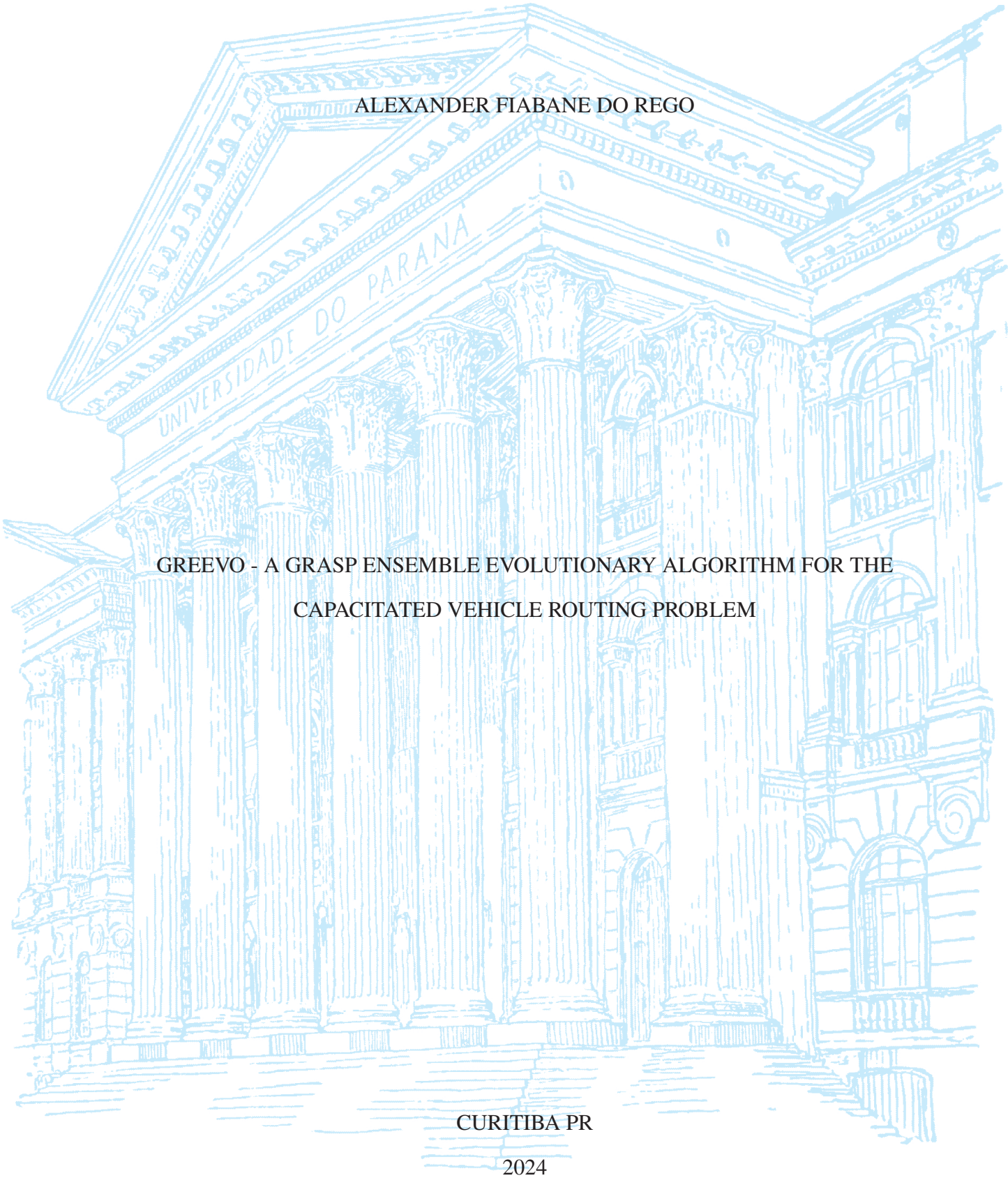
UNIVERSIDADE FEDERAL DO PARANÁ

ALEXANDER FIABANE DO REGO




# GREEVO - A GRASP ENSEMBLE EVOLUTIONARY ALGORITHM FOR THE CAPACITATED VEHICLE ROUTING PROBLEM

CURITIBA PR

2024

ALEXANDER FIABANE DO REGO

# GREEVO - A GRASP ENSEMBLE EVOLUTIONARY ALGORITHM FOR THE CAPACITATED VEHICLE ROUTING PROBLEM

Tese apresentada como requisito parcial à obtenção do grau de Doutor em Ciência da Computação no Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Ciência da Computação*.

Orientador: PhD Aurora Trinidad Ramirez Pozo.

CURITIBA PR

2024

DADOS INTERNACIONAIS DE CATALOGAÇÃO NA PUBLICAÇÃO (CIP)
UNIVERSIDADE FEDERAL DO PARANÁ
SISTEMA DE BIBLIOTECAS – BIBLIOTECA DE CIÊNCIA E TECNOLOGIA

Rego, Alexander Fiabane do
GREEVO - a grasp ensemble evolutionary algorithm for the capacitated vehicle routing problem / Alexander Fiabane do Rego. – Curitiba, 2024.
1 recurso on-line: PDF.

Tese (doutorado) – Universidade Federal do Paraná, Setor de Ciências Exatas, Programa de Pós-Graduação em Informática. Defesa: Curitiba, 04/03/2024.
Orientador: PhD Aurora Trinidad Ramirez Pozo

1. Otimização combinatória. 2. Algorítmos computacionais I. Ramirez Pozo, Aurora Trinidad. II. Título

CDD 005.13

Selma Regina Ramalho Conte
Bibliotecária
CRB-9/888

# TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação INFORMÁTICA da Universidade Federal do Paraná foram convocados para realizar a arguição da tese de Doutorado de **ALEXANDER FIABANE DO REGO** intitulada: **GREEVO - A GRASP ENSEMBLE EVOLUTIONARY ALGORITHM FOR THE CAPACITATED VEHICLE ROUTING PROBLEM**, sob orientação da Profa. Dra. AURORA TRINIDAD RAMIREZ POZO, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de doutor está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

CURITIBA, 04 de Março de 2024.

Assinatura Eletrônica
05/03/2024 11:22:36.0
AURORA TRINIDAD RAMIREZ POZO
Presidente da Banca Examinadora

Assinatura Eletrônica
05/03/2024 12:20:45.0
EDUARDO JAQUES SPINOSA
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Assinatura Eletrônica
06/03/2024 13:21:28.0
ANDRÉ BRITTO DE CARVALHO
Avaliador Externo (UNIVERSIDADE FEDERAL DE SERGIPE)

Assinatura Eletrônica
05/03/2024 11:28:49.0
PAULO HENRIQUE SIQUEIRA
Avaliador Externo (UNIVERSIDADE FEDERAL DO PARANÁ)

*Este trabalho é dedicado aos meus pais: Jorge e Elidia.*

## ACKNOWLEDGEMENTS

Este trabalho só pôde ser realizado e concluído porque recebi auxílio de muitas pessoas que contribuíram direta e indiretamente. Agradeço a Deus e ao Divino Amigo e Mestre Jesus por me sustentarem, principalmente nos momentos difíceis, renovando as minhas forças e permitindo que os amigos queridos me auxiliassem ou influenciassem aqueles que materialmente me ajudaram em todo esse processo. Agradeço à professora Aurora pela oportunidade que me concedeu, pelo convívio e ensinamentos, pela confiança e orientação que foram fundamentais para que eu pudesse alcançar esse objetivo. Agradeço aos meus pais e minha irmã pelo amor, carinho e apoio que jamais deixaram de demonstrar, não só durante esse trabalho, mas ao longo de toda essa caminhada. À minha esposa Indiara, que suportou com paciência e resignação o meu estresse e seguiu dispensando amor, carinho e apoio. Aos amigos do laboratório que, diversas vezes, dedicaram o seu tempo, auxiliando-me com ideias, sugestões e apoio também. Por fim, ao meu filho Benhur, que foi e é o estímulo para que eu concluísse esse trabalho e para que eu siga buscando me aperfeiçoar.

## RESUMO

Desastres, naturais ou não, causam muitos danos em seu entorno. Em geral, afetam edificações, geram impactos negativos na economia e resultam em perda de vidas. Para minimizar e diminuir os efeitos causados por estes, planos de evacuação eficientes e eficazes são essenciais. Nesse sentido, o processo de evacuação pode ser modelado de acordo com um Problema de Roteamento de Veículos (PRV). Este tipo de problema é um problema NP-difícil de otimização combinatorial, que visa minimizar o custo total (em geral, distância percorrida) na utilização de uma frota de veículos para a realização de um determinado serviço. Portanto, em problemas de otimização combinatória, como PRV, metaheurísticas podem obter soluções ótimas ou quase ótimas. No contexto do PRV, muitas abordagens com metaheurística foram propostas nos últimos anos. Algumas dessas abordagens apresentam excelentes desempenhos, tanto em termos de tempo de processamento quanto de qualidade da solução. Recentemente, as técnicas de *ensemble* têm chamado a atenção de pesquisadores de outros problemas de otimização combinatorial. As técnicas de *ensemble* podem tirar vantagem de metaheurísticas robustas existentes através de um mecanismo de seleção inteligente. Desta forma, este trabalho apresenta, diferenciando-se das metodologias convencionais de ensemble, uma abordagem chamada GREEVO (*Greedy Randomized Adaptive Search Procedure Ensemble Evolutionary Algorithm*) para resolver o Problema de Roteamento de Veículos com Capacitade (CVRP). Nossa abordagem combina um GRASP com o algoritmo *Split* e usa uma Hiper-Heurística com Aprendizagem por Reforço para selecionar heurísticas de alto nível no *ensemble*. Nossos experimentos em benchmarks populares da literatura e um estudo de caso mostraram que nossa abordagem supera outras abordagens comparadas, ao mesmo tempo que fornece *insights* valiosos sobre como utilizar conjuntos e componentes de aprendizado de máquina.



Palavras-chave: Otimização Combinatorial, Metaheurística, Hiper-Heurística, Reforço por Aprendizagem, Problema de Roteamento de Veículo, Evacuação.

## ABSTRACT

Disasters, natural or not, cause a lot of damage to their surroundings. In general, they affect buildings, generate negative impacts on the economy, and result in loss of life. In order to minimize and decrease the effects caused by these, efficient and effective evacuation plans are essential. In this sense, the evacuation process can be modeled according to a Vehicle Routing Problem (VRP). This kind of problem is an NP-hard problem of combinatorial optimization, which aims to minimize the total cost (in general, distance covered) in the use of a vehicle fleet to perform a particular service. Therefore, in combinatorial optimization problems such as VRPs, metaheuristics can obtain optimal or near-optimal solutions. In the VRP context, many approaches that use metaheuristics have been proposed over the past few years. Some of those approaches have excellent performances, both in terms of processing time and quality of the solution. Recently, ensemble techniques have drawn the attention of researchers to other combinatorial optimization problems. Ensemble techniques can take advantage of robust metaheuristics existing through an intelligent selection mechanism. Thus, this work presents, distinguishing from the conventional ensemble methodologies, an approach called GREEVO (Greedy Randomized Adaptive Search Procedure Ensemble Evolutionary Algorithm) to address the Capacitated Vehicle Routing Problem (CVRP). Our approach combines a GRASP with the Split algorithm and uses a Hyper-Heuristic with Reinforcement Learning to select high-level heuristics in the ensemble. Our experiments on the popular literature benchmarks and a case study have shown that our approach outperforms other compared approaches while providing valuable insights on how to utilize ensembles and machine learning components.



Keywords: Combinatorial Optimization, Metaheuristic, Hyper-Heuristic, Reinforcement Learning, Ensemble, Vehicle Routing Problem, Evacuation.

## LIST OF FIGURES

## LIST OF TABLES

## LIST OF ACRONYMS

| | |
|---|---|
| ACO | Ant Colony Optimization |
| AILS-PR | Adaptive Iterated Local Search with Path Relinking |
| ALNS | Adaptive Large Neighborhood Search |
| AVNS | Adaptive Variable Neighborhood Search |
| BEP | Bus Evacuation Problem |
| BKS | Best-Known Solutions |
| CVRP | Capacitated Vehicle Routing Problem |
| DHMDSDVRPTW | Dynamic Heterogeneous Multi-Depot Split Delivery Vehicle Routing Problem with Time Window |
| DOM | Disaster Operations Management |
| DRR | Disaster Risk Reduction |
| EAX | Edge Assembly Crossover |
| FILO | Fast Iterated Local Search Localized Optimization |
| EWT | Emergency Water Trucking |
| GA | Genetic Algorithm |
| SA | Simulated Annealing |
| GAP | deviation from BKS in percentage |
| GIPA | Greedy Insertion based on Probability Assignment |
| GPX | Generalized Partition Crossover |
| GRASP | Greedy Randomized Adaptive Search Procedure |
| GREEVO | GRASP Ensemble Evolutionary Algorithm |
| HGSADC | Hybrid Genetic Algorithm with Adaptive Diversity Control |
| HH | Hyper-Heuristic |
| HRVND | Hierarchical Randomized Variable Neighborhood Descent |
| HVRP | Heterogeneous or Mix fleet Vehicle Routing Problem |
| ILS | Iterated Local Search |
| IP | Integer Programming |
| LMDP | Last Mile Distribution Problem |
| LNS | Large Neighborhood Search |
| LS-VND | Local Search Variable Neighborhood Descent |
| MA | Memetic Algorithm |
| MDVRP | Multi-Depot Vehicle Routing Problem |
| MILP | Mixed Integer Linear Programming |
| MIP | Mixed Integer Programming |

| | |
|---|---|
| MS-ILS-RNVD | Multi-Start Iterated Local Search with a Randomised Variable Neighbourhood Descent |
| PDVRP | Pickup and Delivery Vehicle Routing Problem |
| PIX | Periodic crossover with insertions |
| PSO | Particle Swarm Optimization |
| PVR | Problema de Roteamento de Veículo |
| PVRPTW | Periodic VRPTW |
| RCL | Restricted Candidate List |
| RNVD | Randomised Variable Neighbourhood Descent |
| SDVRP | Split Delivery Vehicle Routing Problem |
| SISR | Slack Induction by String Removals |
| SP | Set Partitioning |
| TC | Terminal Condition |
| TSP | Travel Salesman Problem |
| UNISDR | United Nations International Strategy for Disaster Reduction |
| VND | Variable Neighborhood Descent |
| VNS | Variable Neighborhood Search |
| VRP | Vehicle Routing Problem |
| VRPTW | Vehicle Routing Problem with Time Window |
| WS | WorkStream |

## LIST OF SYMBOLS

| | |
|---|---|
| $G$ | undirected/auxiliary graph |
| $V$ | node-set with the depot |
| $E$ | shortest path between nodes |
| $K$ | homogeneous fleet of vehicles |
| $k$ | vehicle |
| $Q$ | vehicle's capacity |
| $n$ | number of customers |
| $i, j$ | node/customer indexes |
| $c_{ij}, C_{ij}$ | cost of arc/trip (i,j) |
| $s_i$ | service time |
| $L$ | limited working time or duration |
| $A$ | set of arcs |
| $x_{ij}^k$ | binary variable |
| $M$ | big positive constant |
| $t_i^k$ | arrival time |
| $GAB$ | graph |
| $EA, EB$ | edges in $GAB$ |
| $pA, pB, PA, PB$ | parents |
| $m$ | number of routes |
| $P_{red}, P_{blue}$ | tours |
| $W$ | demand |
| $\alpha$ | penalization factor/split rate in SISR |
| $\beta$ | penalization factor/split depth in SISR |
| $\omega^Q, \omega^L$ | penalization factor |
| $N_c$ | set of children (offspring) |
| $I$ | individual |
| $N_c lose$ | set of closest neighbours |
| $\delta^H(I, I_2)$ | Hamming distance between $I$ and $I_2$ |
| $\mu$ | minimum population size |
| $\lambda$ | population size threshold |
| $\xi^{REF}$ | target proportion of entirely feasible individuals |
| $\xi^{L,Q}$ | proportion in the last 100 generated feasible individuals concerning route duration and vehicle capacity, respectively |
| $S, S^*, S^r$ | solution, best overall solution and reference solution/shaken solution/local optimum solution, respectively |

| | |
|---|---|
| $n_{cw}$ | neighbor customers |
| $V_c$ | set of customers |
| $r$ | route |
| $U$ | unrouted customers |
| $P$ | probability threshold |
| $\omega$ | length of a walk/walk |
| $T$ | temperature |
| $T_0$ | initial temperature |
| $\mathfrak{I}$ | set of tours |
| $\tau$ | tour |
| $C$ | cooling constant |
| $l_s^{max}$ | maximum string cardinality in SISR |
| $k_s^{max}$ | maximum number of strings in SISR |
| $k_s$ | number of strings to be removed in SISR |
| $R$ | set of ruined tours in SISR |
| $p$ | position in SISR |
| $\gamma$ | sparsification parameter/blink factor |
| $S_t$ | state |
| $A_t$ | action |
| $R_{t+1}$ | reward |

**CONTENTS**

# 1 INTRODUCTION

Disasters can generate significant damage in their surroundings. In general, they affect buildings, generate negative economic impacts, and result in the loss of human life. Since the 1990s, the international community has studied disasters and their effects (Aitsi-Selmi et al., 2016). Disasters are classified as natural or human-made (Esposito Amideo et al., 2019). As the number and intensity of disasters increased, the international community was obligated to think about them through a different perspective, which deals with preparedness and recovery beyond response (Aitsi-Selmi et al., 2016).

The disaster risk management cycle is divided into four phases (Altay and Green, 2006): (i) Mitigation, which aims to prevent and or minimize a future disaster threat; (ii) Preparedness, which corresponds to achieving a satisfactory level of readiness to respond to an emergency based on programs, plans and measures; (iii) Response, which focus on providing assistance, improvement, and support to affected population; and (iv) Recovery, which deals with the restoration of life and infrastructure that support affected people.

In this sense, evacuation plays a vital role, and it fits between the preparedness and response phase (Bayram, 2016). An essential aspect of an evacuation scenario is to plan a responsive routing to evacuate all people in an affected area, which involves the available vehicles, pick-up points, and shelter locations. Thereby, those listed points are variables that can be modeled as a Vehicle Routing Problem (VRP).

Vehicle Routing Problem (VRP) is a generalization of the Travel Salesman Problem (TSP) and a well-known NP-hard problem in combinatorial optimization in the operations research field (P. Toth and D. Vigo, 2014; Labadie et al., 2016). Also, the VRP has many variations, representing different scenarios with specific constraints.

An NP-hard problem can not be solved in a possible computational time. As mentioned before, the VPR has several constraints. In an evacuation scenario, uncertainty is another factor. Hence, to solve a VRP, optimal or near-optimal solution can be reached through metaheuristics (Toth et al., 2014; Luke, 2013). Also, a dynamic or stochastic approach can be added to a metaheuristic to deal with uncertainty.

## 1.1 MOTIVATION

A hazard is a danger or risk, and it can be natural or human-made (Van Wassenhove, 2006). When a hazard seriously affects a community or a society's life, causing human, material, and economic or environmental losses that exceed the community's or society's ability to cope using its resources, it turns into a disaster (IFRC, 2019; UNDRR, 2019).

Disasters can inflict significant damage, especially to human life. According to Figure 1.1, natural disasters have grown in the last decade (2008-2018), and their consequences put tremendous stress on a community or society. Hence, this context has attracted the attention of researchers in the disaster management area.

In 2015 year, three main landmark agreements were settled: (a) the Sendai Framework for Disaster Risk Reduction 2015–2030, (b) Sustainable Development Goals, and (c) the Paris Agreement on Climate Change. In this sense, The United Nations International Strategy for Disaster Reduction (UNISDR) Science and Technology Conference (2016) launched the implementation of the Sendai Framework, discussed and endorsed its four priorities: (1) Understanding disaster risk; (2) Strengthening disaster risk governance to manage disaster risk;

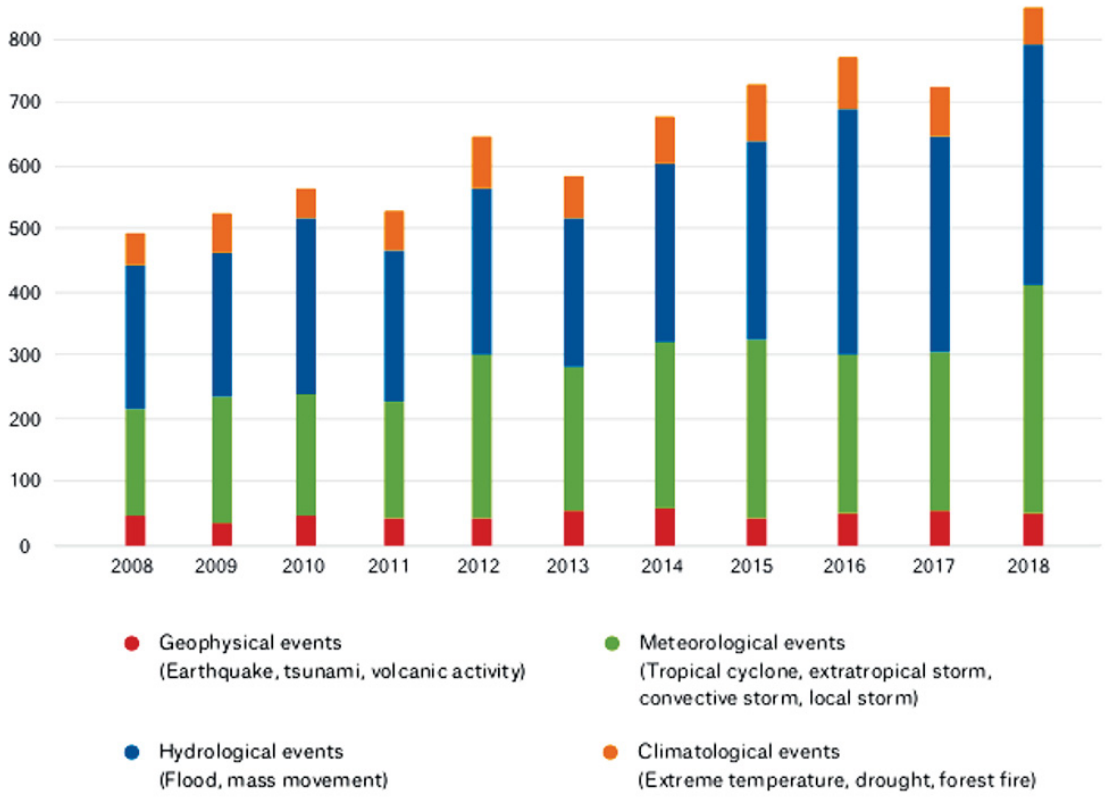



Figure 1.1: Number of relevant natural loss events worldwide 2008 – 2018 (extracted from MunichRe (2019)).

