.07	-	-0.40
10	10	0.71	-0.10	-	-1.32
	20	-3.70	1.08	-	-3.84
	30	-0.75	-2.42	-	-1.29
	40	-3.28	-0.09	-	-0.01
Avg		-1.76	-0.48	-	-1.62
Total		-1.12	-0.03	-	-0.66

5.3.3.3 Insights into the Transportation-Oriented Decisions

For a decision maker who concerns about large transportation costs, it is an easy decision to use HGA over TIH as Table 5.16 justifies the argument by depicting relative differences of transportation costs incurred in the solutions of TIH and HGA. For all the instance groups except the one with 6 periods and 10 retailers, HGA obviously outperforms TIH with the average relative gaps of -7.07%, -6.93%, -9.60% and -7.34% for the classes I, II, III and IV, respectively. This results also justifies that the main part that makes the difference of our HGA is the search ability over visit plans and routing decisions comparing with the adapted TIH.

Table 5. 16: Average relative gaps (%) of transportation costs between TIH and HGA for each instance class

Transportation Cost					
T	N	Class I	Class II	Class III	Class IV
3	10	0.00	0.00	0.00	0.00
	20	-9.45	-4.59	-1.53	-9.41
	30	-4.55	-3.11	-1.97	-7.50
	40	-4.49	-3.14	-3.92	-7.12
	50	-4.83	-9.74	-7.38	-6.45
Avg		-5.83	-5.14	-3.70	-7.62
6	10	1.25	-1.87	-0.02	3.36
	20	-11.46	-18.08	-18.84	-6.83
	30	-8.22	-2.69	-8.90	-7.98
	40	-7.38	-10.06	-11.62	-9.13
	50	-6.58	-4.67	-5.50	-5.41
Avg		-8.41	-8.87	-11.21	-7.34
10	10	-7.50	-6.01	-8.54	-3.89
	20	-8.87	-17.54	-22.36	-11.58
	30	-6.88	-2.61	-12.25	-7.73
	40	-4.66	-0.89	-12.39	-5.02
Avg		-6.98	-6.76	-13.89	-7.05
Total		-7.07	-6.93	-9.60	-7.34

Chapter 6:

CONCLUSION

We introduce a two-echelon perishable production inventory routing problem in which a single plant with a production and storage capacity produces a single perishable product with fixed shelf-life. Production, replenishment and distribution decisions are taken jointly by a central decision maker where geographically dispersed retailers have deterministic demands in a finite planning horizon.

We model the problem under general assumptions of MILP and solve the model by using a commercial solver, CPLEX. We cannot obtain any optimal solution to the problem apart from the instances with ten retailers within a reasonable time due to the computational complexity. Thus, a Hybrid Genetic Algorithm (HGA) that combines local search techniques and mathematical programming within a genetic algorithm framework is proposed. As a benchmark algorithm, a common matheuristic named Two-Phase Iterative Heuristic (TIH) is adapted and implemented. We test our proposed HGA and adapted TIH algorithm on a novel instance set that we generated based on the benchmark instances from the literature of perishable inventory routing problem.

Computational experiments on these instances show that both Two-Phase Heuristic and Hybrid Genetic Algorithm are capable of optimally solving small size instances within reasonable computing time. Then, our proposed Hybrid Genetic Algorithm is able to outperform our benchmark algorithm in terms of solution quality and running times in larger instances. Furthermore, we investigate the performances of the solution methods on different scenarios of cost structures to provide managerial insights with respect to the optimal decision making that minimizes the total operational cost. In particular, Hybrid Genetic Algorithm makes a significant difference by determining more efficient delivery decisions due to its unique search capability over visit decisions. Hence, it is even more effective for the instances where transportation costs are larger.

As a future research direction, more efficient heuristics that further explore and improve the quality of the solutions especially for our novel instances with 10 periods can

be developed. Additionally, another future research can extend the proposed problem by considering multiple-perishable products or multi-plant setting under a stochastic environment.