(3) Investing in disaster risk reduction for resilience; (4) Enhancing disaster preparedness for effective response and to "Build Back Better" in recovery, rehabilitation, and reconstruction. Also, ways to monitor progress and review emerging needs, organizing in four workstreams (Aitsi-Selmi et al., 2016):

- **WS1** - Scientific and Technical Partnership to Support the Implementation of the Sendai Framework
- **WS2** - Understanding Disaster Risk, Risk Assessment, and Early Warning
- **WS3** - Use of Science, Technology and Innovation Tools, Methods, and Standards to Support the Implementation and Reporting of the Sendai Framework
- **WS4** - Leveraging Science through Capacity Development and Research

**WS1** is addressed to study how scientific and technical partnerships would leverage national and international networks to promote multidisciplinary research and connect science, policy, and practice. **WS2** already deals with how disaster risk is comprehended, risks are evaluated, and early warning systems are designed. **WS3** addresses the study about data, standards, and new practices needed to evaluate and report risk reduction. Furthermore, **WS4** is responsible for finding gaps in research and which approaches can be used to overcome difficulties in creating and using science for Disaster Risk Reduction (DRR).

Given this, comparing the workstreams with the Disaster Operations Management (DOM) framework (Altay and Green, 2006) phases of Figure 1.2, it noticed that **WS2** and **WS3**

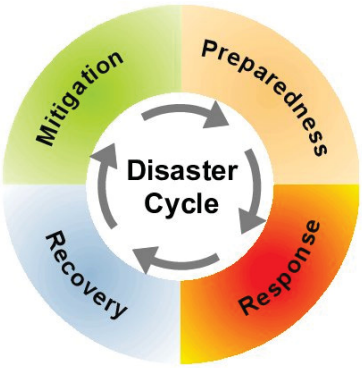



Figure 1.2: Phases of Disaster Operations Management framework (extracted from Flanagan et al. (2011)).

are aligned with the first and second phases of the DOM framework. The DOM framework has four phases (Neal, 1997): (i) Mitigation: related to preventing future disaster threats and or minimizing damages taken by unavoidable threats; (ii) Preparedness: puts effort to achieve a satisfactory level of readiness of an individual and communities to respond a threat through plans and or preparations in advance of it; (iii) Response: include actions are taken to prevent the loss of lives and property damages during a disaster; (iv) Recovery: corresponds to a life support repair, reconstruction and restoration measures to a minimum operating standard.

In this sense, the evacuation plans, which are related to the preparedness and response phases of the DOM (Bayram, 2016; Esposito Amideo et al., 2019), play an essential role in saving lives from a possible disaster (Bayram, 2016). According to the United Office of DRR (UNDRR), formally known as UNISDR, disaster management is the organization, planning, and application of measures for a disaster scenario. In addition, especially the preparedness, response, and recovery (DOM) aspect of decreasing the impact level of a disaster (UNDRR, 2017).

Therefore, to prevent people life's loss, the main strategy used is the evacuation of the disaster area (Bayram, 2016). Evacuation movement is categorized by London Resilience Group (London, 2019) in three types:

1. *self-evacuation*: evacuees, by themselves and without any assistance from a responsible community or agency, move towards safe places;

2. *assisted evacuation*: evacuees use their transportation and move towards shelters, or safe sites, under some assistance from responsible authorities;

3. *supported evacuation*: evacuees with special needs (e.g., disabled, elderly) and or without transportation, requiring support from public authorities to move towards shelters or safe sites.

Also, a combination of any of these three types is called multi-modal evacuation. For example, under a tsunami, an evacuation process may use a combination of land (buses and evacuee vehicles) and air (helicopters) transport.

In that way, one point within the evacuation management context is effective traffic assignment management (FEMA, 2019). Suppose an evacuation traffic management is poorly designed and conducted in a disaster scenario. In that case, e.g., a traffic jam in the transportation network could happen (Dixit and Wolshon, 2014), leaving evacuees in a dangerous situation that may result in further life losses (Yao et al., 2009). However, for designing an effective evacuation plan, an important aspect is evacuation routing planning related to a VRP approach, which is the focus of this work.

The VRP is a generalized form of the Traveling Salesman Problem (TSP) that aims to determine the optimal feasible routes for a fleet of vehicles serving a set of customers. Researchers across various fields have studied the VRP since its introduction by Dantzig and Ramser in 1959, as evidenced by the extensive literature on the subject (Dantzig and Ramser, 1959; Vidal et al., 2013; Esposito Amideo et al., 2019; Bortfeldt and Yi, 2020; Zhang et al., 2020).

In real-life scenarios, routing problems can have numerous constraints and aspects that can significantly increase their complexity, with large numbers of customers often generating considerable instances of the problem. For example, the supply chain (Bortfeldt and Yi, 2020), fuel and battery consumption (Zhang et al., 2020), and emergency and disaster relief (Esposito Amideo et al., 2019) have inspired various extensions to the VRP, including the VRP with Time Window (VRPTW) (Nagata and Bräysy, 2009; Vidal et al., 2013) and the Multi-Depot VRP (MDVRP and MDVRPTW) (Vidal et al., 2013).

Due to the complexity of such scenarios, their variations, and large instances, many approaches utilizing heuristics and metaheuristics have been proposed. The Genetic Algorithm (GA) is a frequently employed approach (Abdallah and Ennigrou, 2020; Zhen et al., 2020; Rabbouch et al., 2019; Li et al., 2014a; Vidal et al., 2013; Nagata and Bräysy, 2009).

Notably, most VRP problems involve capacity and time window constraints, making them particularly challenging. For example, in the Capacitated Vehicle Routing Problem (CVRP), a single or fleet of identical vehicles undertake a tour through all customers, satisfying their demands. The vehicle/fleet departs from the depot and returns to it once the vehicle's capacity is reached or when no more customers are available.

The No Free Lunch theorem (Wolpert and Macready, 1997) states that no algorithm can outperform all other algorithms in solving every possible optimization problem. In essence, this theorem suggests that it is theoretically impossible to create an algorithm that can solve all optimization problems and their features better than any other algorithm (Mallipeddi et al., 2011a).

In this sense, we would like to introduce GREEVO - a Greedy Randomized Adaptive Search Procedure Ensemble Evolutionary Algorithm. GREEVO uses a combination of ensemble techniques, hyper-heuristic (HH), and reinforcement learning (RL). It innovates in the VRP context by utilizing machine learning (RL) to select high-level heuristics. The experiments that were conducted in the CVRP datasets have shown promising results for GREEVO.

## 1.2 OBJECTIVES

This research aims to thoroughly investigate the applicability and advantages of the GREEVO (Greedy Randomized Adaptive Search Procedure Ensemble Evolutionary Algorithm) in the domain of routing problems, with a specific emphasis on its effectiveness in addressing challenges related to the Capacitated Vehicle Routing Problem (CVRP) within the context of evacuation scenarios. The core hypothesis driving this study is that GREEVO can offer a promising solution for complex routing problems, particularly in the CVRP with application in evacuation scenarios, by combining elements of Greedy Randomized Adaptive Search Procedure (GRASP), ensemble techniques, and Hyper-Heuristic (HH). This research seeks to evaluate GREEVO's performance in generating optimal or near-optimal solutions in the classical CVRP and evacuation scenarios considering the capacity constraint. Through comprehensive experiments and a case study, the study aims to contribute valuable insights into the practical applicability of GREEVO in addressing the classical capacitated routing problem challenge associated with evacuation scenarios.

The specific objectives of this research include:

- **GREEVO Algorithm Introduction**: Introduce the GREEVO algorithm, outlining its components and detailing how the integration of GRASP, ensemble techniques, and HH addresses the challenges posed by complex routing scenarios, with a specific focus on the CVRP.
- **Experimental Evaluation**: Conduct a series of experiments to rigorously evaluate the performance of GREEVO in solving the CVRP. Compare GREEVO against existing state-of-the-art algorithms, assessing its efficiency and effectiveness across various datasets.
- **Case Study: New Orleans Evacuation Plan**: Apply GREEVO to a practical case study involving the evacuation plan for hurricanes in New Orleans, USA. Evaluate GREEVO's performance in providing feasible solutions for evacuating residents within specified time windows, considering different fleet speeds and neighborhood sizes.

With this context, these objectives collectively aim to advance the understanding, capabilities, and practical applicability of GREEVO in combinatorial optimization, providing valuable contributions and insights into applying machine learning techniques, especially in the VRP and evacuation scenarios.

## 1.3 HIGHLIGHTS

Based on our conducted experiments, we have ascertained that GREEVO exhibits considerable potential in solving the classical CVRP, as well as presenting an opportunity to leverage machine learning in the VRP. The significant highlights of GREEVO can be enumerated as follows:

- Efficient combination of ensemble and HH;
- Promising adopted strategy in the ensemble;
- Usage of HH to select high-level heuristics in the context of CVRP
- A promising application of machine learning by using the RL with HH in the context of CVRP.

## 1.4 TEXT ORGANIZATION

The remainder of this text is organized as follows:

- **Chapter 2 - VRP**: Enunciates the Vehicle Routing Problem, focusing on the CVRP, to contextualize our work. Furthermore, it describes its basic concepts and a brief description of other variations of the VRP;
- **Chapter 3 - Metaheuristics for VRP**: Presents the theoretical background concerning metaheuristics used in this work. It also describes some notable heuristics and most used operators in the VRP context;
- **Chapter 4 - Literature Review**: Provides a literature review in the VRP, presenting the main works. Additionally, it describes works in the evacuation context;
- **Chapter 5 - Proposal**: Describes the proposed approach, and its components;

- **Chapter 6 - Experiments**: Presents the experiments and results. Moreover, a study case considering the evacuation plan of New Orleans - US. In the end, a brief conclusion;
- **Chapter 7 - Conclusion and Future Work**: Summary the analysis from experiments and their results. Furthermore, it highlights the contributions and limitations of GREEVO and suggests some future works.

## 2 VEHICLE ROUTING PROBLEM

The VRP is a generalization of the TSP (Talebian Sharif and Salari, 2015). Its objective is to find the optimal set of feasible routes to be performed by a fleet of vehicles serving a set of customers. The VRP still is an important and studied combinatorial optimization problem (P. Toth and D. Vigo, 2014). For more than 50 years, since Dantzig and Ramser introduced the problem in 1959, researchers from many fields, especially from operational research area, have studied the VRP (P. Toth and D. Vigo, 2014; Labadie et al., 2016).

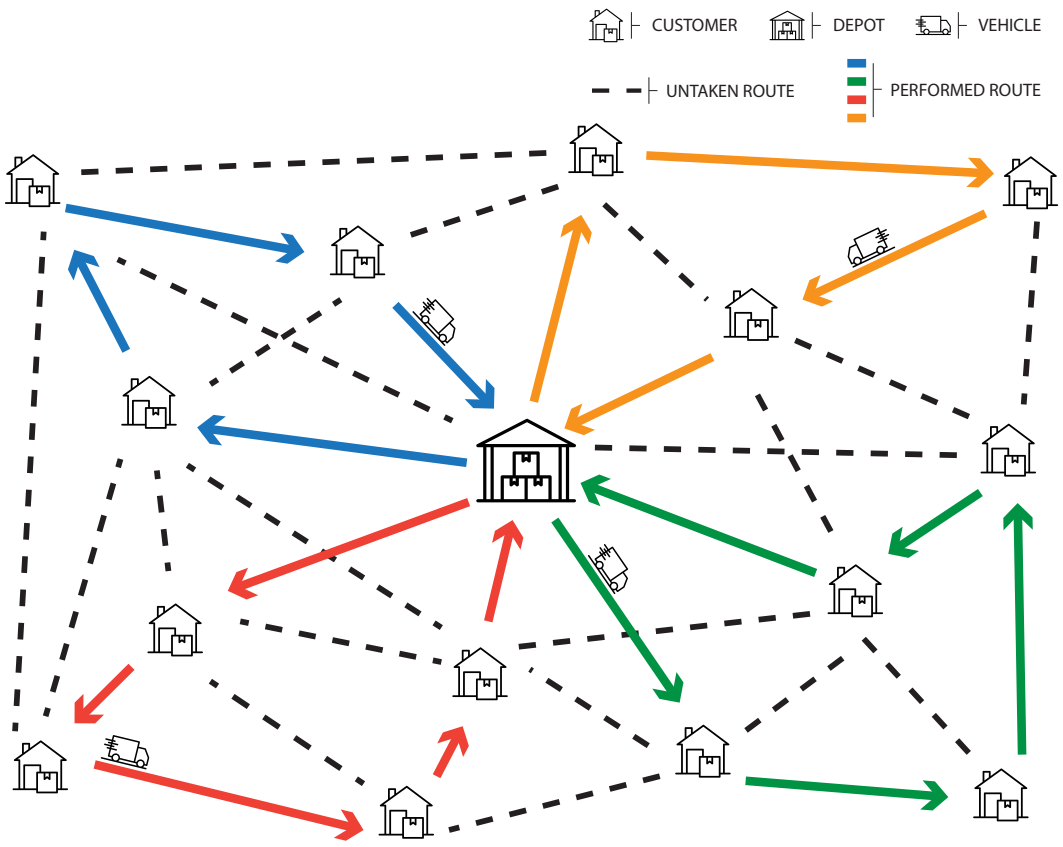



Figure 2.1: Capacited Vehicle Routing Problem Illustration.

The VRP is an NP-hard problem, which means that exact algorithms can not solve it depending on the data size. Pecin et al. (Pecin et al., 2017) have solved to optimally benchmark instances for the Capacitated Vehicle Routing Problem (CVRP) with 199 customers and could solve instances up to 360 customers before solved only by heuristics. Figure 2.1 shows the CVRP, which aims to determine the optimal set of routes to be performed by a fleet of capacitated vehicles to serve the demand of a given customer set (Labadie et al., 2016). Besides, Figure 2.1 defines the CVRP four components (Labadie et al., 2016), which is problem focus of this study:

- Route: usually described through a *graph* (lines);
- Sites to be visited: which can be customers, cities, bus stops, tasks, etc. Generally, is also associated with it a specific request often called demand (customer house);
- Vehicles: which perform the task with a capacity (trucks);

- Depot(s): from where the vehicles start and go (depot in the center).

From a mathematical perspective (Labadie et al., 2016), the CVRP can be modeled as follows: Let $G = (V, E)$ be a complete undirected graph. The $V$ is the node set with the depot ($V_0$) from where a set $K$ of a homogeneous fleet of vehicles with capacity $Q$ can start. Also, customers $n$ have a given demand $q_i, i = 1, 2, ..., n$. Further, each edge $[i, j]$ from $E$ represents the shortest path between nodes $i$ and $j$. Besides, an associated cost $c_{ij}$ is known.

Depending on the authors, the number of available vehicles can be fixed or not. Moreover, a service time $s_i$ can be added for each customer $i$, and each route cost can be delimited by $L$ (corresponding to a limited working time, for instance). In this regard, the main difficulty stands in eliminating sub-tours, i.e., cycles that do not go through the depot.

$$min \sum_{k=1}^{l} \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}^{k} \tag{2.1}$$

$$x_{ij}^{k} = 0, \forall i = j \; and \; \forall k \in K \tag{2.2}$$

$$\sum_{j=1}^{n} x_{V_0 j}^{k} = 1, \forall k \in K \tag{2.3}$$

$$\sum_{i=1}^{n} x_{i V_0}^{k} = 1, \forall k \in K \tag{2.4}$$

$$\sum_{i=1}^{n} x_{ij} = \sum_{h=1}^{n} x_{jh} = 1, \forall j \neq V_0 \tag{2.5}$$

$$Q \geq \sum_{i=1}^{n} w_i^k, \forall k \in K \tag{2.6}$$

$$L \geq \sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij}^{k} (c_{ij} + T), \forall k \in K \tag{2.7}$$

$$x_{ij}^{k} = \begin{cases} 1 & if \; arc \; (i,j) \; is \; taken \; by \; vehicle \; k \\ 0 & otherwise \end{cases} \tag{2.8}$$

Our model presents in (2.1) the objective function. Equation (2.2) ensures that the incoming and outgoing edges will be different nodes. (2.3) and (2.4) guarantee that all vehicles start and finish their tour at depot $V_0$. Equation (2.5) guarantees that all incoming and outgoing edges, regardless of the vehicle, are equal. Inequations (2.6) and (2.7) ensure that the route performed by vehicle k does not exceed its capacity and the maximum travel distance respectively. Finally, equation (2.8) presents $x_{ij}^{k}$ as a binary variable.

Through this simple model, is noticed that an arc represents a link or a sub-route between two vertices or cities/customers/points. Those vertices could also be a depot/warehouse/shelter/hospital from where the vehicles leave and arrive. In that case, when a vertice is a depot, a capacity may also be associated.

Still, according to our model, a sub-route could represent a physical (like a road) or non-physical (like a movement) path from one point to another, it can be performed in one direction or undirected. In all applications, a cost is associated with a link, usually resented

by distance and/or time. Another mentioned point is the demand, for each vertice could be associated with a demand, which can be the number and/or types of products/supplies that should be delivered/picked up, the number of people at a bus stop, among others.

Additionally, we may have some restrictions or regulations that configure the problem. In our example, each vehicle has a capacity, and once this capacity is reached, the vehicle must go to the nearest depot. This is called a constraint, there examples are limited vehicles in a fleet, time window to deliver/pick up, etc. A constraint can be *soft* or *hard*. A soft constraint means that if a possible solution violates it, a penalty will be applied to this solution. On the other hand, a hard constraint does not allow a solution that violates it to be chosen.

## 2.1 VEHICLE ROUTING PROBLEM WITH TIME WINDOW

The Vehicle Routing Problem with Time Window (VRPTW) is an extension of the CVRP, in which the vehicle can only initiate its service once the customer time window is available, i.e., in the interval of time of each customer. If a vehicle arrives earlier than the time window of a customer, it will have to wait until it can start the service in that customer. In terms of costs, a waiting time will increase the total cost, which is the sum of the distances between customers, the service time, and the waiting time. However, the vehicle can not serve a customer after the time window, which is a restriction violation.

As mentioned before, a restriction violation can be *soft* or *hard*. In the case it would be *soft*, then a penalty will be applied to those solutions in which the vehicle arrives after the time window of a customer. Otherwise, infeasible solutions could be repaired.

## 2.2 HETEROGENEOUS OR MIX FLEET VEHICLE ROUTING PROBLEM

Since the early VRP literature, the Heterogeneous or Mixed fleet Vehicle Routing Problem (HVRP) has been known, and the first to study it was Golden et al. (1984). Several variants have been developed considering two points of this problem. On the one hand, the best set of vehicles to be used is called *Fleet Size and Mixed* problem (the fleet size is assumed to be unlimited). On the other hand, in the *Heterogeneous* VRP, the goal is to select the most appropriate vehicles in a limited fleet.

## 2.3 MULTI-DEPOT VEHICLE ROUTING PROBLEM

Unlike the traditional CVRP, in which vehicles depart and arrive at the same single depot, in the Multi-Depot Vehicle Routing Problem (MDVRP), there are two or more depots where vehicles start and end their routes. Even though the literature on this problem is not as vast as in other problems, the MDVRP has many applications (Vidal et al., 2013). As demonstrated in Chapter 5, there are several shelters where refugees are taken by the rescue fleet or on their own vehicles in an evacuation scenario.

## 2.4 SPLIT DELIVERY VEHICLE ROUTING PROBLEM

Dror and Trudeau (1989) introduced the SDVRP. The reason behind the interest in the Split Delivery Vehicle Routing Problem (SDVRP) and its variants is the saving in costs and the number of routes when compared to other problems without this condition (Toth et al., 2014).

In contrast to the classical VRP, in the SDVRP, a customer may be visited more than once by two or more vehicles. Besides, it may be possible that customers with a demand larger

than the capacity of the vehicle require split deliveries (Bortfeldt and Yi, 2020). Although this particularity configures a relaxation of CVRP (Toth et al., 2014; Bortfeldt and Yi, 2020), it is still an NP-hard problem.

## 2.5 STOCHASTIC VEHICLE ROUTING PROBLEM

Most of the studies assume that all the information is previously available, i.e., a deterministic perspective. However, practical applications and real-world environments usually have an uncertainty aspect that affects the parameters of the problem. The uncertainty could come from several different sources, both from variations on known parameters and unknown or unexpected events (Toth et al., 2014).

In this sense, if customers' demands are uncertain, only known when the vehicle visits them, a planned route could be infeasible if the sum of all demands of the customers that belong to that route exceeds the vehicle's capacity. When such a scenario happens, additional decisions should be taken, and the costs usually get increased to produce a feasible solution. Hence, models that explicitly all uncertain features are needed.

Therefore, the stochastic optimization models first define the decision variables based on how and when the values of stochastic parameters throughout the information process are observed. Once these variables are defined, there are two ways to model them: *recourse function* and *probabilistic constraints*. Thereby, the *recourse function* approach is defined as the average cost for a given *priori* solution, and it is considered in the objective function. Still, some particular variables and events in the *probabilistic constraints* approach are described by a probabilistic distribution (Toth et al., 2014).

According to Birge and Louveaux (2011), three parameters are commonly considered to be stochastic:

- **Stochastic demands**: the volume or number of products (or rescuees, supplies) to be delivered or collected are random;
- **Stochastic customers**: the number of customers is associated with a given probability;
- **Stochastic times**: service times and travel times are considered stochastic.

## 2.6 DYNAMIC VEHICLE ROUTING PROBLEM

As in the stochastic approach, uncertain information is also considered, but in contrast, the way Dynamic VRP deals with it is different. Psaraftis (1988) pointed out that there are two aspects in input data, namely *evolution* and *quality of information* (Toth et al., 2014; Psaraftis, 1988). The first one is related to changes even after the route has been performed. The second reflects the uncertainties in the available data. Although those aspects are different, they share some commonalities, as the uncertainty in some input parameters revealed or updated with the routing process.

While information evolves, decisions must be made continuously. Thus, the main goal is to adapt to new information or events and anticipate future events as long as possible. Sometimes, taking advantage of available stochastic information or inferring from past data.

In short, Dynamic VRP deals with decisions that can only be made during the execution of the route plan, handling uncertainty in real time. However, it requires technological support to enable real-time communication between the dispatcher and the vehicle (Toth et al., 2014).

## 2.7 CONCLUDING REMARKS

This Chapter introduced the VRP concept and model focus of this study: the CVRP. Moreover, it presented the essential components of a CVRP and its mathematical model. Lastly, presented other VRP variations, such as VRPTW, HVRP, MDVRP, SDVRP, Stochastic, and Dynamic VRP.

Still, the definitions and concepts shown in this Chapter make clear that real-world scenarios,e.g., evacuation or delivery of goods, can be modeled by a combination of some VRPs. Additionally, this combination can generate new VRPs.

# 3 METAHEURISTICS FOR VRP

In this Chapter, we present some metaheuristics, crossovers, and operators used in the context of Vehicle Routing Problem (VRP) and or by some of the selected techniques that compound our ensemble. Additionally, we detail these techniques, pointing out their benefits.

## 3.1 GENETIC ALGORITHM

Recently, many solutions using Integer Programming (IP) or Mixed IP have been proposed to solve VRP problems (Amiri and Salari, 2019; Li et al., 2019). Despite this, Pecin et al. (2017) optimally benchmarked instances for the Capacitated Vehicle Routing Problem (CVRP) up to 360 customers. However, real-world problems often involve large instances that require quick solutions.

In this sense, metaheuristics can provide an optimal or near-optimal solution in a reasonable computation time (Toth et al., 2014). Metaheuristics are algorithms and techniques that apply some level of randomness to find optimal or near-optimal solutions to hard problems (Luke, 2013).

The term Genetic Algorithm (GA) was first used by Holland (1992), and it is inspired by natural processes, e.g., genetic evolution process (Luke, 2013). Mostly, it starts with an initial population of *solutions*, then some evolutionary process is applied, e.g., crossover and mutation, generating new solutions *offspring*. Usually, this process is done for some generations or another stop criteria. Also, according to Hoos and Stützle (2007), population-based algorithms fit into the formal definition above, considering search positions as a set of individual candidate solutions.

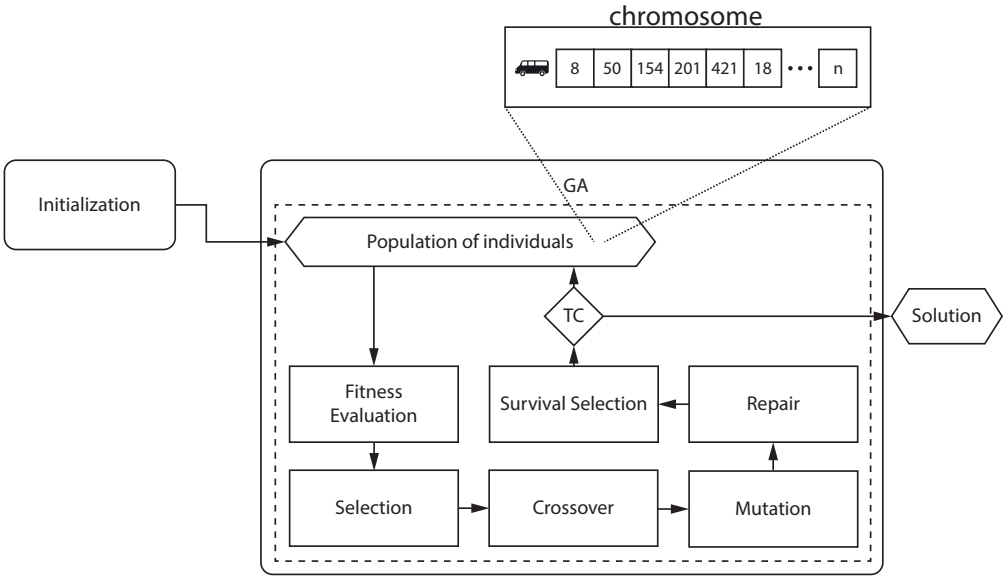



Figure 3.1: A general scheme of a Genetic Algorithm in VRP context.

According to Figure 3.1, a population of individuals (strings or vectors) is used, and these individuals are often referred to as *chromosomes*. Besides, the recombination of individuals is performed using analogies of genetic crossover and mutation. In addition, the search is guided by the results of evaluating the objective function (Fitness Evaluation) for each individual in the population. Based on this evaluation, individuals with better fitness (i.e., higher or lower, which

depends on the type of problem: maximization or minimization) can be selected and receive more opportunities to breed (Labadie et al., 2016). At last, once the Terminal Condition (TC) is achieved, the process stops.

Hereupon, with this explanation, some concepts can be extracted as follows:

- **Individual**: Represents a possible solution to the problem. As mentioned before, an individual is frequently referred to as a *chromosome*. Each chromosome is divided in *alleles*, and each *allele*, usually in the VRP context, corresponds to a customer and or a depot. The definition of an individual, namely *representation*, varies from one problem to another. In this work, a permutation of customers (without delimiters), called giant TSP tour (Prins, 2004; Vidal et al., 2012), is the representation of an individual.
- **Initialization**: A process that generates an individual. It can create individuals randomly or use any problem-specific knowledge. For instance, this work uses a Greedy Randomized Adaptive Search Procedure (GRASP) algorithm to generate good initial solutions.
- **Selection**: When a population of individuals is evaluated, the next step is the selection of parents. The selection is performed to pick up individuals that have good quality or contribute to generating better individuals. The most popular technique is the *Tournament Selection* (Luke, 2013), which is used in this work.
- **Crossover**: In this step, two individuals have their *alleles* mixed and matched to form children. How the information of the parents is combined is crucial for the evolutionary process to achieve better solutions (Luke, 2013).
- **Mutation**: In this procedure, small changes are performed in some individuals (Gendreau and Potvin, 2010). Also, it is a common situation in VRP, applying a local search procedure instead of using standard mutation operators (Prins, 2004; Vidal et al., 2012; Nagata and Bräysy, 2009; Li et al., 2014a; Gutierrez et al., 2018).
- **Survival Selection**: This mechanism is responsible for the management of the population, defining the individuals that will be part of the next generation (Luke, 2013).

Within this context, there is a relevant concept, which is the *diversification*. Diversity is one of the keys to achieving a good performance in GAs, which is related to how different the individuals in the population are. Therefore, depending on how the selection is performed, it can reduce diversity very quickly (Gendreau and Potvin, 2010). In this work, it is considered as diversity management two mechanisms: (a) clone deletion (individuals with the same fitness), and (b) the average *Hamming* distance of the population.

Moreover, another important aspect is the *repair* mechanism, widespread on GAs (Prins, 2004; Nagata and Bräysy, 2009; Vidal et al., 2012). According to Gendreau and Potvin (2010), in some cases, solutions generated by GRASP or GA may not be feasible. Therefore, to achieve feasibility, a *repair* procedure is necessary. In our work, the *repair* mechanism is based on the best insertion approach.

### 3.1.1 Crossovers

The crossover operator plays a crucial role in the GA. Following this principle, as TSP and VRP are related problems, many efficient TSP crossovers can be used (Goldberg et al., 1985; Davis, 1985; Oliver et al., 1987; Syswerda, 1991). Here, we present seven crossovers that are considered in our proposed approach.

#### *3.1.1.1 Partially Mapped Crossover*

Presented by Goldberg et al. (1985), the Partially Mapped Crossover intends to pass some ordering and value information to the offspring. First, a random sub-tour is copied from parent 1 to the offspring. This sub-tour is used to map the alleles of the sub-tour with the corresponding index in parent 2. For example, the sub-tour is composed of alleles 4, 5, and 6. The corresponding alleles in the same index in parent 2 are 1, 6, and 8. The mapping would be 4-1, 5-6, and 6-8. After that, it is copied the alleles of parent 2 into offspring in the same index, and if the customer is already present in the offspring, it is replaced by one in the mapping.

#### *3.1.1.2 Edge Recombination*

The Edge Recombination technique was first introduced by Whitley in 1989. It works by creating a list of neighbors for each allele in both parents. The neighbor list is generated by recording every allele's immediate neighbors, including those that roll around the end of the chromosome, for each parent. These two lists are then combined using a union process that ignores duplicates.

The next step involves randomly selecting an allele $c$ from either parent and adding it to the offspring. This allele is then removed from the list of neighbors. If $c$ has no more neighbors, a new random allele that is not present in the offspring is selected. However, if $c$ still has neighbors, the one with the fewest neighbors is chosen (or a random one if there are multiple with the same number of neighbors).

#### *3.1.1.3 Order Crossover*

Davis (1985) introduced the Order Crossover, which prioritizes the order of alleles. It builds an offspring by choosing a sub-tour in parent 1 and preserving the relative order of alleles of parent 2 by copying the alleles that are not in the selected sub-tour, starting at the last cut point position using the order of alleles of parent 2.

#### *3.1.1.4 Order Based Crossover*

The Order Based crossover was proposed by Syswerda (1991), and it selects several random indexes slightly different from the order crossover of Davis (1985). It searches in parent 2 for the alleles in these indexes, then searches in parent 1 for those alleles and replaces them by ordering appearance in parent 2.

#### *3.1.1.5 Cycle Crossover*

In Cycle Crossover (Oliver et al., 1987) each allele comes from one parent along with its index. For example, start with the allele in the first index of parent 1, go to parent 2 in the same index, and check the allele on it. Search in parent 1 for the allele found in parent 2 and append it to offspring.

#### *3.1.1.6 Edge Assembly Crossover*

Edge Assembly Crossover (EAX) was originally proposed for the traveling salesman problem by Nagata and Kobayashi (1997). It has shown to be among the most efficient crossover, later on, extended to CVRP by Nagata and Bräysy (2009). The EAX has two phases: the first one involves the generation of an incomplete child via the so-called $E$-sets (sub-tours composed of alternating edges from each parent); subsequently, these sub-tours are merged into a single

feasible sub-tours using a greedy repair algorithm. It is worth mentioning, that EAX can link edges that were not inherited. However, the offspring can be very similar to the parents, which leads to a loss of diversity.

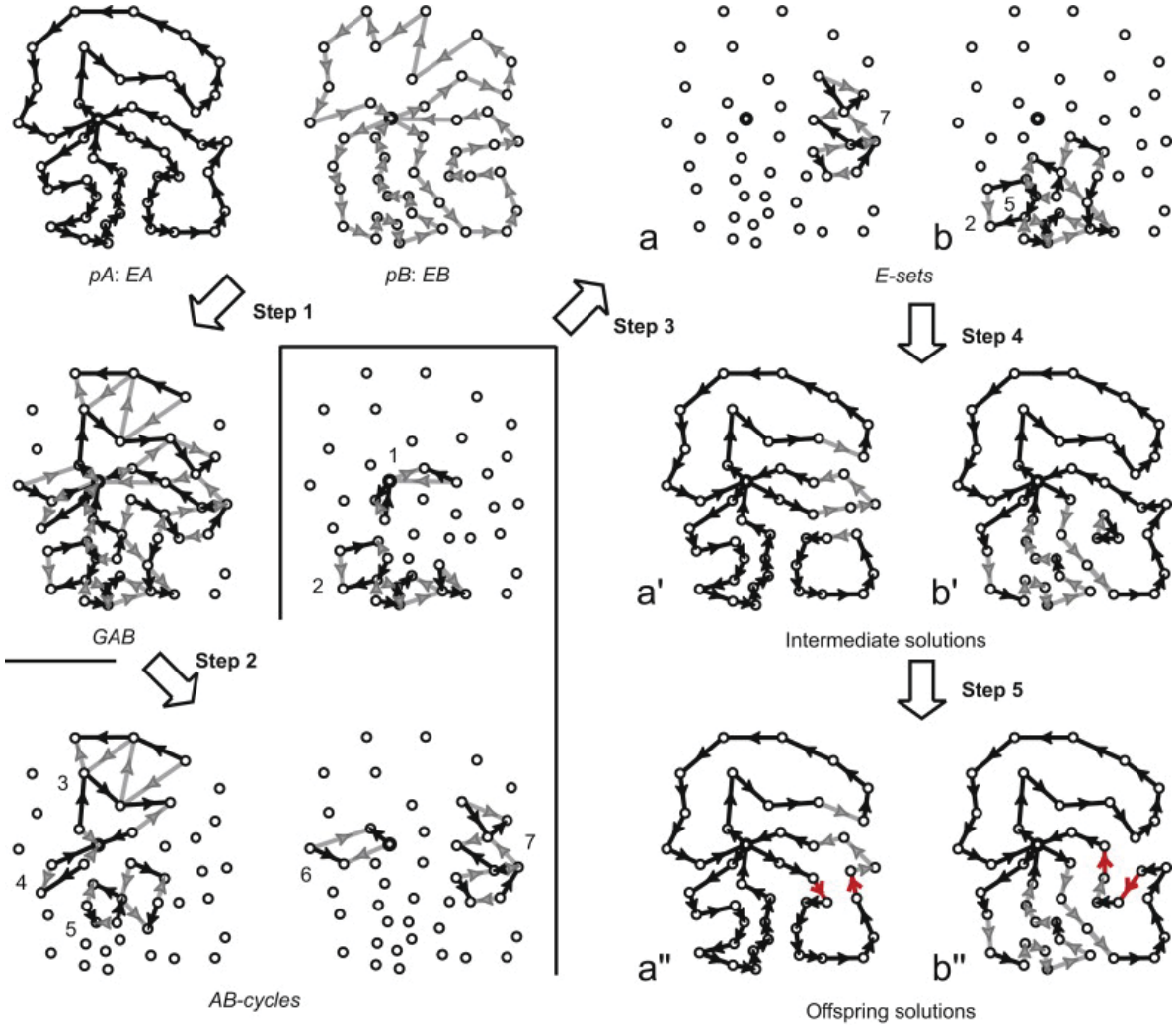



Figure 3.2: The EAX steps illustration (extracted from Nagata et al. (2010)).

Figure 3.2 illustrates the steps performed by the EAX. Thereby, in the first step a graph $GAB$ is defined, $GAB = (V, E_A \cup E_B | E_A \cap E_B)$ in which $EA$ and $EB$ are the edges included in the parents $pA$ and $pB$, respectively. In other words, $GAB$ comprises the edges that are different in the selected parents.

In the sequel, $AB$-cycles are defined on $GAB$ in the second step. Hereupon, an $AB$-cycle is as a cycle on $GAB$ where edges from $pA$ and $pB$ are linked alternately. So, $AB$-cycles are formed by randomly selecting a starting point on $GAB$. Then, edges belonging to $pA$ and $pB$ are selected in their turn until an $AB$-cycle is completed. Also, each edge can belong to only one $AB$-cycle. Thus, edges included in an $AB$-cycle are deleted from $GAB$ every time an $AB$-cycle is completed. Therefore, $AB$-cycles are generated until all edges in $GAB$ have been removed. However, if several alternative edges are connected to a given node in $GAB$, the next edge is selected randomly among the candidates.

In the third step, a *combination* of $AB$-cycles is defined as $E$-sets. Here, two strategies are applied to form E-sets (Nagata and Bräysy, 2009). After that, in the fourth step, $pA$ is selected as the base solution. Intermediate solutions ($a'$ and $b'$) are created by removing from

$pA$ and adding from $pB$ the edges in the selected $E$-set. As Figure 3.2 shows, based on the $GAB$ definition, common edges in the parents are necessarily included in the intermediate solutions ($a'$ and $b'$). The intermediate solutions are constrained by requiring that the number of edges connected to the depot is $2m$ and the number of edges connected to all other nodes is two, where $m$ is the number of routes in the parents. As a result, intermediate solutions consist of $m$ routes originating from the depot and possibly one or more sub-tours (cycles not including the depot).

Finally, the sub-tours are merged in random order with the routes using a greedy approach or some local move in the fifth step. Accurately, for a randomly selected sub-tour, an edge to be removed is selected from both the sub-tour and one of the routes, and then two new edges are introduced to connect them. Here, all possible combinations are attempted, and the move that minimizes the distance is executed. An offspring solution is obtained by repeating this step.

### 3.1.2 Local Move Operators

A local search process starts from an initial solution (e.g., usually provided by some constructive heuristic) and moves from the current solution to a solution in its neighborhood. Frequently, each solution has a few feasible neighbor solutions, and a local move operator determines those moves (Caric and Gold, 2008).

Also, in the literature, we found two groups of operators: *Intra-Route* and *Inter-Route* operators. The first operator group moves one or more customers from one position to another in the same route. The main goal of these operators is the reduction of the overall distance. The *Inter-Route* operators group moves customers between two routes or more routes, and they are used to reduce overall distance. Besides, in some cases, they can reduce the number of vehicles as well (Toth et al., 2014). Additionally, they can also be viewed as destroying and repair operators, which alternate between moves that destroy part of the solution and reconstruct it (Toth et al., 2014).

Therefore, we present the most used local moves, which are considered and will be evaluated to be part of the local search procedure in the GRASP.

#### *3.1.2.1 Destroy and Repair Operators*

Usually, these operators are applied in pairs. First, it is selected a destroy operator, then a repair one. Thus, three destroy and two repair operators are presented as follows:

- **Related Destroy**: remove $n$ customers from the solution based on some similarity measure. This approach makes the repair process easier and more likely to succeed. Here, we present a Related Destroy that uses a similarity metric based on Shaw similarity (Pisinger and Ropke, 2007), which was designed for the Pickup and Delivery Vehicle Routing Problem (PDVRP). Therefore, we had to modify it to the CVRP as follows:

$$shaw = distance(customer_a, customer_b) + |q_{customer_a} - q_{customer_b}| \tag{3.1}$$

  Thereby, randomly $n$ customers are selected (*shaw candidates*), next is picked up one customer from the *shaw candidates* list and calculated the $shaw$ for each customer that does not belong to the *shaw candidates* and inserted into the *shaw sample*. Next, the *shaw sample* is sorted by the $shaw$ metric, and a customer is picked-up randomly by:

$$max\left(\left(\frac{size\_of(shaw\ sample)}{2} - 1\right) \times random\_n, 0\right) \tag{3.2}$$

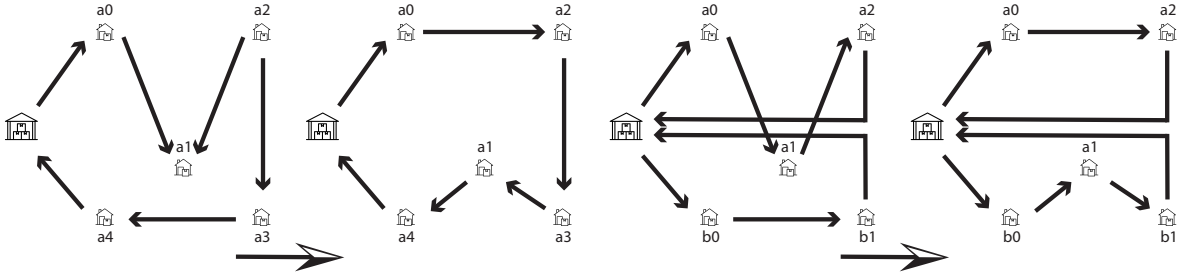
The $random\_number$ ($random\_n$) is a float number between 0 and 1.

- **Worst Destroy**: remove $n$ customers with the highest cost (distance) from the solution. For each pair $(i, i+1)$ of customers, the cost is calculated, and the $n$ highest is removed from the solution.
- **Random Destroy**: remove $n$ randomly selected customers from the solution.
- **Greedy Repair**: inserts a customer in the least cost position in the solution (best insertion).
- **Noise Repair**: inserts a customer in the least cost position in the solution given a noise parameter. Therefore, the noise factor is used as a random ratio for the cost in the best insertion method as follows:

$$noise\_factor = cost * (random\_number - 0.5) \tag{3.3}$$

$$cost = cost + noise\_factor \tag{3.4}$$

### *3.1.2.2 Relocate*



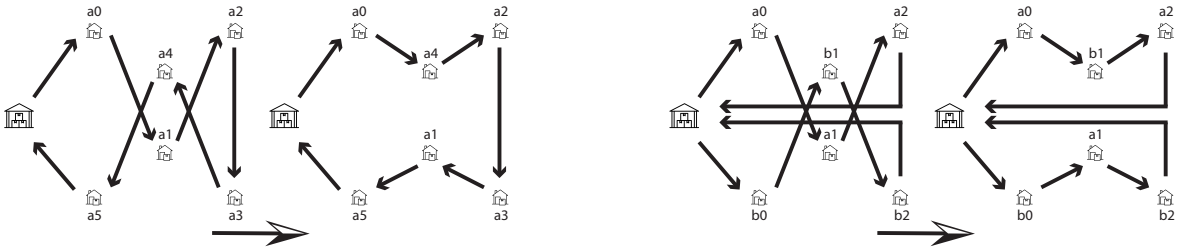

(a) Relocate intra-route (b) Relocate inter-route

Figure 3.3: Relocate Illustrations.

As Figure 3.3(a) shows a relocate intra-route operation, $a1$ is relocated from the original position (between customers $a0$ and $a2$) to between $a4$ and $a3$ the customer. The saving calculation is the maximization of the subtraction $x - y$, where $x$ is the result of three arc deletion ($a0$, $a1$), ($a1$, $a2$), and ($a4$, $a3$). And $y$ is the result of three arc addition of ($a0$, $a2$), ($a3$, $a1$), and ($a1$, $a4$) (Caric and Gold, 2008). For an inter-route operation, Figure 3.3(b), the saving is calculated through the overall maximization of the subtraction mentioned. However, deleting ($a0$, $a1$) and ($a1$, $a2$) arcs, and inserting ($b0$, $a1$) and ($a1$, $b1$) arcs.

### *3.1.2.3 Exchange*

Exchange operation swaps two customers, $a1$ and $a4$ (Figure 3.4(a)), and $a1$ and $b1$ (Figure 3.4(b)). Therefore, in the intra-route movement, the ($a0$, $a1$), ($a1$, $a2$), ($a3$, $a4$), and ($a4$, $a5$) arcs are deleted. After that, arcs ($a0$, $a4$), ($a4$, $a2$), ($a3$, $a1$), and ($a1$, $a5$) are inserted. Nevertheless, in the inter-route operation, customer $b1$ is relocated to route $a$, and customer $a1$ to route $b$.




(a) Exchange intra-route

(b) Exchange inter-route

Figure 3.4: Exchange Illustrations.

### *3.1.2.4 2-Opt*

A 2-opt move consisted of removing two arcs from a route and reconnecting them in the same route (if a cost-saving exists) while changing the orientation of the sub-path that is not connected to the depot shown by Figure 3.5.

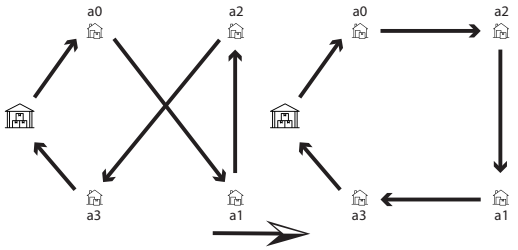



Figure 3.5: 2-Opt Illustration.

### *3.1.2.5 2-Opt Star (2-Opt*)*

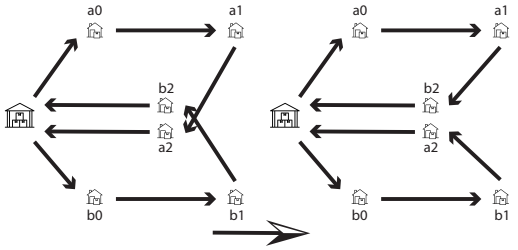



Figure 3.6: 2-Opt Star Illustration.

the 2-opt* move is very similar to the 2-opt, but two routes are modified instead of one. As in the 2-opt, two arcs ($a1$, $a2$) and ($b1$, $b2$) from two distinct routes are deleted and reconnected by inserting the arcs ($a1$, $b2$) and ($b1$, $a2$). According to Figure 3.6, the difference from 2-opt is that the tails of the routes are affected, i.e., they are exchanged.