BIBLIOGRAPHY

- Abdelmaguid, T. F., & Dessouky, M. M. (2006). A genetic algorithm approach to the integrated inventory-distribution problem. *International Journal of Production Research*, 44(21), 4445-4464. doi:10.1080/00207540600597138
- Absi, N., Archetti, C., Dauzère-Pérès, S., Feillet, D., 2014. A two-phase iterative heuristic approach for the production routing problem. *Transp. Sci.* 49 (4), 784–795.
- Adulyasak, Y., Cordeau, J.-F., Jans, R., 2014a. Formulations and branch-and-cut algorithms for multivehicle production and inventory routing problems. *INFORMS J. Comput.* 26 (1), 103–120.
- Adulyasak, Y., Cordeau, J. -, & Jans, R. 2014b. Optimization-based adaptive large neighborhood search for the production routing problem. *Transportation Science*, 48(1), 20-45. doi:10.1287/trsc.1120.0443
- Adulyasak, Y., Cordeau, J.-F., Jans, R., 2015a. Benders decomposition for production routing under demand uncertainty. *Oper. Res.* 63 (4), 851–867.
- Adulyasak, Y., Cordeau, J.-F., Jans, R., 2015b. The production routing problem: A review of formulations and solution algorithms. *Comput. Oper. Res.* 55, 141–152.
- Alkaabneh, F., Diabat, A., Gao, H.O., 2020. Benders decomposition for the inventory vehicle routing problem with perishable products and environmental costs. *Comput. Oper. Res.* 113, 104751.
- Alvarez, A., Cordeau, J.-F., Jans, R., Munari, P., Morabito, R., 2020. Formulations, branch-and-cut and a hybrid heuristic algorithm for an inventory routing problem with perishable products. *Eur. J. Oper. Res.* 283 (2), 511–529.
- Alvarez, A., Miranda, P., & Rohmer, S. U. K. (2022). Production routing for perishable products. *Omega (United Kingdom)*, 111 doi:10.1016/j.omega.2022.102667
- Archetti C, Bertazzi L, Paletta G, Speranza MG. Analysis of the maximum level policy in a production-distribution system. *Comput Oper Res* 2011;38(12):1731–46.

- Archetti, C., Boland, N., Grazia Speranza, M., 2017. A matheuristic for the multivehicle inventory routing problem. *INFORMS J. Comput.* 29 (3), 377–387.
- Armentano, V.A., Shiguemoto, A.L., Løkketangen, A., 2011. Tabu search with path relinking for an integrated production–distribution problem. *Comput. Oper. Res.* 38 (8), 1199–1209.
- Avci, M., Yildiz, S.T., 2019. A matheuristic solution approach for the production routing problem with visit spacing policy. *Eur. J. Oper. Res.* 279 (2), 572–588.
- Avci, M., Yildiz, S.T., 2020. A mathematical programming-based heuristic for the production routing problem with transshipments. *Comput. Oper. Res.* 123, 105042.
- Azadeh, A., S. Elahi, M. H. Farahani, B. Nasirian, A genetic algorithm-taguchi based approach to inventory routing problem of a single perishable product with transshipment, *Computers Industrial Engineering* 104 (Supplement C) (2017) 124 – 133
- Bard JF, Nananukul N. Heuristics for a multiperiod inventory routing problem with production decisions. *Comput Ind Eng* 2009;57(3):713–23.
- Bard JF, Nananukul N. A branch-and-price algorithm for an integrated production and inventory routing problem. *Comput Oper Res* 2010;37(12):2202–17.
- Bell W., Dalberto L., Fisher M., Greenfield A., Jaikumar R., Mack R. and Prutzman P. (1983) Improving the distribution of industrial gases with an on-line computerized routing and scheduling system. *Interfaces* 13, 4-23.
- Belo-Filho, M.A.F., P. Amorim & B. Almada-Lobo (2015) An adaptive large neighborhood search for the operational integrated production and distribution problem of perishable products, *International Journal of Production Research*, 53:20, 6040-6058,
- Ben Ahmed, M., Hvattum, L. M., Agra, A., The effect of different mathematical formulations on a matheuristic algorithm for the production routing problem, *Computers & Operations Research*, Volume 155, 2023, 106232.

- Boudia M, Louly M, Prins C. A reactive grasp and path relinking for a combined production-distribution problem. *Comput Oper Res* 2007;34(11):3402–19.
- Boudia M, Prins C. A memetic algorithm with dynamic population management for an integrated production-distribution problem. *Eur J Oper Res* 2009;195(3):703–15.
- Brahimi, N., Aouam, T., 2016. Multi-item production routing problem with backordering: a milp approach. *Int. J. Prod. Res.* 54 (4), 1076–1093.
- Chandra, P., 1993, A dynamic distribution model with warehouse and customer replenishment requirements. *Journal of the Operational Research Society*, 44(7), 681±692.
- Chandra, P., & Fisher, M. L. (1994). Coordination of production and distribution planning. *European journal of operational research*, 72(3), 503-517.
- Chekoubi, Zakaria; Trabelsi, Wajdi; Sauer, Nathalie ; Majdouline, Ilias The Integrated Production-Inventory-Routing Problem with Reverse Logistics and Remanufacturing: A Two-Phase Decomposition Heuristic (2022) *Sustainability (Switzerland)*, 14 (20), art. no. 13563
- Chen, H. K., Hsueh, C. F., & Chang, M. S. (2009). Production scheduling and vehicle routing with time windows for perishable food products. *Computers & operations research*, 36(7), 2311-2319.
- Chitsaz, M., Cordeau, J.-F., Jans, R., 2019. A unified decomposition matheuristic for assembly, production, and inventory routing. *INFORMS J. Comput.* 31 (1), 134–152.
- Chu F, Li Y, Feng C, Chu C, Zhou M. Integrated production inventory routing planning for intelligent food logistics systems. *IEEE Trans Intell Transp Syst* 2018;20(3):867–78.
- Coelho, L.C., Laporte, G., 2014. Optimal joint replenishment, delivery and inventory management policies for perishable products. *Comput. Oper. Res.* 47, 42–52.
- Crama, Y., Rezaei, M., Savelsbergh, M., Woensel, T.V., 2018. Stochastic inventory routing for perishable products. *Transp. Sci.* 52 (3), 526–546.