### *3.1.2.6 OR-Opt*

Whenever savings exist, the intra-route operator Or-Opt relocates the sub-path ($a1$,$a2$) to a different position in the route (inserting the sub-path between $a4$ and $a5$ as shown by Figure 3.7). The Or-opt is a practical operator due to its speed (Caric and Gold, 2008).

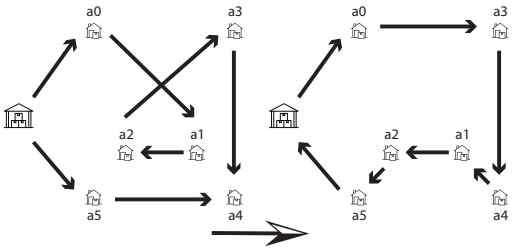



Figure 3.7: OR-Opt Illustration.

### *3.1.2.7 Cross-Exchange*

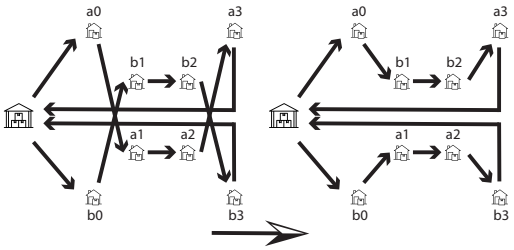



Figure 3.8: Cross-Exchange Illustration.

The swaps of two groups of customers from one route to another are called Cross-Exchange (Figure 3.8). Therefore, this operation is done by deleting four arcs ($a0$, $a1$), ($a2$, $a3$), ($b0$, $b1$), and ($b2$, $b3$), after inserting ($a0$, $b1$), ($b2$, $a3$), ($b0$, $a1$), and ($a2$, $b3$). When only sub-paths that contain a few customers are considered, the size of the cross-exchange neighborhood is reduced (Toth et al., 2014). Figure 3.8 shows an inter-route operator, but the cross-exchange can be used in an intra-route operation as well.

## 3.2 GREEDY RANDOMIZED ADAPTIVE SEARCH PROCEDURE

Feo and Resende (1989) presented the GRASP as an iterative metaheuristic (Feo and Resende, 1995), in which each iteration performs two phases: construction and a local search phase, as shown by Figure 3.9. Additionally, the best overall solution is kept (Memory).

In the construction phase, a feasible solution is iteratively constructed. At each construction iteration, the choice of the next element to be added is determined by ordering all elements in a candidate list concerning a greedy function. The adaptive aspect of the GRASP is that the benefits associated with every element are updated at each iteration of the construction phase, reflecting the changes brought on by selecting the last element.

Moreover, the probabilistic component of the GRASP is characterized by randomly choosing one of the best candidates in the list, but not necessarily the best candidate. Therefore, the list of best candidates is called the Restricted Candidate List (RCL). Despite this choice technique allowing for different solutions to be obtained at each GRASP iteration, the power of the adaptive greedy component of the method is not necessarily compromised (Feo and Resende, 1989).

As in many deterministic methods, the solutions generated by a GRASP construction are not guaranteed to be locally optimal concerning simple neighborhood definitions. Hence, a local search phase starts to improve each constructed solution (Gendreau and Potvin, 2010; Labadie et al., 2016).

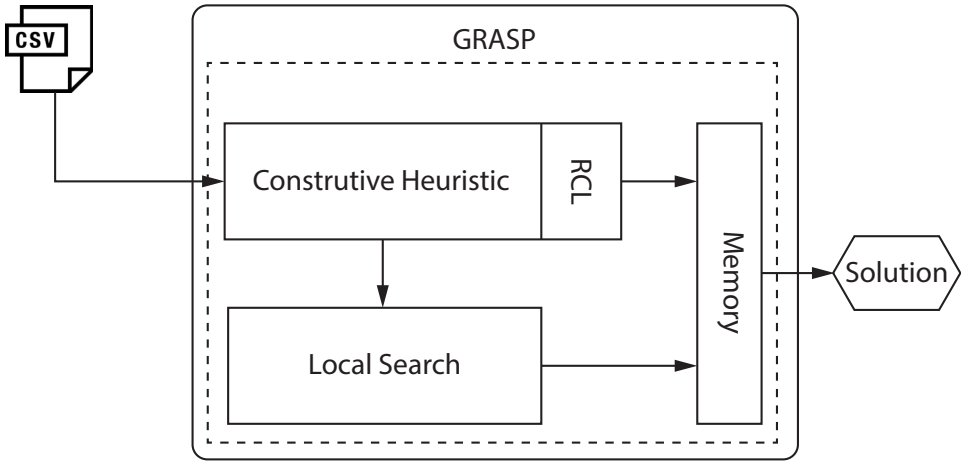



Figure 3.9: A general scheme of the GRASP.

## 3.3 SIMULATED ANNEALING

Simulated annealing (SA), proposed by Kirkpatrick et al. (1983) is a stochastic optimization algorithm that belongs to the family of metaheuristic algorithms. This algorithm is based on the annealing process of metallurgy, where a material is gradually cooled after being heated to achieve a low-energy crystalline state.

Simulated annealing applies the same concept to navigate the solution space of an optimization problem. The algorithm initiates with an initial solution and gradually explores adjacent solutions while steadily reducing the exploration parameter, usually called *temperature*.

Figure 3.10 shows a general scheme of the SA. The process starts with the temperature's initialization and a copy of the initial solution to $S^*$. After, the heuristic is applied until the terminal condition is achieved $T > T_{final}$. The $S^*$ is evaluated by the object function given by the heuristic and through a *acceptance criterion*, which usually uses the temperature $T$ as an adaptive factor. Finally, the temperature is cooled and the solution is returned in case the terminal condition was achieved.

Notably, as the temperature is gradually reduced, the inferior solutions (considering the objective function) to the previous ones are more rejected during the acceptance criterion phase.

## 3.4 SPLIT ALGORITHM

The *Split algorithm*, proposed by Prins (2004), is a route-first cluster-second heuristics and modern genetic algorithms for vehicle routing problems. In this respect, a route-first cluster-second heuristic is a constructive method, in which in the first phase a giant TSP tour is created, disregarding side constraints. Then, it is decomposed into feasible vehicle routes in a second phase. As highlighted by the survey of Prins et al. (2014), around 70 recent articles use this technique.

The objective of *Split algorithm* is to partition the giant TSP tour into $m$ disjoint sequences of consecutive visits. Therefore, each sequence is associated with a route, which originates from the depot. After, visits its respective customers and then returns to the depot. Therefore, the total distance of all routes should be minimized.

The *Split algorithm* aims to find the delimiters for the route as a shortest path problem. Define an auxiliary graph $G = (V, A, c)$, where $V$ is the set of nodes with $n + 1$ elements indexed from 0 to $n$. $A$ an arc $(i, j)$ with $i < j$. Each arc represents a trip $(r_{i+1,j})$ from depot $V_0$, visiting

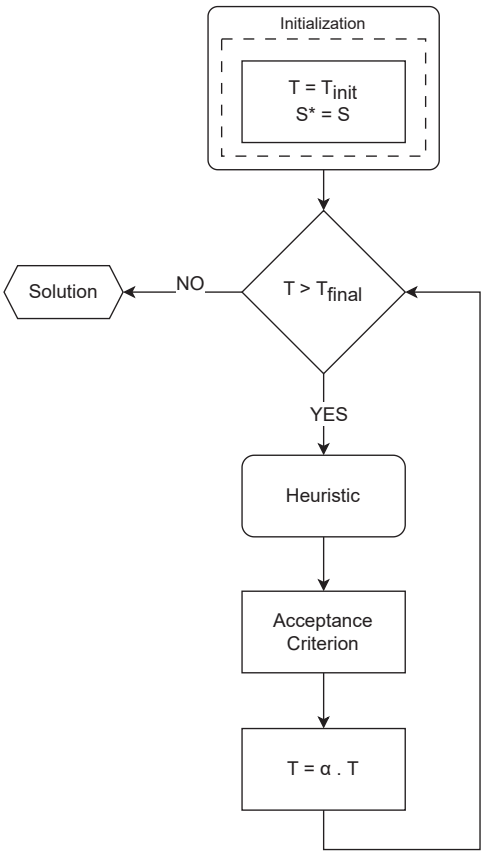



Figure 3.10: A general scheme of the SA.

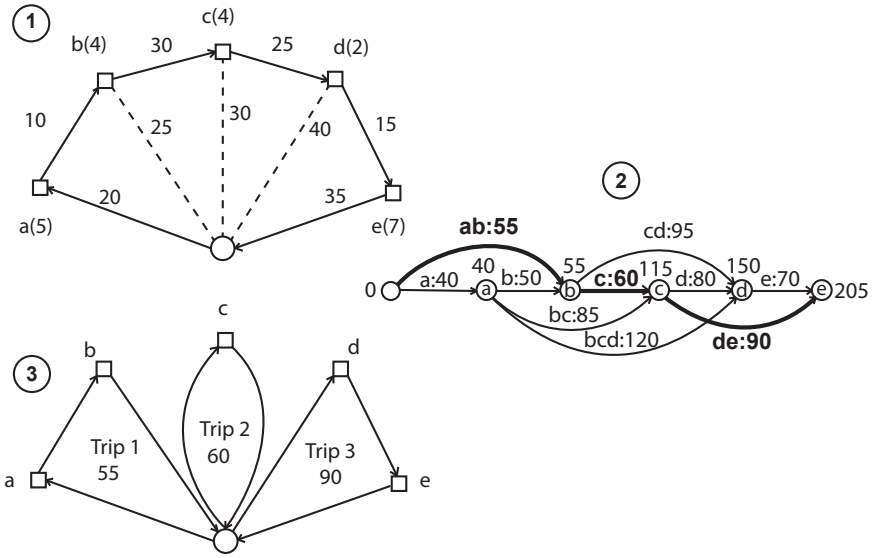



Figure 3.11: Illustration of the *split algorithm* process (extracted from Prins (2004)).

customers $V_{i+1}$ to $V_j$, and returning to the depot $V_0$ with a $d$ cost associated. If a trip is feasible according to load (equation 3.5) and cost (equation 3.6) conditions, the $C_{ij}$ is equal to the trip cost.

$$\forall (i,j) \in A : \sum_{x=i+1}^{j} q_{r_x} \leq W, \tag{3.5}$$

$$\forall (i,j) \in A : C_{i,j} = V_{0,r_{i+1}} + \sum_{x=i+1}^{j} (d_{r_x} + V_{V_x,V_{x+1}}) + d_{r_j} + V_{r_j,0} \leq L. \tag{3.6}$$

Figure 3.11 shows a sequence $V = (a, b, c, d, e)$ with $W = 10$ and $L = \infty$, the demand of each customer is in brackets (1). $G$ (2) contains, e.g., arcs (bold lines) $a - b$ with $C_{a,b} = 55$ for the trip $r_{a,b}$ $(0 - a - b - 0)$, $c$ with $C_c = 60$ for the trip $r_c$ $(0 - c - 0)$, and $d - e$ with $C_{d,e} = 90$ for the trip $r_{d,e}$ $(0 - d - e - 0)$ with a total cost of 205. The lower part (3) gives the VRP solution with three trips.

## 3.5 HYBRID GENETIC SEARCH WITH AN ADAPTIVE DIVERSITY CONTROL (HGSADC)

One of the particular algorithms for solving several VRPs is the Hybrid Genetic Algorithm with Adaptive Diversity Control (HGSADC) proposed by Vidal et al. (2012). The HGSADC is based on the GA introduced by Holland (1975) but includes some advanced features in terms of solution evaluation, offspring generation and improvement, and population management, contributing to its originality and high-performance level.

According to Figure 3.12, the algorithm evolves two populations of individuals (feasible and infeasible solutions). The two populations contain between $\mu$ and $\mu+\lambda$ individuals. Hereafter, the populations are generated by the **Initialization** process, where $4\mu$ individuals are created by a **Random** procedure. Then, the **Education** is performed, executing two local search-based procedures. First, a route improvement procedure (RI) is executed, which explores for each period a neighborhood based on relocations and exchanges of customer visit sequences, with eventual inversions. After, the pattern improvement procedure (PI) evaluates the best combination of re-insertions (Relocate) within periods for each customer in random order. Any improving re-insertion is directly performed until no more improvement can be found. Lastly, the RI is called again, providing efficient service sequencing and assignment characteristics. Finalizing the **Initialization** process, a **Repair** procedure (which also calls the **Education**) is applied with a certain probability to infeasible individuals.

Any individual $I$ in the population (**Infeasible Population** + **Feasible Population**) is characterized by its solution cost (Equation 3.8), where $\omega^Q$ and $\omega^L$ represent the penalties for exceeding the vehicle capacity and the route maximum duration, respectively. Also, $cost(r)$, $l(r)$, and $q(r)$ are the total distance, total duration, and total capacity of route $r$, respectively. Still, the individual diversity is defined as the average distance from $I$ to its closest neighbors $N_close$ related to its population (Equation 3.9). Here, the distance $\delta^H(P, P_2)$ is the Hamming distance. Therefore, the Equation 3.10 presents the evaluation of an individual, where $nbElit$ is the number of elite individuals one desires to survive to the next generation, and $nbIndiv$ stands for the rank of an individual $I$ in its population of size $nbIndiv$.

$$\phi(r) = cost(r) + \omega^L \times max(0, l(r) - L) + \omega^Q \times max(0, q(r) - Q) \tag{3.7}$$

$$fit(I) = \sum_{\forall r in I} \phi(r) \tag{3.8}$$

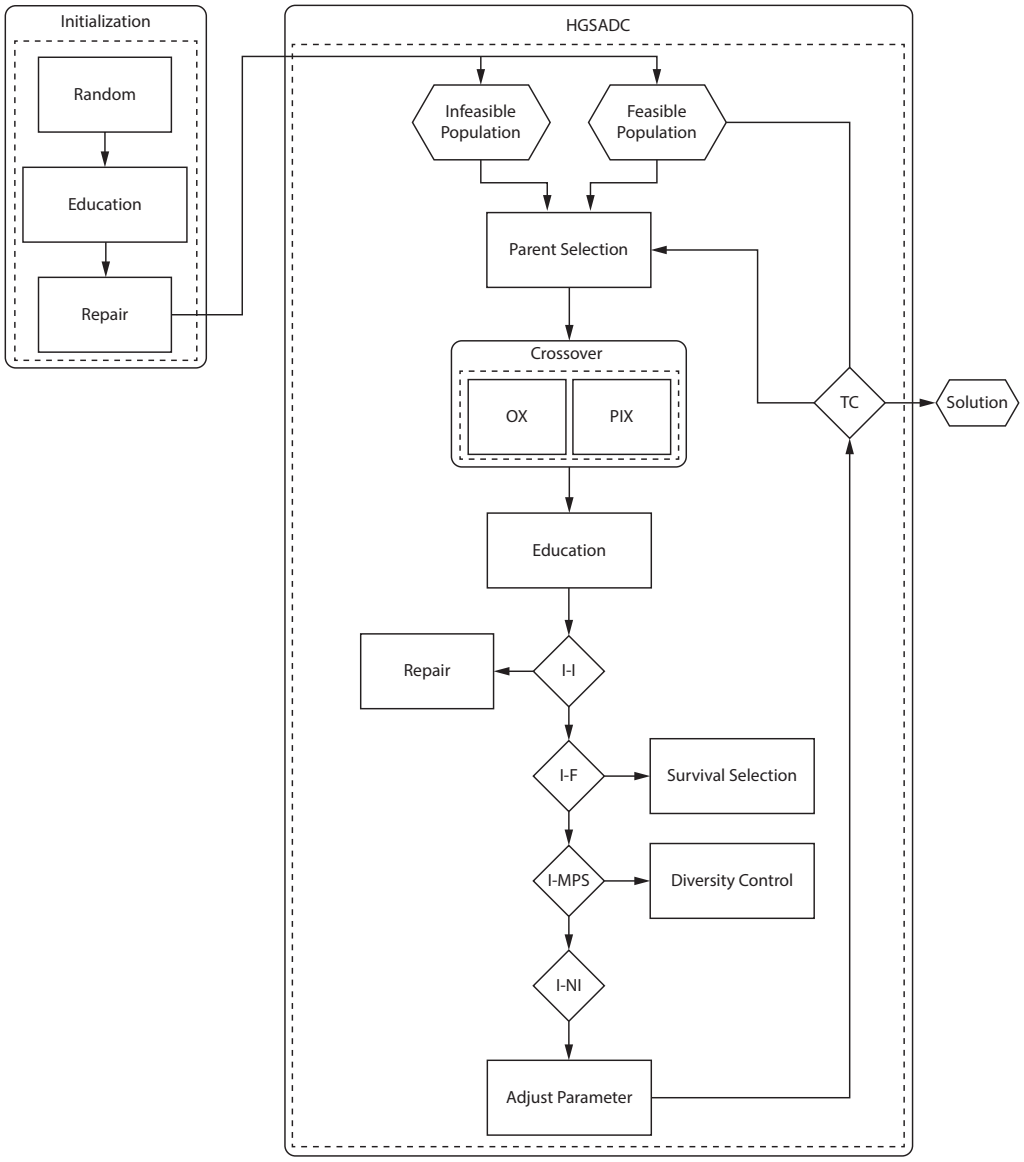



Figure 3.12: A general scheme of the HGSADC.

$$dc(I) = \frac{1}{N_{close}} \times \sum_{I_2 \in N_{close}} \delta^H(I, I_2) \tag{3.9}$$

$$BF(I) = fit(I) + (1 - \frac{nbElit}{nbIndiv}) \times dc(I) \tag{3.10}$$

Continuing with the process, two parents are selected (**Parent Selection**) by a binary tournament in the union of both **Infeasible Population** and **Feasible Population**, then are assigned to a crossover operator. The periodic crossover with insertions (PIX) is used for the Periodic VRPTW (PVRPTW) and enables to inheritance of good sequences of visits from both parents. Also, it recombines visit patterns. In contrast, for the other VRPs, the HGSADC relies on the Ordered Crossover.

Thereafter, the offspring undergoes the *Split algorithm* to extract its routes. Then, **Education** is applied, followed by a **Repair** phase (executed with a certain probability) if the current solution is infeasible (**I-I**). Moreover, the **Repair** procedure increases the penalty values

by a factor of 10, and the process is repeated with a penalty increase of 100 if the offspring remains infeasible.

When the maximum size ($\mu + \lambda$) of a population is reached (**I-MPS**), the **Survival Selection** is applied, removing iteratively $\lambda$ times the worst clone in terms of biased fitness (BF - Equation 3.10), or the worst individual when no clone exists.

Also, a **Diversity Control** is triggered whenever $It_i = 0.4 \times It_{TOTAL}$ (**I-NI**) iterations are performed without improving the best solution. For that reason, it is retained the best $\frac{\mu}{3}$ individuals of each population, and $4\mu$ new individuals are added (introducing new genetic material). After that, the **Survival Selection** is executed again.

Lastly, the **Adjust Parameter** is performed every 100 iterations, where $\omega^L$ and $\omega^Q$ are the parameters to be updated. $\xi^{REF}$ is a target proportion of entirely feasible individuals, and $\xi^{L,Q}$ is the proportion in the last 100 generated feasible individuals concerning route duration and vehicle capacity, respectively. Thus, the parameters are updated as follows:

- if $\xi^{L,Q} \leq \xi^{REF} - 0.05$, then $\omega^{L,Q} = \omega^{L,Q} \times 1.20$
- if $\xi^{L,Q} \leq \xi^{REF} + 0.05$, then $\omega^{L,Q} = \omega^{L,Q} \times 0.85$

### 3.5.1 Benefits

The HGSADC presents as benefits the following aspects:

- **Education**: The utilization of the RI and PI methodology has proven to be effective in optimizing movement by exploring the neighborhood for each period and evaluating the most suitable combination to apply.
- **Feasible and Infeasible population**: The management of these two populations enlarged the search space, which may have helped the HGSADC to escape from local optimums.
- **Diversity Control**: This mechanism works as a "shack procedure", keeping the population at a certain level of diversity by introducing new genetic material. This procedure may have helped the crossover phase combine a wide variety of genetic material, generating better solutions.

## 3.6 FAST ITERATED LOCAL SEARCH LOCALIZED OPTIMIZATION (FILO)

The FILO metaheuristic, according to Figure 3.13, consists of a **Initialization** phase, which builds an initial feasible solution using a restricted version of the savings algorithm. Then, it follows an improvement phase aimed at further enhancing the initial solution quality. More precisely, the improvement phase may first employ a **Route minimization** procedure to possibly reduce the number of routes in the initial solution when it is considered to be using more routes than necessary. After, a core **Optimization** procedure uses an iterative and localized optimization scheme to improve the solution quality further. Both route minimization and core optimization follow the Iterated Local Search paradigm (ILS) in which shakings, performed in a ruin-and-recreate fashion, and local search applications interleave for a prefixed number of iterations.

The initial solution is built using an adaptation of the well-known savings algorithm by (Clarke and Wright, 1964a). It was set $n_{cw} = 100$ and computed the savings values for the arcs connecting each customer $i$ to its $n_{cw}$ neighbor customers $j$ using a lexicographic

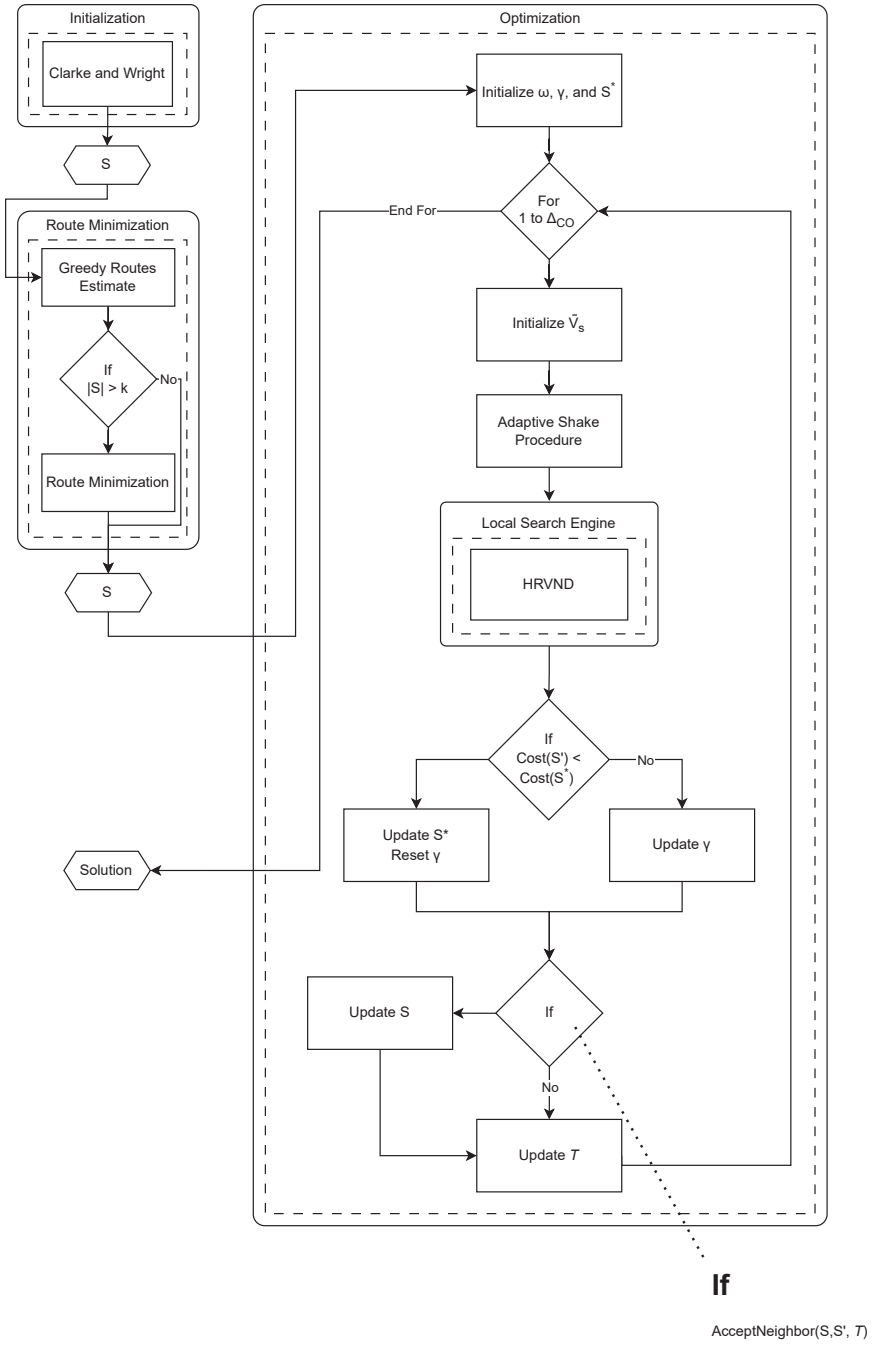



Figure 3.13: A general scheme of the FILO.

order for the customers to avoid symmetries. More precisely, this set is given by the arcs $\{(i,j) : i \in V_c, j \in N_i^{n_{cw}}(\{j \in V_c : i < j\})\}$.

After, the **Route Minimization** starts. This procedure is applied to a solution $S$ built by the initial construction phase whenever the number size of $S$ (number of routes) is found to be greater than an ideal estimated number of routes $m$. The value $m$ is computed by heuristically solving a bin-packing problem with an item of weight $q_i$ for each customer $i \in V_c$ and bins of capacity $Q$ through a simple greedy first-fit algorithm.

When the condition is triggered, the algorithm begins with the initialization of the best-found solution $S^*$, which is set to the initial solution generated during the construction

phase. During each iteration, pairs of routes $r_i, r_j$ are selected from the solution. Customers from these routes are temporarily removed from the solution and placed in a list of unrouted customers $U$. Route selection is guided by specific criteria. The first route $r_i$ is selected using a random customer seed, while the second route $r_j$ is determined based on customer neighbors with increasing cost, defined by the cost function $c_{ij}$.

Unrouted customers in list $U$ undergo either a random shuffling or sorting based on their demand. For each unrouted customer $i \in U$, the algorithm endeavors to find a feasible position in the existing routes that minimizes insertion costs. If a suitable position cannot be located and capacity constraints are breached, a decision is made to create a new route. The choice to create a new single-customer route hinges on the number of existing routes and a probability threshold $P$, calculated by Equation 3.11.

$$P = z \cdot P \quad \text{where} \quad z = \left(\frac{P_f}{P_0}\right)^{\frac{1}{\Delta_{RM}}} \tag{3.11}$$

Where:

- $P$ is the updated probability threshold.
- $P_0$ is the initial probability.
- $P_f$ is the final probability.
- $\Delta_{RM}$ represents a specific number of iterations.

The algorithm further involves a mechanism for updating the threshold $P$ based on an exponential schedule. Subsequently, following customer insertion, a restricted Hierarchical Randomized Variable Neighborhood Descent (HRVND) is applied to the solution. This secondary improvement phase is focused on the first tier of move generators and is executed for a limited number of iterations $\Delta_{RM}$.

The iterative procedure continues until a solution is obtained, where all customers are routed. In this case, the best solution $S^*$ is replaced with the new solution if the latter offers a lower cost or the same cost but fewer routes. The procedure also incorporates an early stopping condition based on the number of routes in the best solution, preventing further iterations when a desired route count is achieved.

Before moving on to the next iteration, if a partial or feasible solution has a cost higher than the current best solution, it is reset to the best solution. The procedure continues for a specified number of iterations $\Delta_{RM}$. Ultimately, the final result is the best solution $S^*$ achieved through the iterative process.

Moving forward to the Optimization module, the Hierarchical Randomized Variable Neighborhood Descent (HRVND) combines the principles of random and fixed neighborhood exploration strategies. It does this by using a slightly more structured approach whereby local search operators are grouped into tiers, each of which is a compound operator that applies its subset of local search operators by following the RVND principles. The overall HRVND links the tiers together once they have been ordered according to the criteria defined by the VND. This includes the overall computational complexity of the operators involved in the tier. The HRVND can thus be seen as a standard VND in which each tier is a compound local search operator and where successively more expensive tiers are used to escape from the local optima of the previous ones.

The HRVND local search applies the following operators in two tiers: (1) 10EX, 11EX, SPLIT, TAILS, TWOPT, 20EX, 21EX, 22EX, 20REX, 21REX, 22REX, 22REX*, 30EX, 31EX,

32EX, 33EX, 30REX, 31REX, 32REX, 33REX, 32REX* and 33REX*, and (2) ejection-chain (EJCH). The first tier contains operators defining neighborhoods of quadratic cardinality, and they all have very similar execution times. On the other hand, the second tier contains the most expensive operator employed by the local search engine.

The operator's neighborhood is thoroughly explored, and all the enhancements are implemented before moving on to the next operator on the list. The next level is only utilized when the prior levels have reached a local optimum solution.

Still, the Optimization module involves a *shaking* step, where a random walk (of length $\omega_i$) is initiated. The walk begins from a randomly selected customer, denoted as $i \in V_c$, within the customer subgraph. The walk unfolds within a subgraph defined by $G(V_c, A_c)$, where $V_c$ represents the set of customers, and $A_c$ comprises the arcs connecting these customers. As the walk progresses, visited customers are removed from the solution, and the subgraph is updated accordingly. This involves modifying $V_c$ and $A_c$ to reflect the removal of customer $i$.

The walk may be extended by moving either forward or backward within the same route $r_i$. Alternatively, it may involve jumping to a neighboring route. The decision regarding whether to continue on the same route or jump to another is made probabilistically, with equal probabilities assigned to each choice. If a jump to a neighboring route is selected, the algorithm assesses customers in the vicinity, focusing on those with increasing $c_{ij}$ cost. The goal is to identify a suitable route, denoted as $r_j$, that meets specific requirements.

Following the *shake* phase, the algorithm proceeds to the *recreate* step. Here, the previously removed customers are reintegrated into the solution, with a focus on minimizing the insertion cost. The algorithm employs various strategies, including random shuffling or sorting of customers based on demand, distance from the depot, or other relevant criteria.

Throughout the algorithm, there is a mechanism for controlling the ruin intensity. This involves adaptively adjusting the length of the random walk, represented as $\omega_i$. The iterative adaptation process aims to identify disruptive actions that align with the specific problem and solution.

An integral part of the algorithm is the application of HRVND to the shaken solution $S$. HRVND is used to identify a local optimum solution, denoted as $S^r$.

The algorithm employs a Simulated Annealing (SA) acceptance strategy to determine whether the new solution $S^r$ should be accepted as the next point in the search trajectory, which is given by Equation 3.12 and temperature cooling by Equation 3.13.

$$c(S^r) < c(S) + T \cdot \ln \mathcal{U}(0,1) \tag{3.12}$$

$$T = c \cdot T, \quad with \quad c = \frac{T_f}{T_0}^{\frac{1}{\Delta_{CO}}} \tag{3.13}$$

This decision is influenced by factors such as cost differentials and a temperature parameter, denoted as $T$. The temperature parameter is initially set as $T_0$ and is adjusted at the end of each core optimization iteration.

The optimization process takes advantage of dynamic move generators on a vertex-wise basis. Sparsification parameters, denoted as $\gamma_i$, are utilized and adaptively adjusted based on the number of non-improving iterations involving a particular vertex $i$, given by the Equation 3.14. The value of $\gamma_i$ is reset to a base value, denoted as $\gamma_{\text{base}}$ when a solution improving $S^*$ is discovered during a local search involving vertex $i$.

$$\frac{\delta \cdot \Delta_{CO} \cdot \text{average}(|\bar{V}_S|)}{|V|} \tag{3.14}$$

Where:

- $\delta$ is a reduction factor, and it belongs to the closed interval $[0, 1]$.
- $\Delta_{CO}$ represents some change or cost related to the optimization problem being solved.
- average$(|\bar{V}_S|)$ denotes the average number of vertices cached after previous local search executions.
- $|V|$ is the total number of instance vertices.

The algorithm also incorporates cache size management, where a cache is maintained to track vertices. This cache is updated based on conditions such as the number of non-improving iterations or the discovery of a better solution. In cases where the cache size is smaller than the total number of vertices, certain vertices may not be considered for updates. This is particularly true when the cache size, denoted as $C$, is less than the total number of instance vertices, denoted as $|V|$.

### 3.6.1 Benefits

Despite its complexity, FILO has some notable aspects:

- **Adaptive Shake Procedure**: This process involves adjusting the length of a random walk iteratively to identify disruptive actions that align with the specific problem and solution. This may help to destroy the parts of the solution that are more likely to be improved.
- **Sparsification combined with Local Search Engine**: The application of the sparsification factor in conjunction with the Local Search Engine warrants a more structured approach that effectively blends the randomness of exploration strategies with fixed neighborhood exploration strategies. This is achieved through the hierarchical grouping of local search operators into tiers, each of which represents a compound operator that applies a distinct subset of local search operators. In other words, this mechanism facilitates the control of the application of the subset of local operators in the destroyed solution.

## 3.7 SLACK INDUCTION BY STRING REMOVALS (SISR)

Despite being based on basic operators such as ruin and recreate, the SISR approach showcases a performance that is comparable to FILO and HGSADC (Christiaens and Berghe, 2020).

Figure 3.14 shows the solution generation on the SISR algorithm. The initialization of a solution, which is denoted as $S = \{\mathfrak{I}, U\}$, consisted of allocating one customer per tour. In this context, $\mathfrak{I}$ denotes the collection of tours, each of which fulfills at least one customer. On the other hand, $U$ represents the group of customers who are not serviced by any tour $\tau \in \mathfrak{I}$. Following the algorithm process, the temperature $T$ is set to the initial temperature $T_0$, then the solution $S$ is saved as the best solution $S^*$ and checks if the fleet should be minimized or not.

The SISR algorithm utilizes simulated annealing to search through the neighborhood. The algorithm starts with an initial temperature of $T_0$, which is gradually reduced to its final temperature $T_f$ in $f$ iterations using an exponential cooling schedule (Equation 3.15). The cooling constant $C$ is determined using Equation 3.16.

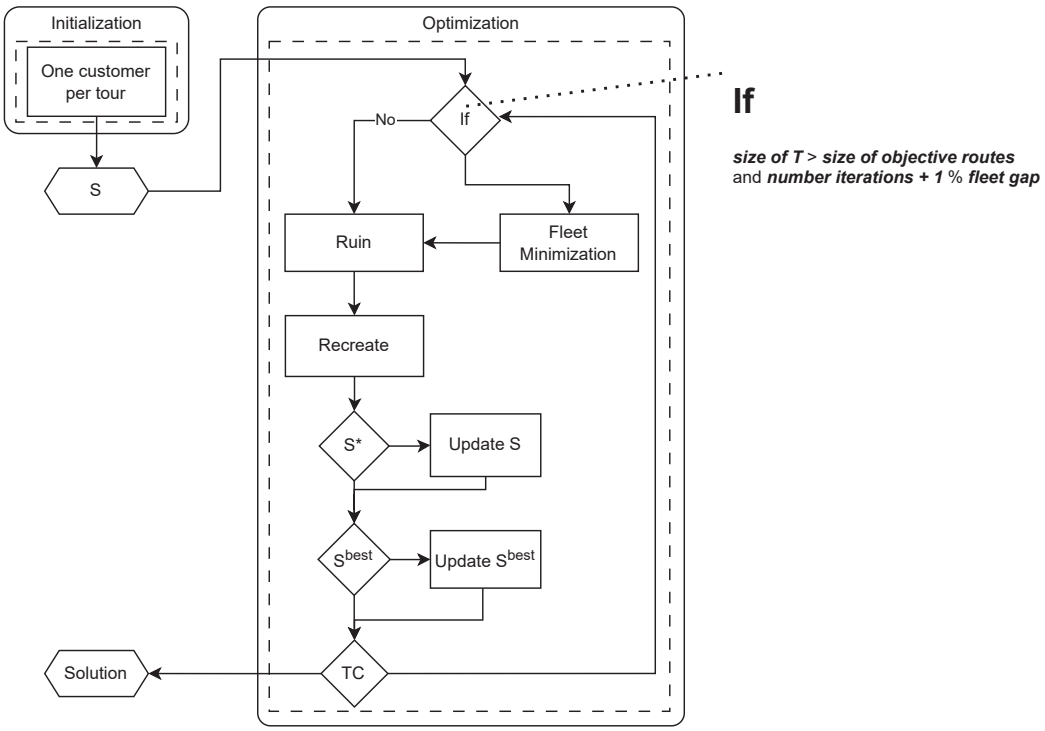



Figure 3.14: A general scheme of the SISR.

$$T_{x+1} = C \cdot T_x, 0 \leq x < f \tag{3.15}$$

$$C = (\frac{T_f}{T_0})^{\frac{1}{f}}, T_0 > T_f > 0 \tag{3.16}$$

The fleet minimization procedure is shown in Figure 3.15. The optimal solution with the fewest vehicles is denoted by $S^*$. Each client $c$ is monitored using the absence counter $abs_c$ to keep track of the number of times they were not included in any tour (i.e. absent from the solution). Initially, this counter is set to 0 for all clients. The following steps are taken in each iteration. First, we obtain a new solution $S^*r$ by using the Ruin and Recreate procedures. This solution is accepted if either (1) the number of absent clients decreases or (2) the sum of absence counters associated with the absent clients $sumAbs(U) = \sum_{c \in U}(abs_c)$ decreases. If the current solution is feasible ($U^* = 0$), the $S^*$ is updated. Afterward, the tour that serves clients with the lowest sum of absence counters is removed from the current solution. Finally, each absent client in the new solution has its associated absence counter incremented.

The Ruin operator ($\mathfrak{R}^-$) is based on two concepts, capacity slack and spatial slack, and three fundamental premises: (1) *Remove a "Sufficient" Number of Customers*, (2) *Remove "Adjacent" Customers*, and (3) *Remove Strings of Customers*. According to Figure 3.16, the process starts by computing $l_s^{max}$ (Equation 3.17), $k_s^{max}$ (Equation 3.18), and $k_s$ (Equation 3.19) related to maximum string cardinality, the maximum number of strings, and the number of strings to be removed respectively.

$$l_s^{max} = min\{L^{max}, \overline{|t \in T|}\}, l_s^{max} \in \mathbb{R}^+ \tag{3.17}$$

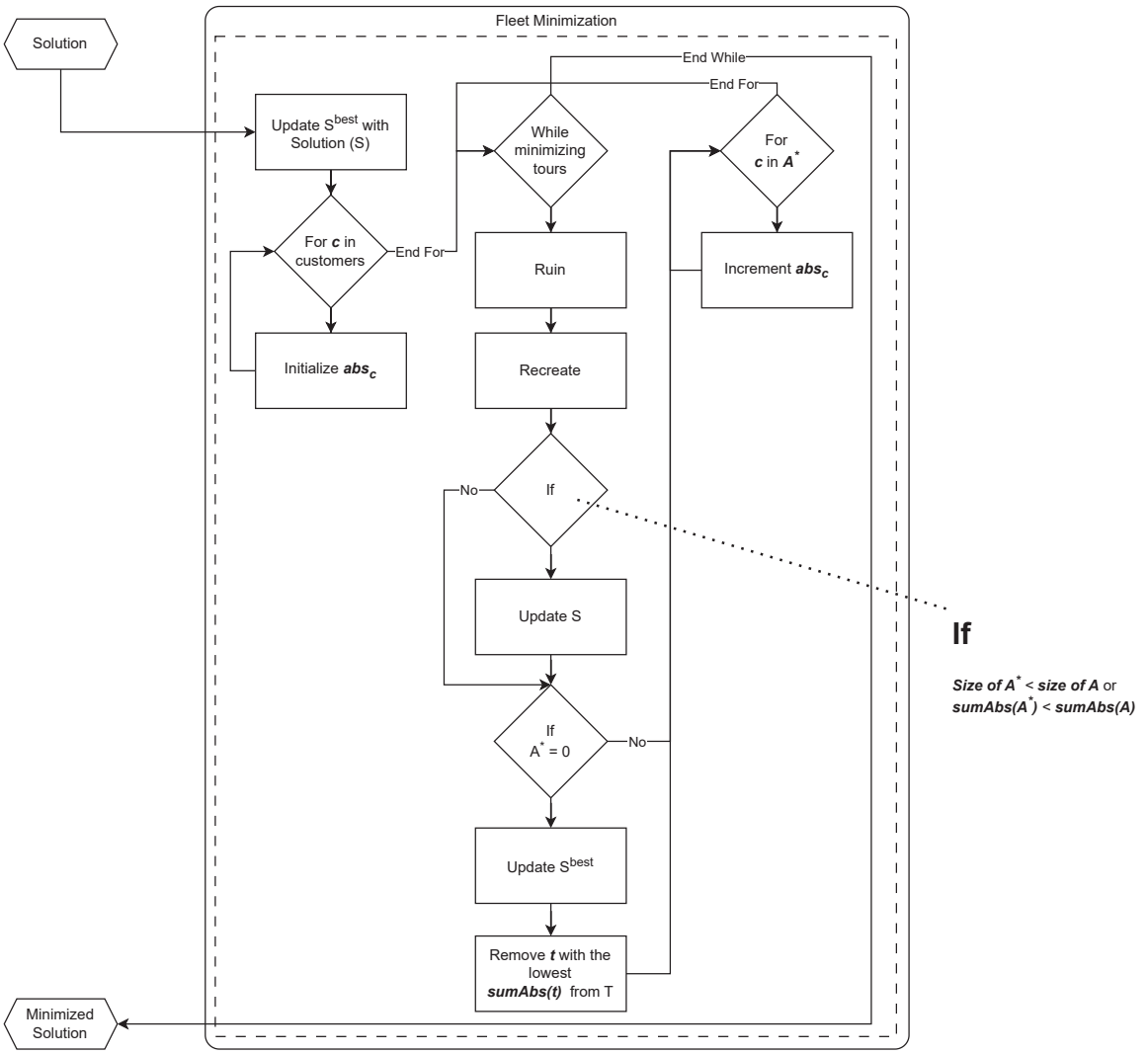



Figure 3.15: A general scheme of the Fleet Minimization of SISR.

$$k_s^{max} = \frac{4\bar{c}}{1 + l_s^{max}} - 1, k_s^{max} \in \mathbb{R}^+ \tag{3.18}$$

$$k_s = \lfloor \mathcal{U}(1, k_s^{max} + 1) \rfloor, k_s \in \mathbb{N}^+ \tag{3.19}$$

The process of removing strings begins by selecting a random customer, labeled as $c_s^{seed}$, to serve as the seed customer. At this point, the set $R$ of tours that have been ruined is initialized. The Ruin operator then examines whether there is a customer $c$ in the adjacency list from $c_s^{seed}$ ($adj(c_s^{seed})$) and whether the set $R$ is smaller than $k_s$. If these conditions are met, the Ruin operator checks whether the customer $c$ is not already in the set $U$ and whether the tour $\tau$ is not in the set $R$. If these conditions are satisfied, the customer $c$ is stored as $c_t^*$, which represents the closest customer to $c_s^{seed}$ in the tour $\tau$. The values of $l_t^{max}$ and $l_t$ are then calculated using Equations 3.20 and 3.21 respectively. Finally, a string is removed from the tour $\tau$ based on $c_t^*$ and $l_t$, and it is stored in the set $U$. The tour $\tau$ is also stored in the set $R$.

$$l_t^{max} = min\{|t|, l_s^{max}\}, l_t^{max} \in \mathbb{R}^+ \tag{3.20}$$

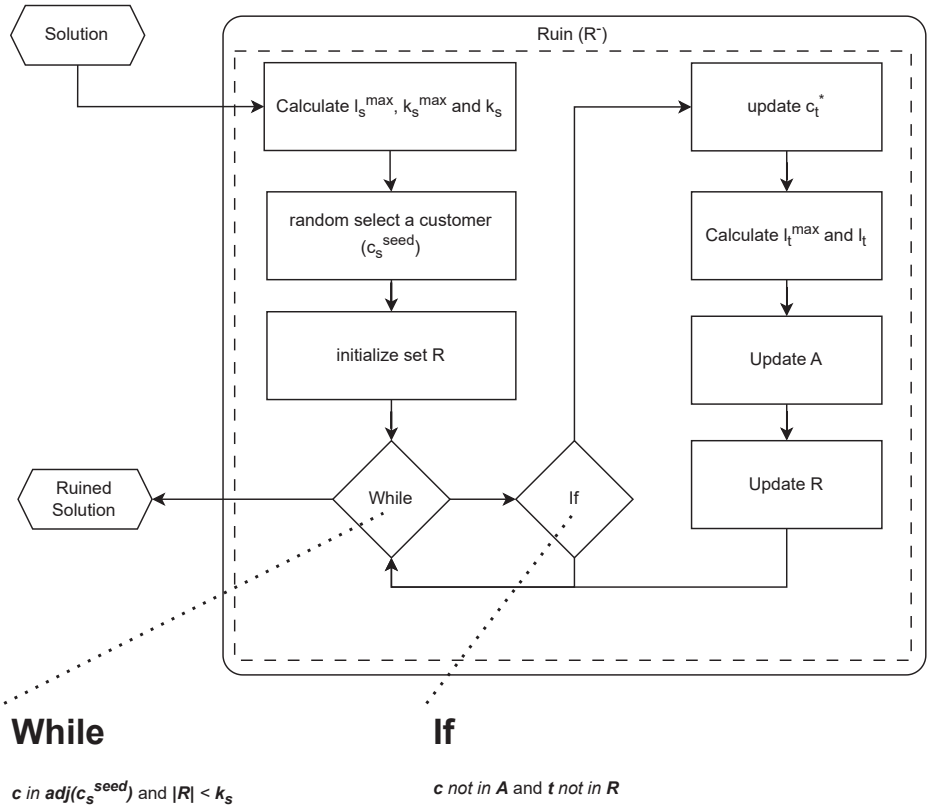



Figure 3.16: A general scheme of the Ruin Operator of SISR.

$$l_t = \left\lfloor \mathcal{U}(1, l_t^{max} + 1) \right\rfloor, l_t \in \mathbb{N}^+ \tag{3.21}$$

During the *string removal* process, a tour $\tau$ containing customer $c_t^*$ will have a randomly selected string of customers removed, with the string's cardinality being $l_t$. The *split string removal* procedure is similar but removes a string of customers with a cardinality of $l + m$, which contains customer $c_t^*$ while preserving a random substring of $m$ customers. To determine the value of $m$, the process starts with $m = 1$ and will maintain that value if a random number $\mathcal{U}(0, 1)$ is greater than the *split depth* parameter $\beta$, or if $m = m^{max} = |t| - l$. Otherwise, the process will increment $m$ and repeat. The probability of using the *split string removal* process is determined by the *split rate* $\alpha$ parameter.