- Dantzig G.B., Ramser J.H. Optimum routing of gasoline delivery trucks (1959) World Petroleum Congress Proceedings, 1959-May, pp. 229 – 238
- Darvish, M., Coelho, L., 2018. Sequential versus integrated optimization: production, location, inventory control, and distribution. *Eur. J. Oper. Res.* 268, 203-214.
- Diabat, A., Abdallah, T., Le, T.: A hybrid tabu search based heuristic for the periodic distribution inventory problem with perishable goods. *Ann. Oper. Res.* (2014).
- Fahimnia, B., Farahani, R.Z., Marian, R., Luong, L., 2013. A review and critique on integrated production–distribution planning models and techniques. *J. Manuf. Syst.* 32 (1), 1–19. FAO, 2011
- Farahani, P., Grunow M. and H.-O. Gunther, Integrated “production and distribution planning for perishable food products, *Flexible Services and Manufacturing Journal* 24(1) (2012), 28–51.
- Federgruen, A., and M. Tzur. 1999. “Time-partitioning Heuristics: Application to One Warehouse, Multiitem, Multiretailer Lot-sizing Problems.” *Naval Research Logistics (NRL)* 46 (5): 463–486.
- Fortes A., Camargo R., Muniz L.R., Lima F.M.D.S., Cota F.D.R. Efficient matheuristics to solve a rich production-routing problem (2022) *Computers and Industrial Engineering*, 171, art. no. 108369
- Fumero, F., Vercellis, C., 1999. Synchronized development of production, inventory, and distribution schedules. *Transp. Sci.* 33 (3), 330–340.
- Geismar, H. N., G. Laporte, L. Lei, C. Sriskandarajah. 2008. The integrated production and transportation scheduling problem for a product with a short life span and non-instantaneous transportation time. *INFORMS J. Comput.* 20(1): 21–33.
- Gen, M., and Syarif, A., “Hybrid genetic algorithm for multi-time period production/distribution planning,” *Comput. Ind. Eng.*, vol. 48, no. 4, pp. 799–809, 2005.

- Hiassat A, Diabat A, Rahwan I. A genetic algorithm approach for location-inventory-routing problem with perishable products. *J. Manuf. Syst* 2017;42:93–103.
- Jia, T., Li, X., Wang, N., Li, R., 2014. Integrated inventory routing problem with quality time windows and loading cost for deteriorating items under discrete time. *Mathematical Problems in Engineering* , 1–14.
- Karakostas, P., Sifaleras, A., Georgiadis, M., 2020. Adaptive variable neighborhood search solution methods for the fleet size and mix pollution location-inventory routing problem. *Expert Syst. Appl.* 153.
- Kumar, R. S., Kondapaneni, K., Dixit, V., Goswami, A., L. Thakur, M. Tiwari, Multi-objective modeling of production and pollution routing problem with time window: A self-learning particle swarm optimization approach, *Computers & Industrial Engineering* 99 (2016) 29–40.
- Le, T., Diabat, A., Richard, J.-P., Yih, Y., 2013. A column generation-based heuristic algorithm for an inventory routing problem with perishable goods. *Optim. Lett.* 7 (7), 1481–1502.
- Lei L, Liu S, Ruszczyński A, Park S. On the integrated production, inventory, and distribution routing problem. *IIE Trans* 2006;38(11):955–70.
- Li, Y., Chu, F., Yang, Z., & Calvo, R. W. (2016). A Production Inventory Routing Planning for Perishable Food with Quality Consideration. *IFAC-PapersOnLine*, 49(3), 407–412.
- Li, Y., Chu, F., Côté, J.-F., Coelho, L.C., Chu, C., 2020. The multi-plant perishable food production routing with packaging consideration. *Int. J. Prod. Econ.* 221, 107472.
- Luo Q., Fan Q., Deng Q., Guo X., Gong G., Liu X. Solving bi-objective integrated scheduling problem of production, inventory and distribution using a modified NSGA-II (2023) *Expert Systems with Applications*, 225, art. no. 120074
- Manousakis, Eleftherios G., et al. "An infeasible space exploring metaheuristic for the production routing problem." *European journal of operational research* 298.2 (2022): 478-495.