Still on SISR, the Recreate operator, which is shown in Figure 3.17, introduces the concept of *greedy insertion with blinks*. The operator takes the solution $S$ as input and sorts the set of absent customers $U$ based on random, demand, far, or close orders, which are weighted four, four, two, and one, respectively. For each customer $c \in U$, the algorithm tries to insert it into solution $S$ at the optimal position $p$, initially set to $null$. The operator then searches through all tours ($\tau \in \mathfrak{I}$) to locate one with adequate capacity slack to serve $c$. After finding such a tour, the algorithm evaluates each position $p_t$ inside that tour with a probability of $1 - \gamma$, where $\gamma$ is the blink rate. If the cost of inserting $c$ at any position $p_t$ is lower than the current best position $p$, $p_t$ becomes the new best position. If no position is found in the existing tours, the algorithm creates a new empty tour to serve $c$. Finally, $c$ is inserted at $p$ and removed from the set of absent customers $U$. A blink rate of $\gamma = 0$ indicates that customers are always inserted at the best position.

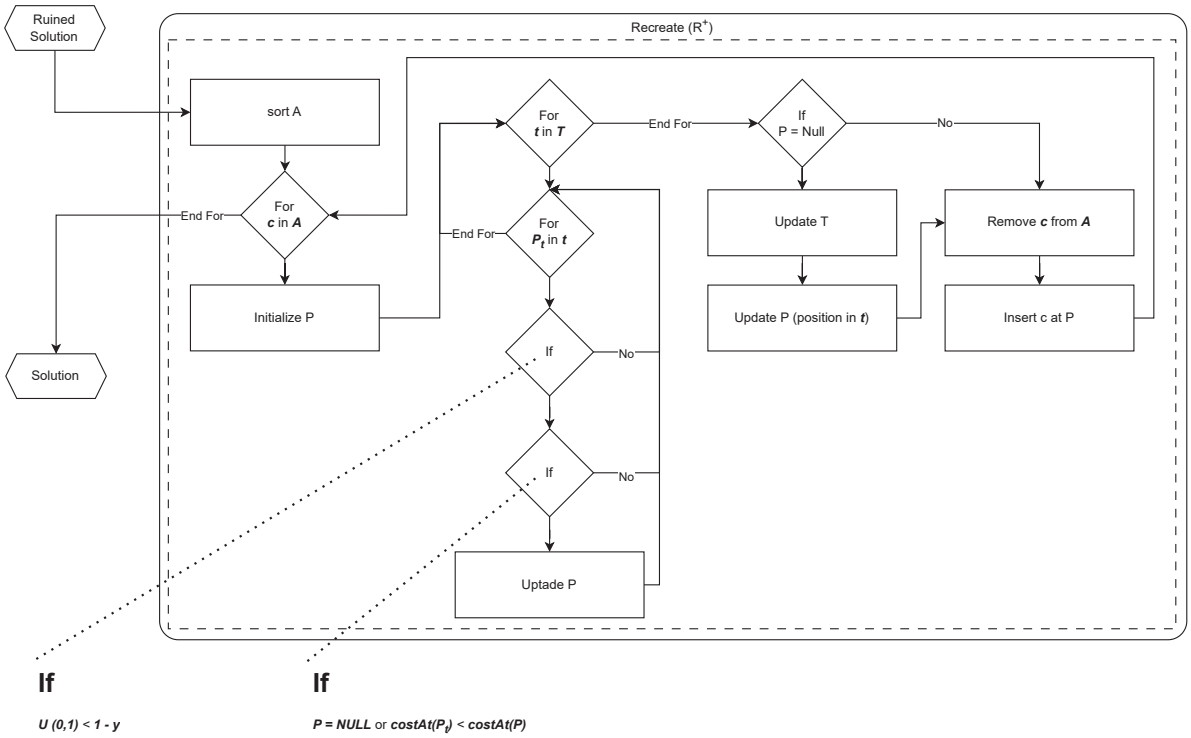



Figure 3.17: A general scheme of the Recreate Operator of SISR.

### 3.7.1 Benefits

The SISR is the simplest technique among the selected for our ensemble. However, its mechanism to escape from local optimums is very interesting and may help to achieve good results throughout the evolutionary process. We highlighted the following aspects:

- **Strategies in the Ruin Operator**: Through the application of string removal and split string removal strategies, and by taking into consideration the customer closest to $c^{seed}$ with a value smaller than $k_s$, the ruin operator has been observed to produce a solution that enables the recreate operator to avoid getting trapped in a local optimum.

## 3.8 ENSEMBLE IN POPULATION-BASED ALGORITHMS

According to past items and Chapter 3, several operators and algorithms are used and can solve many VRPs. As stated by the No Free Lunch theorem (Wolpert and Macready, 1997), there is no better algorithm than the other algorithms in solving all possible optimization problems. In other words, it indicates that it is impossible (theoretically) to develop an algorithm that is superior to all other algorithms in solving different optimization problems and their features (Mallipeddi et al., 2011a).

That said, an ensemble strategy could provide a powerful tool for implementing a flexible approach. In this sense, regarding population-based algorithms, the ensemble can be defined as a combination of different strategies, operators, parameter values, and methods. That means an ensemble can generate better results on a set of optimization problems than a single set of strategies, operators, parameter values, and methods (Wu et al., 2019).

In this sense, according to Figure 3.18, ensembles in population-based algorithms can be classified into three strategies as follows (Wu et al., 2019):

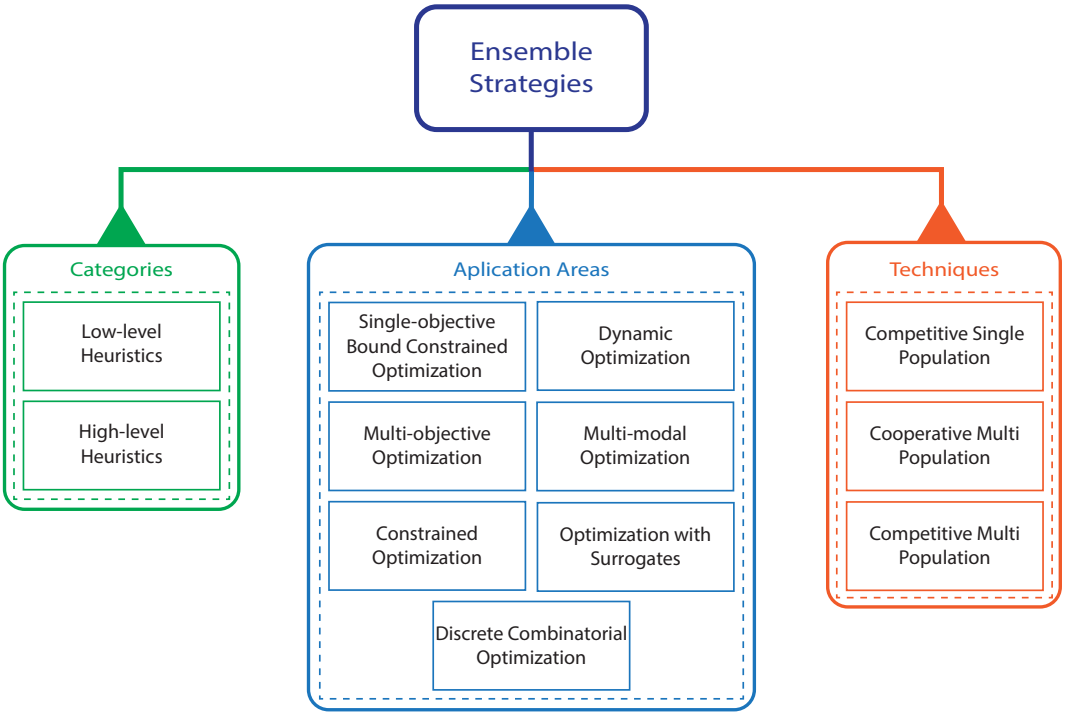



Figure 3.18: An overview of different ensemble strategies (adapted from Wu et al. (2019)).

- **Categories**
  - *Low-level Heuristics*: It is the ensemble of multiple components, including search strategies, parameter values, constraint handling techniques, and neighborhood structures, among others;
  - *High-level Heuristics*: It comprehends the ensemble of different population-based algorithm variations. Also, it is possible to include low ensemble strategies.
- **Application Areas**
  - *Single-objective Bound Constrained Optimization*: It refers to the selection of strategies and their control parameters. However, the selection of a suitable algorithmic configuration is not straightforward, and it requires extensive parameter tuning. Additionally, throughout the evolution process, different algorithmic configurations can be appropriate at different stages;
  - *Multi-objective Optimization*: Concerning a group of solutions to represent the whole Pareto Front appropriately, it is crucial to avoid the clustering of solutions so that the diversity can be improved. In multi-objective optimization literature, $\epsilon$ non-dominance sorting is one such technique that does not allow more than one solution with a difference in all objective values less than $\epsilon$ to be non-dominated by each other. However, the effectiveness of the $\epsilon$ non-domination sorting method depends on the selection of the individual $\epsilon$ values. In other words, different objectives and different optimization algorithms require different $\epsilon$ values to maintain the diversity of the population;
  - *Dynamic Optimization*: It deals with environmental changes. Therefore, it is essential to increase the diversity of the population and guarantee that the algorithm does not converge to react and evolve further when environmental change is detected;

  - *Constrained Optimization*: It is essential to select an appropriate constraint handling technique to effectively solve a problem by exploiting the information present in infeasible individuals. However, according to the No Free Lunch theorem, it is difficult to pick a single constraint-handling technique that can efficiently solve diverse constrained optimization problems. Moreover, while solving a single constrained problem, different constraint-handling methods can be effective during different stages of the search process;
  - *Multi-modal Optimization*: These are related to niching methods, which are regularly used to maintain a diverse population by forming subgroups. In literature, various niching methods such as crowding, sharing, and clearing have been proposed (Wu et al., 2019);
  - *Optimization with Surrogates*: Approximations have been employed in conjunction with population-based algorithms. Thereby, the surrogate models are commonly used to evaluate the competitiveness of different constituent parameter/strategy combinations (Gong et al., 2015; Mallipeddi and Lee, 2015). In other words, surrogate models are an approximation of the original functions, and the accuracy or performance of the surrogate models depends on many factors, such as the availability of a sufficient number of samples.;
  - *Discrete Combinatorial Optimization*: Solving optimization problems with discrete variables depends on destruction and construction procedures that work as a mutation operator and crossover operator.

- **Techniques**
  - *Competitive Single Population*: In a competitive single population ensemble, the individual methods/parameters/strategies compete for resources while operating on a single population;
  - *Competitive Multi Population*: In a competitive multi-population ensemble, each of the constituent methods/parameters is assigned a population. Competitive multi-population ensembles differ in the way the resources are allocated to the different populations;
  - *Cooperative Multi Population*: Generally, the cooperative ensembles are multi-population in nature. In a cooperative ensemble, the individual methods are allocated predefined (sometimes equal) resources, and they try to cooperate with each other by exchanging information.

Regarding **Techniques**, there are three methods to implement them (Wu et al., 2019):

- *Hyper-Heuristic (HH)*: Automates the heuristic design process based on the structure of the problem to be solved. Hence, hyper-heuristics offer benefits such as exploring novel heuristics for a given problem that is unlikely to be created by a human analyst. Furthermore, different heuristics can create a different subset of solutions, which are more likely to be better than those obtained by one general heuristic;

- *Island models*: They are also known as coarse-grained models, multi-deme models, distributed algorithms or portfolio optimization (Huberman et al., 1997). Island models parallelize the evolution process by splitting the population into multiple sub-populations called islands. Thus, the sub-populations on each island evolve separately for most of

the time. However, they exchange solutions between the islands periodically through a process called migration;

- *Adaptive operator selection*: During the credit assignment and operator selection, it is essential to assign a higher probability to operators that produce better offspring (exploitation). Simultaneously, it needs to keep operators that perform poorly for the future search (exploration) by assigning minimum probability (Li et al., 2014b).

## 3.9 HYPER-HEURISTIC AND REINFORCEMENT LEARNING

As previously mentioned, one of the techniques used to select the heuristics in an ensemble is the Hyper-Heuristic (HH). HHs explore the search space of heuristics to avoid getting stuck in local optima solutions. Good HHs must know which is the appropriate heuristic to explore a different area of the search space at the time. Algorithm 1 shows a standard selection Hyper-Heuristic algorithm. The algorithm iteratively selects and applies a heuristic to the current solution and computes the reward. Then, the acceptance criteria decide if the new solution is accepted. Finally, the HH calls the update method of the corresponding selection model.

**Algorithm 1:** Selection Hyper-Heuristic

**Input:** Initial solution $S$
**while** *stop criteria is not met* **do**
  $heuristic \leftarrow SelectHeuristic()$
  $S^r \leftarrow ApplyHeuristic(S, heuristic)$
  $reward \leftarrow GetReward(f(S), f(S^r))$
  **if** *AcceptSolution*($S^r$) **then**
    $S \leftarrow S^r$
  **end**
  UpdateSelectionModel($reward$)
**end**
**return** *Best found solution*

The reward for the selected operator is normally based on recent performance using fitness and diversity measures. Additionally, acceptance criteria must be defined.

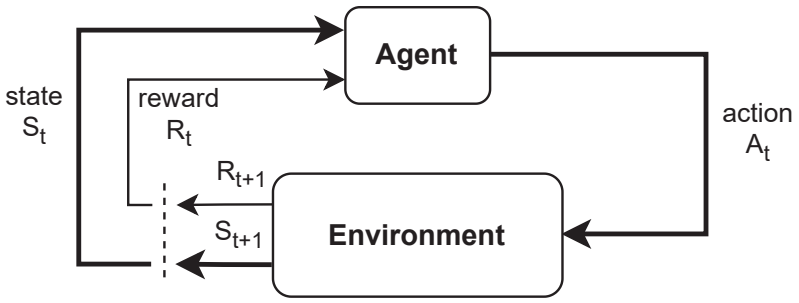



Figure 3.19: Agent-environment interaction of a Markov Decision Process (extracted from Sutton and Barto (2018)).

Reinforcement Learning (RL) techniques (Sutton and Barto, 2018) have been widely researched for applications in HH and Agent-Operator Systems (AOS). RL is a computational approach that learns how to map situations to actions by interacting with an environment (Sutton

and Barto, 2018). Unlike other machine learning paradigms like supervised and unsupervised learning, there is no pre-existing dataset. Instead, the learning agent must be able to sense the state of its environment and decide which action to take based on its observations, with the goal of maximizing a numerical reward signal (Sutton and Barto, 2018).

The task of learning from interaction to achieve a goal can be framed as a Markov Decision Process (MDP). MDPs are a classical formalization of sequential decision-making, in which actions influence not only the immediate rewards but also the subsequent situations (Sutton and Barto, 2018). The interaction between the agent and the environment is illustrated in Figure 3.19. At each time step $t$, the agent receives a representation of the environment's state $S_t$ and selects an action $A_t$ based on that. After acting, the agent moves to a new state $S_{t+1}$ and receives a numerical reward $R_{t+1}$.

The definition of the state is a fundamental component of a Reinforcement Learning system. In general, the state can be any information available to the agent about its environment.

## 3.10 CONCLUDING REMARKS

Many metaheuristics, heuristics, and operators have been proposed to solve a variety of VRPs. This Chapter presented some well-known metaheuristics, e.g., GRASP, GA, SA, and fascinating state-of-the-art approaches. Further, some relevant heuristics (*Split algorithm*) and the EAX were explained in detail.

It is worth mentioning that, although we mentioned the PIX and EAX, we will not use them in this work since they either do not attend to the restrictions addressed in this work or do not compose our approach.

# 4 RELATED WORKS

In the last decades, disasters have been one of the major research subjects due to the significant loss of human lives (Kim, 2014; Yu and Liu, 2014; Le et al., 2015; Maldonado et al., 2017; Kropat and Meyer-Nieberg, 2016; Amit et al., 2016; Kitajima et al., 2016; Kaneko, 2017; Wang et al., 2018). Several approaches of Machine Learning, Evolutionary Computing, and Bio-inspired Computing have been used in several areas and, specifically, in the field of evacuation problem (Shahparvari et al., 2019; Shahparvari and Abbasi, 2017; Goerigk et al., 2015; Atmojo and Sachro, 2017; Zhao et al., 2017; Li and Zhou, 2018). In this literature review, we present some of the most relevant works related to our proposal in the evacuation context and in the VRP.

## 4.1 EVACUATION WORKS

Using a variant of the VRP, Vitali et al. (2017) proposed a method based on the GRASP algorithm to solve a so-called Bus Evacuation Problem (BEP), which uses only buses (public transportation) as a fleet. After that, apply the solution to a real-world scenario based on the wildfire in Valparaiso (Chile).

The solution is represented using an array, where each index corresponds to one of the buses. Each bus is a list of $trips$, which corresponds to the complete route schedule for the respective bus. Therefore, as an initialization method, they use a custom one, which identifies all the possible trips to perform on the graph, i.e., the arcs.

Thereafter, the solution is improved in the GRASP. Regarding the local search phase in the GRASP, Vitali et al. (2017) use an improvement Hill Climbing, with a custom shift local move (i.e., resembling a relocate classic local move). At last, experiments are conducted in two sets of instances generated: (a) a random one and (b) one based on the real case of the Great Valparaiso Fire in Chile 2014.

Still, in a disaster context, Penna et al. (2018) presents a hybridization of a Set Partitioning formulation and a Multi-Start Iterated Local Search with a Randomised Variable Neighbourhood Descent to solve the Heterogeneous Multi-Depot Multi-Trip VRP with Site Dependency, referred to as the Last Mile Distribution Problem (LMDP). The motivation was the Port-au-Prince earthquake in 2010.

Their approach, called MS-ILS-RNVD, is an extension of the work of Subramanian et al. (2012). However, it adds a multi-start mechanism (MS) to diversify the search using different initial solutions. It replaces the local search of classical ILS with an RVND. In addition, another refinement step consisted of memorizing a pool of good routes during the metaheuristic phase to solve a Set Partitioning (SP) problem periodically.

In this sense, as an initialization method, they use a randomized parallel insertion method, which provides MS-ILS-RVND with good and diversified initial solutions. Furthermore, in the perturbation phase, a diversification mechanism is applied based on multiple feasible and randomized swaps or reinsertions, adding or removing $trips$ in a route (multi-trip) and splitting a route into several short routes. Concerning the local search procedure, it is applied five variants of swaps and relocations. In the inter-route moves, Relocate(1,0), Relocate(2,0), Swap(1,1), Swap(2,1) and SWAP(2,2) are used. Also, RVND employs intra-route moves, which are Relocate(1,0), Relocate(2,0), Relocate(3,0), Swap(1,1) and 2-OPT.

Besides, every five ILS iterations, the SP model is built from the solutions gathered in the pool and then solved using a black-box Mixed Integer Linear Programming (MILP) solver.

Experiments are performed on data based on the Port-au-Prince earthquake and the Salhi and Sari (1997) benchmark.

In the same context of the distribution of relief goods, Vieira et al. (2021) proposed an Ant Colony Optimization (ACO) metaheuristic hybridized with an RVND (MACS-RVND) to solve the Emergency Water Trucking (EWT). The EWT is the reactive response to address droughts most adopted by developing countries (Vieira et al., 2021). Besides, the problem was modeled as an MDVRP.

Therefore, the proposed method is designed in two steps. In step one, it used the transportation model to assign demand points to a water source. In step two, it modeled the problem as multiple CVRPs, to address its size and complexity. Hence, in the second step of the developed procedure, results from the MACS-RVND are compared with those obtained by Clarke and Wright heuristics (Clarke and Wright, 1964b) with 2-opt, and with results from an Adaptive Large Neighborhood Search (ALNS) approach proposed by Erdoğan (2017). The ALNS of Erdoğan (2017) diversifies the search by randomly removing customers from the solution and intensifies through reinsertion of the customers and local search by utilization of four local search operators: Exchange, 1-opt, 2-opt, and Vehicle-Exchange.

Regarding MACS-RVND, which is an improved version of the MACS-VRPTW proposed by da Silva Junior et al. (2021), in the RVND step, is used five inter-route moves: Shift(1,0), Shift(2, 0), Swap(1,1), Swap(2,1), Swap (2,2), Cross-exchange and k-shift. Moreover, it uses five intra-route moves: 2-opt, Exchange, and Or-opt1, Or-opt2, Or-opt3. Finally, the approach was applied to a real water distribution case in the Brazilian semi-arid region.

## 4.2 VEHICLE ROUTING PROBLEM WORKS

Prins (2004) proposed a Hybrid GA or MA. Furthermore, he was the first to use a giant TSP tour successfully. Concerning about the initialization method, Prins generates three chromosomes with Clarke and Wright (1964b), Mole and Jameson (1976), and Gillett and Miller (1974). The trips of each solution are concatenated into a chromosome, and then a local search process is applied, followed by the application of the *Split algorithm*. The remaining population is generated by a random permutation after the *Split algorithm* is applied for each one.