- Manousakis, Eleftherios G., Christos D. Tarantilis, and Emmanouil E. Zachariadis. "The cyclic production routing problem." *International Journal of Production Research* (2023): 1-20.
- Melo, R. A., & Wolsey, L. A. (2012). MIP formulations and heuristics for two-level production-transportation problems. *Computers & Operations Research*, 39(11), 2776-2786.
- Miranda, P.L., Cordeau, J.-F., Ferreira, D., Jans, R., Morabito, R., 2018. A decomposition heuristic for a rich production routing problem. *Comput. Oper. Res.* 98, 211–230.
- Mirzaei, S., Seifi, A., 2015. Considering lost sale in inventory routing problems for perishable goods. *Comput. Ind. Eng.* 87, 213–227.
- Moin, N. H., & Yuliana, T. (2015). Three-phase methodology incorporating scatter search for integrated production, inventory, and distribution routing problem. *Mathematical Problems in Engineering*, 2015.
- Mousavi, R., Bashiri, M., & Nikzad, E. (2022). Stochastic production routing problem for perishable products: Modeling and a solution algorithm. *Computers & Operations Research*, 142, 105725.
- Moscato, P., Cotta, C., & Mendes, A. (1999). Memetic algorithms. *Handbook of approximation algorithms and metaheuristics*.
- Neves-Moreira, F., Almada-Lobo, B., Cordeau, J. F., Guimarães, L., & Jans, R. (2019). Solving a large multi-product production-routing problem with delivery time windows. *Omega*, 86, 154-172.
- Qiu, Y., Qiao, J., Pardalos, P.M., 2016. A branch-and-price algorithm for production routing problems with carbon cap-and-trade. *Omega* 68, 49–61.
- Qiu Y, Wang L, Xu X, Fang X, Pardalos PM. A variable neighborhood search heuristic algorithm for production routing problems. *Appl Soft Comput* 2018;66:311–18.

- Qiu, Y., Qiao, J., Pardalos, P.M., 2019. Optimal production, replenishment, delivery, routing and inventory management policies for products with perishable inventory. *Omega* 82, 193–204.
- Qiu, Y., Zhou, D., Du, Y., Liu, J., Pardalos, P. M., & Qiao, J. (2021). The two-echelon production routing problem with cross-docking satellites. *Transportation Research Part E: Logistics and Transportation Review*, 147, 102210.
- Rahimi, M., Baboli, A., & Rekik, Y. (2017). Inventory routing problem for perishable products by considering customer satisfaction and green criteria. In *Dynamics in Logistics: Proceedings of the 5th International Conference LDIC, 2016 Bremen, Germany* (pp. 445-455). Springer International Publishing.
- Russell, R. A. (2017). Mathematical programming heuristics for the production routing problem. *International Journal of Production Economics*, 193, 40-49.
- Rohmer, S., Claassen, G., Laporte, G., 2019. A two-echelon inventory routing problem for perishable products. *Comput. Oper. Res.* 107, 156–172.
- Rong, A., Akkerman, R., & Grunow, M. (2011). An optimization approach for managing fresh food quality throughout the supply chain. *International Journal of Production Economics*, 131(1), 421-429.
- Schenekemberg, C.M., Scarpin, C.T., Pécora Jr., J.E., Guimarães, T.A., Coelho, L.C., 2021. The two-echelon production-routing problem. *Eur. J. Oper. Res.* 288 (2), 436–449.
- Solyalı, O., Süral, H., 2017. A multi-phase heuristic for the production routing problem. *Comput. Oper. Res.* 87, 114–124.
- Soysal, M., Bloemhof-Ruwaard, J.M., Haijema, R., van der Vorst, J.G., 2015. Modeling an inventory routing problem for perishable products with environmental considerations and demand uncertainty. *Int. J. Prod. Econ.* 164, 118–133.
- Torkaman, S., Jokar, M.R.A., Mutlu, N., Van Woensel, T., 2020. Solving a production-routing problem with price-dependent demand using an outer approximation method. *Comput. Oper. Res.* 123, 105019.

- Torkaman, S., Jokar, M. R. A., Mutlu, N., & Van Woensel, T. (2022). Rolling horizon-based heuristics for solving a production-routing problem with price-dependent demand. *Computers & Operations Research*, 148, 105973.
- Vadseth, S. T., Andersson, H., & Stålhane, M. (2021). An iterative matheuristic for the inventory routing problem. *Computers & Operations Research*, 131, 105262.
- Wagner, H. M., & Whitin, T. M. (1958). Dynamic version of the economic lot size model. *Management science*, 5(1), 89-96.
- Wang, S., & Chu, F. (2022). A decomposition-based heuristic method for inventory routing problem. *IEEE Transactions on Intelligent Transportation Systems*, 23(10), 18352-18360.