Besides, instead of using a mutation operator, he used a local search procedure with a linear ordered crossover. In the local procedure, Prins used the *Split algorithm* combined with insertion (relocate), swap (exchange), 2-opt (intra-route), and 2-opt* (inter-route) moves. Prins also presents a population management mechanism by removing clones (chromosomes with the same fitness) and inserting a new one from his initialization method.

Still on MA or Hybrid GA, Nagata and Bräysy (2009) presented an evolutionary algorithm using the EAX. EAX crossover was originally proposed for the traveling salesman problem by Nagata and Kobayashi (1997). It has shown to be among the most efficient crossover, later extended to CVRP by Nagata and Bräysy (2009). The EAX combines two parents by generating $AB-cycles$, which are used to cut each parent solution. After this step, the sub-paths in each parent are connected again with a greedy method (in Nagata and Kobayashi (1997, 2013)) or, in CVRP Nagata and Bräysy (2009), by 2-opt*. It is worth mentioning, that EAX can link edges that were not inherited. However, the offspring can be very similar to the parents, which leads to a loss of diversity.

Moreover, Nagata and Bräysy also used as a representation of a giant TSP tour, later converted into sub-tours (routes). As an initialization method, they generate a solution through saving costs heuristics, then apply a local search (slightly different from the original one, by evaluating the moves with the penalty function). After, is performed a modification mechanism,

followed by the original local search. The goal of the modification mechanism is to eliminate capacity and duration violations of a solution.

Following through the process, once the EAX is applied, infeasible offspring can be generated, then the modification mechanism is applied. After that, the local procedure is executed. They extended the work to solve the VRPTW (Nagata and Bräysy, 2009). In this work, they introduce a guided local search with ejections to improve the past approach.

Inspired by some ideas of Prins (2004) and Nagata and Bräysy (2009), Vidal et al. (2013) proposed a Hybrid Genetic Search with an Adaptive Diversity Control (HGSADC). The HGSADC follows the principle of using a giant TSP tour and a local procedure called *Education*. In the *Education* procedure, they use *Split algorithm* (Prins, 2004) and four local movements: swap (exchange), relocate, 2-opt, and 2-opt*. Thereby, the population is initialized randomly, and then they apply the *Education*. If the solution is not feasible, they repair it. Still, in the crossover phase, Vidal et al. uses an ordered crossover for most VRPs. For the Periodic VRP (PVRP), they proposed a new crossover called *Periodic Crossover with Insertions* (PIX).

Further, they use relaxation as shown in (Nagata and Bräysy, 2009). The *Adaptive Diversity Control* used the fitness function, called *biased fitness function*, which uses the Hamming distance and a penalty function (due to relaxation) to evaluate the chromosome. Further, promotes the *survivor selection*, which determines which *pop size* chromosomes will go to the next generation. First, it removes the clones, which are chromosomes with the same fitness (*biased fitness*), and then the worst chromosomes. Second, a *diversification* process, which keep the best $\frac{pop\ size}{3}$ chromosomes, and introduce new $4 \times pop\ size$ chromosomes. Finally, the *survivor selection* is applied again.

Li et al. (2014a) presented a hybridization of GA (HGA-ALS) and an Adaptive Local Search. After generating the initial population, parents are selected by binary tournament and then crossed by an improved ordered crossover. The mutation operator was substituted by an Adaptive Large Neighborhood Search (ALNS), which uses ten local movements, two destroy operators, and one repair operator called *Greedy Insertion based on Probability Assignment* (GIPA). Rather than partition the solution into different sub-solutions, their adaptive mechanism selects the best neighborhood method (local movements) to improve the current solution.

Concerning neighborhood metaheuristics, Sze et al. (2016) proposed a hybridization of an Adaptive Variable Neighborhood Search (AVNS) with a Large Neighborhood Search (LNS). Basically, their AVNS-LS has two phases: a learning phase (stage 1) and a multi-level Variable Neighborhood Search (VNS) with a guided local search (stage 2). The adaptive feature is performed in the local search process, which is a set of insertion, exchange, interchange, swap, cross-exchange, 2-opt, 2-opt*, and a cross-tail (originally proposed by Jun and Kim (2012)).

The initial solution is generated by the saving method of Clarke and Wright (1964b), and then it is refined using 2-opt and 2-op*. Thereafter, in stage 1, a shaking procedure is done by three criteria: (a) distance between customers, (b) distance between the depot and customers, and (c) angle between the depot and customers. Then, *Best improvement local search* is applied, and for each operator, its score is stored along with the probability of selection. This information is used in stage 2 and configures the learning phase. At the end of stage 1, Dijkstra's algorithm is used as a post-optimiser. As a diversification strategy, Sze et al. (2016) uses an LNS, which uses four destroy operators: Gain ratio deletion, Overlapping deletion, Worst-edge deletion, and Conflicting sector deletion. They used a greedy least-cost insertion repair operator in the solution reconstruction process.

On stage 2, all stage 1 steps are done, but a multi-level process is inserted in the place of the *Best improvement local search*. The multi-level improvement approach mimics the local search in the VNS. In addition, it adopts a mechanism that balances between the best and the

first improvement strategies, which is defined as the $k^{th}$ improvement. So, the main difference between the two stages of the AVNS algorithm is in the local search procedure.

Again, on the population-based metaheuristics, Gutierrez et al. (2018) presents an MA with a GRASP to solve the VRP with stochastic demands. As a chromosome, they used a twofold representation, which is a giant TSP as in Prins (2004), and the detailed routes. Therefore, to decode the giant TSP into a set of routes, they also use the *Split algorithm* from Prins (2004). The initial population is generated by three methods: Clarke and Wright (1964b), Gillett and Miller (1974) and best insertion heuristic. Moreover, a restart procedure is proposed based on the GRASP, where the GRASP generates $\frac{(Popsize-1)}{2}$ individuals, and the remaining ones are randomly generated.

Still, individuals are selected by a binary tournament, and then an ordered crossover is applied. They also apply a random mutation operator right after the *Split algorithm*. A Variable Neighborhood Descent (VND) is used in the local search procedure, along with OR-opt, 2-opt, and inter-cross local moves.

In turn, Rabbouch et al. (2019) proposed a GA with a recombination step for the MDVRPTW with a heterogeneous fleet. Their GA is a typical GA, and the population is generated by assigning customers to the nearest *m* depots. After, a greedy heuristic is applied to fulfill the routes. Also, it uses the route-based crossover (recombination) proposed byPotvin and Bengio (1996), followed by a repair procedure right after, and then a swap as mutation operator. Although their GA's results are not competitive, in some instances of the tested benchmarks, it achieved a GAP of 3.02% considering that Rabbouch et al. (2019) does not use any local search mechanism.

On the other hand, Liu and Jiang (2019) presents a hybridization of ALNS with GA. The initial solution is generated by a Quick insertion heuristic, and its representation is a set of routes. After, a cross-exchange is applied, followed by an in-out heuristic. The in-out heuristic is composed of two destroy operators (Shaw removal and Worst removal) and one repair operator (best insertion repair). After, an intra-route process is applied, where is used the OR-opt local move. If the solution generated by the ALNS meets the diversity criterion, then it is inserted into the population.

In the GA phase, the parents are selected randomly, and an ordered crossover is applied. In this process, the solution that came from the ALNS is converted into a giant TSP, and at the end of the process, it is reconverted into a set of routes by the *Split algorithm*. At the end of the GA phase, the worst $\frac{popsize}{2}$ are deleted, and the whole process starts again.

Later on, Abdallah and Ennigrou (2020) proposed a hybrid solution, composed of a Particle Swarm Algorithm (PSO), a GA, and a MA, to solve the MDVRPTW with a heterogeneous fleet. In the GA, they used the ordered crossover and exchange mutation operator, and in the MA, in the local procedure, a $\Lambda$-exchange method inspired by Shi et al. (2017) is used. To communicate between the metaheuristics called agents, they used an ACL message protocol every time an agent improves its best result.

In the same sense, Zhen et al. (2020) formulated a MIP model to solve small instances of the Multi-Depot Multi-Trip VRPTW and release dates. For large instances of the problem, they presented two solutions, a PSO and a GA, both hybridized with a Local Search Variable Neighborhood Descent (LS-VND). In the PSO, each particle is described as a two-dimensional array, where the first array is with the customer id order, and the second corresponds to the customer's random position. Regarding the local search procedure, they used a VND with three local moves: Reinsertion, Exchange, and Reverse.

In the GA, each chromosome has the same structure as a particle of the PSO, a two-dimensional array. As a final step of the initialization, they group the customers by the depot.

Moreover, they proposed a custom crossover. Instead of using a mutation operator, they apply the same VND used with the PSO. Besides some customization at crossover, Most of their ideas for the GA are inspired from Vidal et al. (2012).

Máximo and Nascimento (2021) proposed an Adaptive Iterated Local Search with a Path Relinking (AILS-PR) to solve the CVRP. In their approach, an initial solution is generated by a custom constructive heuristic, and it is a set of routes with a set of customers in each route. The process continues with a perturbation mechanism composed of four removal operators (Concentric removal, Proximity removal, and Vertex Sequence removal) and two reinsertion operators (Proximity insertion and Cost insertion). After that, a local search procedure is applied with four local moves: Shift, Swap, 2-opt, and 2-opt*. The local search procedure is performed both in an intra and inter-route approach. Regarding the adaptive feature, Máximo and Nascimento (2021) stores the information of how successfully each perturbation operator was, followed by an acceptance criterion. It is worth mentioning that an elite set of solutions is kept during this phase.

Thereafter, the Path Relinking phase starts. Two solutions with the same number of routes are selected from the elite set of solutions, defining one as the initial solution and the other as the guide solution. The Path Relininkg of Máximo and Nascimento (2021) is combined with a Tabu Search metaheuristic. Hence, the main idea of this combination is to investigate the paths between two solutions in order to find better-quality intermediate solutions.

The Slack Induction by String Removals (SISR) approach introduced by Christiaens and Berghe (2020), while rooted in fundamental operators like ruin and recreate, demonstrates competitive performance compared to FILO and HGSADC. The SISR algorithm initializes a solution, denoted as $S = T, A$, with tours and a set of unserved customers. Simulated annealing is employed, gradually reducing temperature to explore the solution space. Fleet minimization is conducted, aiming to minimize the number of vehicles. The fleet minimization process involves iteratively updating a solution based on ruin and recreate procedures. The absence counter monitors unserved customers, guiding solution changes. Simulated annealing acceptance criteria determine solution updates.

The Ruin operator is based on capacity and spatial slack, removing strings of customers following key premises. It dynamically determines string removal parameters, such as maximum cardinality and number of strings. The Recreate operator introduces "greedy insertion with blinks," sorting absent customers and probabilistically inserting them into tours, with blink rates affecting insertion behavior. SISR's effectiveness lies in its strategic use of ruin and recreates procedures, simulated annealing, and innovative string removal and insertion strategies. This combination contributes to its competitive performance in solving VRPs.

Recently, The Fast Iterated Local Search Localized Optimization (FILO), proposed by Accorsi and Vigo (2021), follows a structured process for solving a set of VRPs. The algorithm begins with an Initialization phase, constructing an initial feasible solution using a modified savings algorithm of Clarke and Wright (1964a). Subsequently, an improvement phase employs Route Minimization to reduce routes if necessary and an Optimization procedure using Iterated Local Search (ILS) principles.

The Route Minimization procedure dynamically adjusts the number of routes in the solution, guided by a heuristic solving a bin-packing problem. During iterations, pairs of routes are selected, and customers are temporarily removed, enabling route reshaping. The algorithm uses specific criteria for route selection and employs a probability-based decision for creating new routes.

The Optimization module employs a Hierarchical Randomized Variable Neighborhood Descent (HRVND), combining random and fixed neighborhood exploration. The HRVND

explores tiers of local search operators, progressing to more expensive tiers if needed. A shaking step initiates a random walk, altering routes, and a recreate step reintegrates removed customers with cost minimization.

Ruin intensity control adapts the length of the random walk iteratively. Simulated annealing guides the acceptance of new solutions based on cost differentials and a temperature parameter. Dynamic move generators and sparsification parameters contribute to vertex-wise adjustments during optimization.

The algorithm's cache management tracks vertices for efficiency, updating based on non-improving iterations or discovering better solutions. The entire process iterates until a solution is obtained, maintaining the best solution throughout. Overall, FILO integrates constructive heuristics, local search, and metaheuristics for effective combinatorial optimization.

## 4.3 CONCLUDING REMARKS

In the literature review, especially in the evacuation context, the works are not entirely clear about their performance. Most of it is due to the lack of comparative studies with other approaches in the same problem or related VRP approaches (Esposito Amideo et al., 2019; Shahparvari et al., 2019; Shahparvari and Abbasi, 2017; Goerigk et al., 2015; Atmojo and Sachro, 2017; Zhao et al., 2017; Li and Zhou, 2018; Vitali et al., 2017; Vieira et al., 2021). Although Penna et al. (2018) perform a comparison with other related VRP approaches, their results are not good as the state-of-the-art. Hence, we searched into related VRP works to comprehend better how the problem is modeled, the metaheuristics involved in how they are combined (or not), and their relation with used operators.

Tables 4.1 and 4.2 present the summarization of our literature review through two perspectives as follows:

- **The metaheuristic applied**: Along with selection, diversity control mechanism, and problems that the approach solves.
- **The operators applied**: Considering not just local movement operators but also crossovers and mutation operators.

On the one hand, according to Table 4.1, the majority of works use GA or some hybridization with it (i.e., MA). In this sense, there are some fascinating works (Nagata and Bräysy, 2009; Vidal et al., 2013; Christiaens and Berghe, 2020; Máximo and Nascimento, 2021; Accorsi and Vigo, 2021), which, some of them constitute our baseline in this study. Those works share some important aspects of a sound performance of a GA as diversity control. However, only Nagata and Bräysy (2009) had proposed an outstanding recombination method (EAX). Despite that, as mentioned earlier, the EAX can generate infeasible and similar to the parent's offspring.

On the other hand, all works, except Rabbouch et al. (2019) (which results are not competitive), rely on local move operators along the evolutionary process to achieve competitive results, as shown by Table 4.2. Local move operators tend to be the bottleneck of the approach's performance.

Table 4.1: Evacuation and related VRP approaches - Metaheuristics

| **Authors** | **Initialization / Representation** | **Selection** | **Diversity Control** | **Metaheuristics** | **VRPs** |
|---|---|---|---|---|---|
| Prins (2004) | Three chromosomes: Clarke and Wright (1964b), Mole and Jameson (1976), and Gillett and Miller (1974) combined with a local search. Remain pop is generated by random mutation / Giant TSP tour without delimiters | Binary Tournament | Population Management Mechanism | MA | CVRP |
| Nagata and Bräysy (2009); Nagata and Kobayashi (2010) | Custom initialization method: two local procedures and a modification mechanism / Twofold: an array of routes, and in each route an array of customers | Random | Replacement strategy | MA | CVRP and VRPTW |
| Vidal et al. (2013) | Random followed by *Education*, and a repair procedure / Twofold: a custom pattern and a giant TSP tour without delimiters | Binary tournament | Adaptive Diversity Control with Survivor Selection | MA | CVRP, VRPTW, PVRPTW, MDVRPTW, and Site Dependency VRPTW |
| Li et al. (2014a) | First, assign each customer to its closest depot, then cheapest-insertion heuristic is applied to construct a tour permutation / Giant TSP tour with delimiters. | Binary Tournament | Hamming distance | GA and ALNS | MDVRPTW |
| Sze et al. (2016) | Clarke and Wright (1964b), then refined using 2-opt and 2-op* / Twofold: an array of routes, and each route is an array of customers | AVNS | LNS | AVNS and LNS | CVRP |
| Vitali et al. (2017) | GRASP / Twofold: an array of buses, and in each bus is an array of evacuee | not apply | not apply | GRASP | BEP |
| Penna et al. (2018) | Randomised parallel insertion / Twofold: an array of routes, and in each route an array of customers | not apply | Multi-start mechanism | ILS and RVND | LMDP |
| Gutierrez et al. (2018) | Four methods are applied: Clarke and Wright (1964b), Gillett and Miller (1974), best insertion, and GRASP / Twofold: giant TSP tour and decoded into routes | Binary Tournament | Clone deletion | MA and VND | VRP with Stochastic Demands |
| Rabbouch et al. (2019) | First, assign customers to their nearest depot, then a greedy heuristic is applied to fulfill the routes / Giant TSP tour without delimiters. | Binary tournament | not apply | GA | HMDVRPTW |
| Liu and Jiang (2019) | Quick insert-based heuristic / Set of routes in the ALNS phase, later converted to a giant TSP. At last, reconverted to a set of routes by *Split algorithm* | Random selection | Multi-start scheme (based on *broken pair distance*) | ALNS and GA | Cum-CVRPTW |

| | | | | | |
|---|---|---|---|---|---|
| Abdallah and Ennigrou (2020) | Randomly generated / Giant TSP with delimiters | Tournament selection | not apply | PSO, GA, and MA | HMDVRPTW |
| Zhen et al. (2020) | Randomly generated / Two-dimensional array with a permutation of customers, and the depot of each customer | not mentioned | Two criteria: *broken pair distance* and Clone deletion | PSO and GA: both hybridized with VND | Multi-Depot Multi-Trip VRPTW and release dates |
| Vieira et al. (2021) | Generated by some method that solves the VRPTW / Giant TSP with delimiters | not mention | not apply | ACO and RVND | EWT (modeled as a MDVRP) |
| Christiaens and Berghe (2020) | Randomly generated one customer per route | not apply | Ruin & Recreate operators and Acceptance criterion | SA | CVRP, VRPTW, MDVRP, VRP with Backhauls, OVRP, Cum-VRP, and PDPTW |
| Máximo and Nascimento (2021) | Custom Constructive heuristic / Set of routes | not apply | Two mechanisms are used: Degree of perturbation and Acceptance criterion | AILS, Path Relinking, and Tabu Search | CVRP |
| Accorsi and Vigo (2021) | A adaptation of Clarke and Wright (1964b) algorithm | not apply | *shake*, *recreate* steps, and acceptance strategy | SA, ILS and HRVND | CVRP |

Table 4.2: Evacuation and related VRP approaches - Operators

| **Authors** | **Local Move Operators** | **Crossovers** | **Mutation** |
|---|---|---|---|
| Prins (2004) | Insertion (Relocate), Swap (Exchange), 2-opt, and 2-opt* | Linear Ordered Crossover | not apply |
| Nagata and Bräysy (2009); Nagata and Kobayashi (2010) | 2-opt, 2-opt*, Insertion (Relocate), Or-exchange, and Swap (Exchange) | EAX | not apply |
| Vidal et al. (2013) | Swap (Exchange), Relocate, 2-opt, and 2-opt* | Ordered Crossover and Periodic Crossover with Insertions | not apply |
| Li et al. (2014a) | Inter-depot move (Relocate), Inter-depot Exchange-Cross (Cross-Exchange), Inter-depot 1-0 exchange (Relocate), Inter-depot 1-1 exchange (Exchange), Inter-depot LNS, Intra-depot move (Relocate), Intra-depot Exchange-Cross (Cross-Exchange), Intra-depot 1-0 (Relocate), Intra-depot 1-1 (Exchange), Intra-depot LNS, Related removal, Infeasible routes removal, Greedy Insertion based on Probability Assignment (GIPA) | Improved Ordered Crossover | ALNS |
| Sze et al. (2016) | Insertion, Exchange, Interchange, Swap, Cross-exchange, 2-opt, 2-opt*, and Cross-tail | not apply | not apply |
| Vitali et al. (2017) | Shift (Relocate) | not apply | not apply |
| Penna et al. (2018) | Inter-route: Relocate(1,0), Relocate(2,0), Swap(1,1), Swap(2,1), and Swap(2,2); Intra-route: Relocate(1,0), Relocate(2,0), Relocate(3,0), Swap(1,1), and 2-OPT | not apply | not apply |
| Gutierrez et al. (2018) | OR-opt, 2-opt, and Inter-Cross | Ordered Crossover | RVND |
| Rabbouch et al. (2019) | not apply | Route-based Crossover | Swap |
| Liu and Jiang (2019) | Cross-exchange, Exchange, OR-opt, Shaw removal, and Worst removal | Ordered Crossover | not apply |
| Abdallah and Ennigrou (2020) | Λ-Exchange method inspired by Shi et al. (2017) | Ordered Crossover | Exchange |
| Zhen et al. (2020) | Reinsertion, Exchange, and Reverse | Custom Crossover | not apply |
| Vieira et al. (2021) | Exchange, 1-opt, 2-opt, Vehicle-Exchange, Shift(1,0), Shift(2, 0), Swap(1,1), Swap(2,1), Swap (2,2), Cross-exchange, k-shift, Or-opt1, Or-opt2, and Or-opt3 | not apply | not apply |
| Christiaens and Berghe (2020) | not apply | not apply | not apply |
| Máximo and Nascimento (2021) | Concentric removal, Proximity removal, Vertex Sequence removal, Proximity insertion, Cost insertion, Shift, Swap, 2-opt, and 2-opt* | not apply | not apply |
| Accorsi and Vigo (2021) | 10EX, 11EX, SPLIT, TAILS, TWOPT, 20EX, 21EX, 22EX, 20REX, 21REX, 22REX, 22REX*, 30EX, 31EX, 32EX, 33EX, 30REX, 31REX, 32REX, 33REX, 32REX*, 33REX*, and ejection-chain (EJCH) | not apply | not apply |

Although it is possible to design better approaches for one or a set of specific VRPs, evidence frequently shows that researchers can hardly design one that can generate better results than all other state-of-the-art (Wu et al., 2019). In this sense, it could be difficult for a practicing

engineer to know which algorithm is suitable for a new VRP. Further, one may find adapting a particular algorithm to this new VRP unique feature hard.

With that being said, the ensemble approach is one of the most promising strategies according to Wu et al. (2019). Besides, it has shown many robust and adaptable population-based ensemble variants for many optimization problems (Wang and Li, 2010; Zhao et al., 2012; Wu et al., 2016). Therefore, a refined population-based ensemble with practical and outstanding components may deal with different variations of VRP. Moreover, as far as we know, there is only one work with ensemble (Wang et al., 2021) in the VRP context.

After the discussion above, we proposed GREEVO - a GRASP Ensemble Evolutionary Algorithm. GREEVO combines an ensemble of high-level techniques (HGSADC, FILO, and SISR) with a Q-learning HH to select the appropriate technique during the evolutionary process which will be detailed in the chapter 5.

# 5 GREEVO: A GRASP ENSEMBLE EVOLUTIONARY ALGORITHM

The Capacitated Vehicle Routing Problem (CVRP) is a well-known optimization problem in which a fleet of vehicles with limited capacity must serve a set of customers with known demands while minimizing the total distance traveled. The problem is NP-hard, which means that finding the optimal solution for large instances is computationally intractable.

According to past items, several operators and algorithms are used and can solve many VRPs. As stated by the No Free Lunch theorem (Wolpert and Macready, 1997), there is no better algorithm than the other algorithms in solving all possible optimization problems. In other words, it indicates that it is impossible (theoretically) to develop an algorithm that is superior to all other algorithms in solving different optimization problems and their features (Mallipeddi et al., 2011a).

With that being said, we present the **GREEVO**, a Grasp Ensemble Evolutionary algorithm. As shown by Figure 5.1, the flow of our approach starts with the **Initialization**, which generates individuals through a GRASP. After, the *Split Algorithm* is applied.

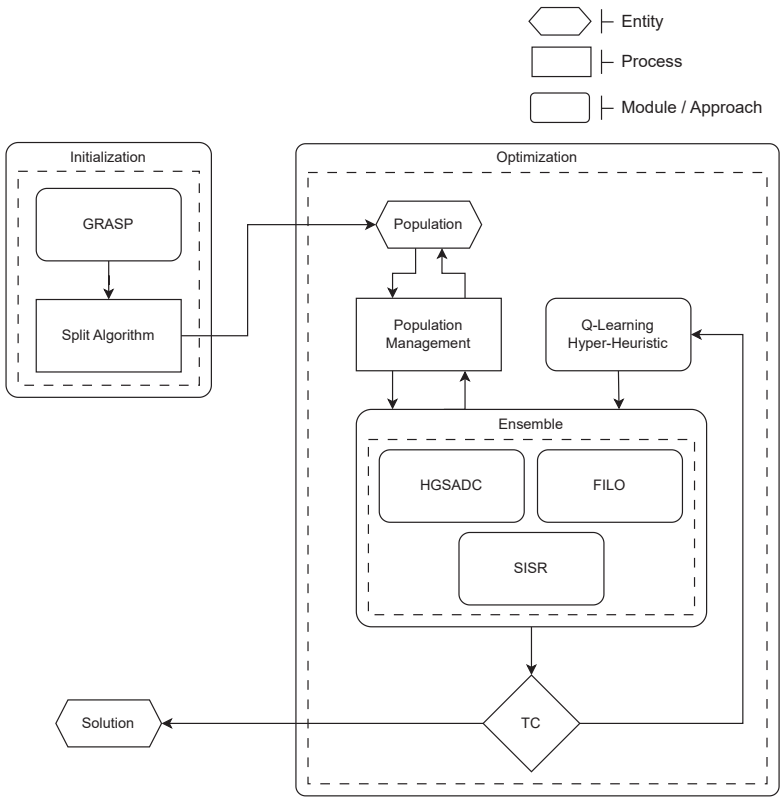



Figure 5.1: GREEVO Architecture

Once the population reaches its maximum size, the evolutive process is initiated. While each baseline approach has its management solution and population mechanism, the **Population Management** process adopts a cooperative strategy (as described in Section 5.3), resulting in a single population. The **Population Management** is responsible for passing the solution or a population of solutions to the selected approach and maintaining the best solutions found during the evolutive process.

The **Q-Learning Module** (Section 5.2) selects one of the baseline approaches. During the **Optimization** process, the search strategy needs to be updated to suit the search process as the search landscape changes during the population evolution towards the global optimal solution. Hence, an ensemble of multiple strategies (baseline approaches) with a proper adaptation mechanism (**Q-Learning Module**) could enable a selection of the most appropriate strategy during the optimization process (Mallipeddi et al., 2011b).

Finally, by the end of the **Optimization** process, the best overall solution is retrieved.

## 5.1 GRASP - INITIALIZATION

The solution format that we adopted was inspired by Prins (2004), which was the first to use this representation successfully. The format involves a giant TSP tour without delimiters as a chromosome, where each allele in the chromosome represents a customer. However, it requires an algorithm to find a chromosome segmentation in routes, retrieving the cost and the solution. To achieve this, we used the *Split algorithm* from Prins (2004).

Our GRASP algorithm generates only feasible solutions. To randomize the selection of the nearest customers to the depot, we use a custom restricted candidate list (RCL). The size of the RCL is determined by an adaptive random factor (Equation 5.1), which is used in the *RandomChoice()* method. The rest of the chromosome is constructed using a greedy approach. Finally, the chromosome is optimized using the *Split algorithm*.

$$random\_factor = \frac{total\_demand}{vehicle\_max\_capacity} \tag{5.1}$$

The Algorithm 2 creates a set of available customers ($n$) with all unserved customers and the maximum number ($k$) of available vehicles. The algorithm checks if the selected vehicle still has the capacity to serve the available customers. Two methods are used to select the next customer to be added to the $trip$, namely, (i) Random Choice($n$) and (ii) Closest Customer($trip, n$). In the Random Choice method, the available customers are sorted by distance. Half of the sorted available customers are chosen randomly (*random factor*) to ensure diversity in the initial population. The *ClosestCustomer()* method prioritizes the distance between the last customer added to the $trip$ and the next one. The algorithm stops when it creates a feasible chromosome.

After a solution is generated by our GRASP, it is optimized by the *Split Algorithm*, which was proposed by Prins (2004). The *Split algorithm* is a heuristic used for solving vehicle routing problems. This algorithm follows a route-first cluster-second approach wherein a TSP tour is constructed first, ignoring any side constraints. The tour is split into feasible vehicle routes in the second phase. According to a survey conducted by Prins et al. (2014), around 70 recent articles have used this technique.

The objective of *Split Algorithm* is to partition the giant TSP tour into $m$ disjoint sequences of consecutive visits. Therefore, each sequence is associated with a route originating from the depot. After, it visits its respective customers and then returns to the depot. In addition to that, the total distance of all routes should be minimized.

## 5.2 Q-LEARNING HYPER-HEURISTIC

According to (Blum et al., 2011), heuristic approaches are extremely valuable for solving intricate optimization problems in real-world situations where precise methods may be impractical. However, the efficacy of these techniques is heavily reliant on appropriate configuration settings that are tailored to fit the specific problem domain. To tackle this challenge, Burke et al. (2010)

**Algorithm 2:** GRASP algorithm

```
while chromosome is empty do
    n ← available customers;
    K ← max number of vehicles;
    k ← 1;
    k_capacity ← Q;
    k_duration ← L;
    while K > k and size of n > 0 do
        trip ← empty list;
        while k_capacity > 0 and k_duration > 0 do
            if trip is empty then
                c ← RandomChoice(n);
            end
            else
                c ← ClosestCustomer(trip, n);
            end
            k_capacity ← k_capacity - c_demand;
            k_duration ← k_duration - distance(trip, c) - c_time service;
            append c into trip;
            remove c from n;
        end
        k ← k + 1;
        if trip is not empty then
            append trip into chromosome;
        end
    end
    if chromosome is not feasible then
        chromosome ← empty list;
    end
end
return chromosome
```

said that there are a number of adaptive search methodologies available, which are commonly referred to as Hyper-Heuristics (HH) or Adaptive Operator Selection (AOS) in the literature.

In this sense, Dantas and Pozo (2022) proposed an approximate Q-learning algorithm, which uses an Artificial Neural Network as a function approximator. The Q-learning algorithm of Dantas and Pozo (2022) was chosen according to the following criteria:

- **Promising approach**: In research conducted in collaboration with Dantas et al. (2021), we carried out a series of experiments on some optimization problems using the Hyflex Framework, comparing it with other heuristic methods, where Q-learning proved to be a very promising alternative.
- **Standout perform on VRP**: In tests conducted to confirm this in our scenario, we compared FRRMAB, which according to our knowledge was the state-of-the-art in comparisons of heuristic methods, Q-learning had better performance.

We adapted their HH as shown by Figure 5.2.

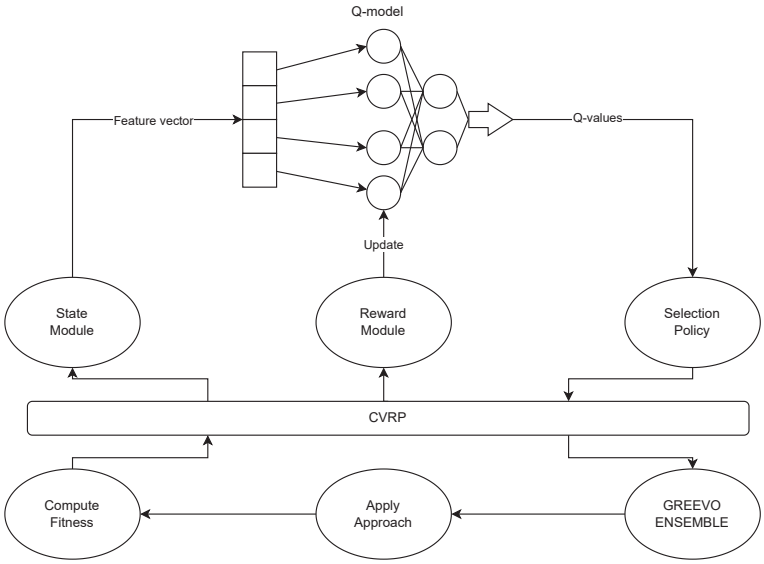



Figure 5.2: Diagram of the Q-learning (adapted from (Dantas and Pozo, 2022)).

Regarding the State Module, we followed the Dantas and Pozo (2022) recommendation by using the *Fitness Improvement Rate* ($FIR$) and *Elapsed Time*, given by the Equations 5.2 and 5.3 respectively.

$$FIR_{i,t} = \frac{pf_{i,t} - cfi,t}{pf_{i,t}} \tag{5.2}$$

where $pf_{i,t}$, is the fitness value of the original solution, and $cfi,t$ is the fitness value of the offspring.

$$Elapsed\ Time = \frac{current\ time}{max\ time} \tag{5.3}$$

Additionally, the Reward Module is given by Equation 5.2 between the current and the previous solutions. As stated by Dantas and Pozo (2022), by using the $FIR$ as a reward mechanism and $\epsilon - greedy$ as the policy, it was possible to obtain a balance between exploitation and exploration.

Lastly, we used the Q-model, an Artificial Network model, to estimate the Q-values in the $\epsilon - greedy$ Selection Policy. The weights of the network were updated using the Equation 5.4 according to the Q-Learning algorithm of Watkins and Dayan (1992).

$$T_{t+1} = R_{t+1} + \gamma max_a Q(s_{t+1}, a) \tag{5.4}$$

where $s_{t+1}$ is the next state after performing an action, $max_a Q(s_{t+1}, a)$ is the highest Q-value of all possible actions from state $s_{t+1}$. The discount factor $\gamma([0, 1])$ controls the influence of future estimated rewards.

## 5.3 ENSEMBLE AND POPULATION MANAGEMENT

When it comes to ensemble, there are three approaches to population management: competitive Single Population, competitive multi-population, and cooperative multi-population. GREEVO

utilizes the cooperative multi-population strategy since HGSADC has its own population. The **Population Management** of GREEVO stores the top 100 solutions in the main population entity. Before the **Q-Learning module** selects the next approach to continue the evolutionary process, the **Population Management** merges the main population with the retrieved solution (in case the approach retrieves only one solution) or with the final population of the previously selected approach. Finally, it keeps the 100 best solutions.

GREEVO's ensemble module relies on three techniques: HGSADC, FILO, and SISR. Each of them was described in the following Sections 3.5, 3.6, and 3.7 respectively. These techniques were chosen according to the subsequent specifications:

- Being state-of-the-art approach. Based on the premise of machine learning: Garbage In - Garbage Out.
- To have open-source code or a clear article with the components of the technique clearly and thoroughly explained.

Moving forward, each technique is selected by the **Q-Learning module**, and at the end of its process, receives a reward. Based on this reward, each technique will increase or decrease the chances of being selected in a future iteration.

## 5.4 CONCLUDING REMARKS

The Capacitated Vehicle Routing Problem (CVRP) remains a challenging optimization problem due to its NP-hard nature, especially when dealing with large instances. The No Free Lunch theorem underscores the inherent trade-offs among different algorithms, emphasizing the need for adaptive approaches to tackle diverse optimization problem characteristics.

GREEVO introduces a novel solution strategy for the CVRP. Combining elements of GRASP, ensemble techniques, and a Q-learning hyper-heuristic, aiming to adapt to the evolving search landscape during the optimization process.

The GRASP initialization module generates feasible solutions by employing a restricted candidate list and a randomized selection mechanism. The subsequent optimization using the Split Algorithm further refines the solutions, ensuring adherence to constraints and improving overall quality, and proved to be a very promising initialization (Do Rego and Pozo, 2021).

The Q-learning hyper-heuristic module introduces adaptability to the optimization process. The ensemble of baseline approaches managed cooperatively, allows GREEVO to leverage the strengths of individual algorithms, enhancing overall performance.

The incorporation of adaptive Q-learning and ensemble techniques in GREEVO contributes to its ability to handle a wide range of instances. While no algorithm can claim superiority across all scenarios, GREEVO presents a promising solution for the CVRP, offering a balance between exploration and exploitation.

In summary, GREEVO presents a robust and adaptive framework for addressing the CVRP, showcasing the potential of combining high-level heuristics in an ensemble and hyper-heuristic techniques for effective optimization in complex routing scenarios.

# 6 EXPERIMENTS

To assess the efficiency of GREEVO, we conducted a series of experiments comparing it to established approaches: HGSADC, SISR, and FILO. We were given access to the authors' implementation of HGSADC and FILO. In the FILO implementation, we had to modify it to handle duration constraints. We implemented the SISR based on (Christiaens and Berghe, 2020) since the original implementation was unavailable, but the results did not match the original.

To ensure the integrity of our results, we ran each benchmark instance 30 times with a 10-minute time limit. Our benchmark set, introduced by Golden et al. (1998), comprised 20 instances with customer counts ranging from 200 to 483, divided into two categories: those with duration constraints (Instances 1-8) and those without (Instances 9-20). Notably, all instances in this set maintained geometric symmetry.

We also incorporated Uchoa et al. (2017)'s dataset, encompassing 100 CVRP instances with vertex sizes spanning from 100 to 1000.

Table 6.1: Experiment Configuration Set

| **Parameter** | **Value** |
|---|---|
| **GRASP** *pop* size | 100 |
| **HGSADC** *max iter* without improvement | 20000 |
| **FILO** *max iter* without improvement | 20000 |
| **SISR** *max iter* without improvement | 3000 |
| **Q-Learning** Epsilon ($\epsilon$) | 0.05 |
| **Q-Learning** Reward | Raw Improvement Penalty (RIP) |
| **Q-Learning** Acceptance Criteria | ALL |
| **Q-Learning** State | Fitness Improvement Rate and Elapsed Time |

Table 6.1 displays the configuration set used in the experiments. The GRASP $pop$ size was defined considering the HGSADC initialization method, which has a $pop$ size 25 and multiplies it by 4 before triggering the survival selection, resulting in a temporary $pop$ size of 100.

All tested approaches (HGSADC, FILO, SISR, and GREEVO) were allocated a 10-minute time limit per dataset instance and run 30 times each. Moreover, we empirically defined stop criteria iterations for HGSADC 20000, FILO 20000, and SISR 3000 iterations without improvements. The values were derived through empirical analysis, taking into account those reported by the authors.

Meanwhile, for the Q-learning HH, we utilized an epsilon ($\epsilon$) value of 0.5 for exploration, $RIP$ as the reward function, $ALL$ as the acceptance criteria, and $FIR$ and $Elapsed\ Time$ as the state. The values were obtained from the authors.

To evaluate GREEVO's performance, we conducted various statistical tests. Our analysis included the Friedman hypothesis test and a pairwise post-hoc test with the Bergmann correction. We also utilized the Kruskall-Wallis and post-hoc Dunn test with Bonferroni correction. To further understand the statistical difference, we conducted a Cliff's Delta effect size test, comparing the results of both the Friedman and Kruskall-Wallis tests.

The GREEVO was implemented in two different languages: `Python` 3.11 and `C/C++` (compiled with g++ 12.3.0). To connect `Python` with `C/C++`, we used the `Pybind11` library.

The experiments were run in an Intel Xeon E5-2699 v3 (72) with a 3.6GHz clock and 128GB RAM.

## 6.1 STATISTICAL ANALYSIS

The Friedman test is a statistical tool that identifies dissimilarities in treatments across multiple trials. It is utilized when the assumptions of one-way ANOVA with repeated measures cannot be satisfied. In contrast to comparing means, the Friedman test compares the median of the data.

After establishing that there are significant differences between groups through the Friedman test, post-hoc pairwise comparisons are frequently carried out to identify which specific groups are distinct from each other. To manage the familywise error rate, several post-hoc correction methods have been developed. The Bergmann-Hommel correction is a prominent example.

Table 6.2: Friedman test with post-hoc using Bergmann-Hommel correction

| | **SISR** | **FILO** | **HGSADC** | **GREEVO** |
|---|---|---|---|---|
| **SISR** | NA | 0 | 0 | 0 |
| **FILO** | | NA | 1.768346e-04 | 4.234828e-04 |
| **HGSADC** | | | NA | 1.039169e-12 |
| **GREEVO** | | | | NA |

On the one hand, considering the Friedman test with post-hoc using Bergmann-Hommel correction (Friedman, 1937; Bergmann and Hommel, 1988), as shown by Table 6.2, the obtained $p$-values, show a notable distinction between GREEVO and other approaches at conventional significance levels, such as 0.05. The null hypothesis is declined for all mentioned hypotheses, indicating a statistically significant difference between the approaches.

Table 6.3: Kruskal-Wallis using post-hoc Dunn's Test with Bonferroni Correction

| | **FILO** | **GREEVO** | **HGSADC** | **SISR** |
|---|---|---|---|---|
| **FILO** | NA | 0.020 | 1.000 | 7.9e-07 |
| **GREEVO** | | NA | 0.035 | 1.3e-15 |
| **HGSADC** | | | NA | 2.8e-07 |
| **SISR** | | | | NA |

On the other hand, a Kruskal-Wallis test with Dunn's Test and Bonferroni correction (Kruskal and Wallis, 1952; Dewey and Seneta, 2001) offers a distinct perspective as shown in Table 6.3. This non-parametric statistical test compares three or more unpaired groups to detect any significant differences between them. The test is an extension of the Mann-Whitney U test and is utilized when the assumptions of one-way ANOVA are not met due to non-normality or heteroscedasticity of the data. Post-hoc tests like Dunn's test are utilized to determine which groups vary from each other.

The Kruskal-Wallis outcomes indicate that the $p$-value is exceptionally small, signifying that at least two of the groups differ significantly in their medians.

After reviewing Table 6.3, it is apparent that FILO displays distinct performance compared to GREEVO and SISR, while no significant difference is observed between HGSADC and FILO. The data also indicates that GREEVO has notable differences in performance compared

to both HGSADC and SISR, and HGSADC and SISR exhibit significant differences as well. The low $p$-values provide solid evidence against the null hypothesis for those pairs, suggesting that these methods have distinct performance levels.

To assess the variance between two non-parametric distributions, we utilized Cliff's Delta, also known as $d$ or Cliff's dominance statistic. This metric enables us to comprehend the magnitude of the differences between the approaches by determining the probability of a randomly selected value from one distribution being bigger or smaller than one from the other. The outcome varies from 0 (representing identical distributions) to values closer to 1 or -1 (indicating more significant differences). This measure is highly correlated with the rank-biserial correlation and can be easily transformed to and from it.

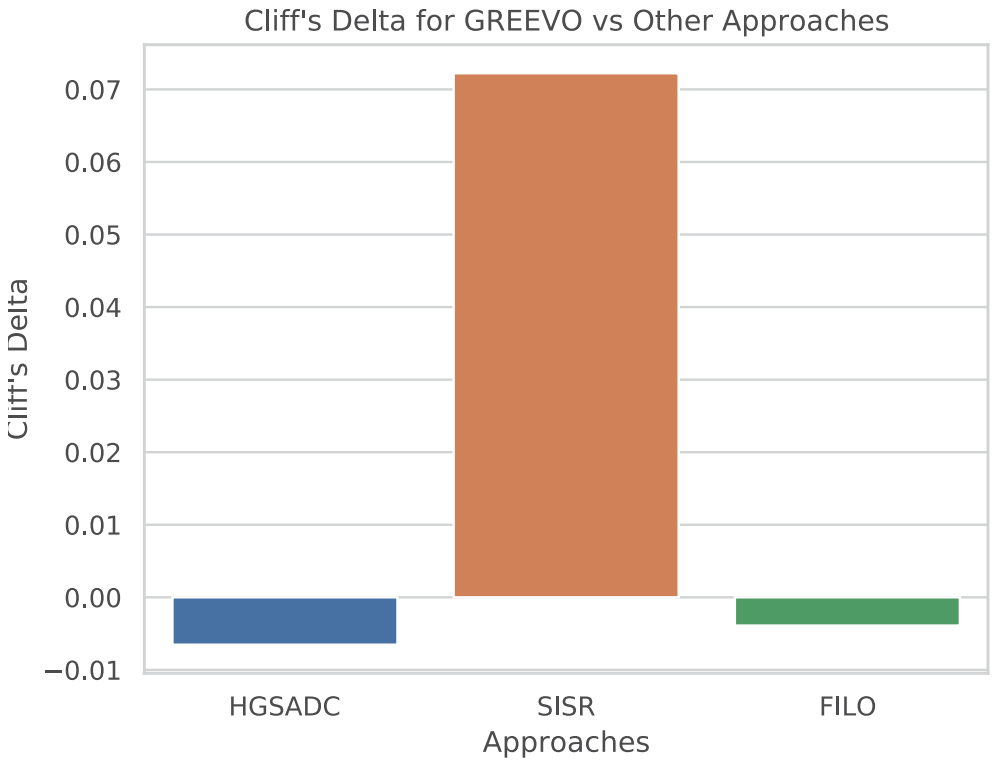



Figure 6.1: Friedman - Cliff's Delta Comparison

Figure 6.1 shows the comparison between SISR and GREEVO, FILO and GREEVO, and HGSADC and GREEVO, considering the Friedman test with post-hoc using Bergmann-Hommel correction. The Cliff's Delta value is 0.0722, indicating a small effect size but with a positive sign showing that GREEVO tends to perform slightly better than SISR. However, the practical significance of this effect size may be limited due to its small magnitude. For FILO vs. GREEVO, the negative sign of the Cliff's Delta value (-0.0039) suggested that FILO performed minimally better than GREEVO. However, the effect size was so small that it was essentially negligible. Lastly, in the comparison of HGSADC vs. GREEVO, the negative Cliff's Delta value (-0.0065) suggested that HGSADC performed slightly better than GREEVO, but the effect size was negligible as well.

Considering the Kruskall-Wallis results, the Cliff's Delta Comparison in Figure 6.2 shows GREEVO's overall performance compared to other algorithms, the examination of Cliff's Delta values offers valuable insights into the nature and magnitude of observed differences. When compared with HGSADC, Cliff's Delta (0.0373) reveals a small positive effect, suggesting

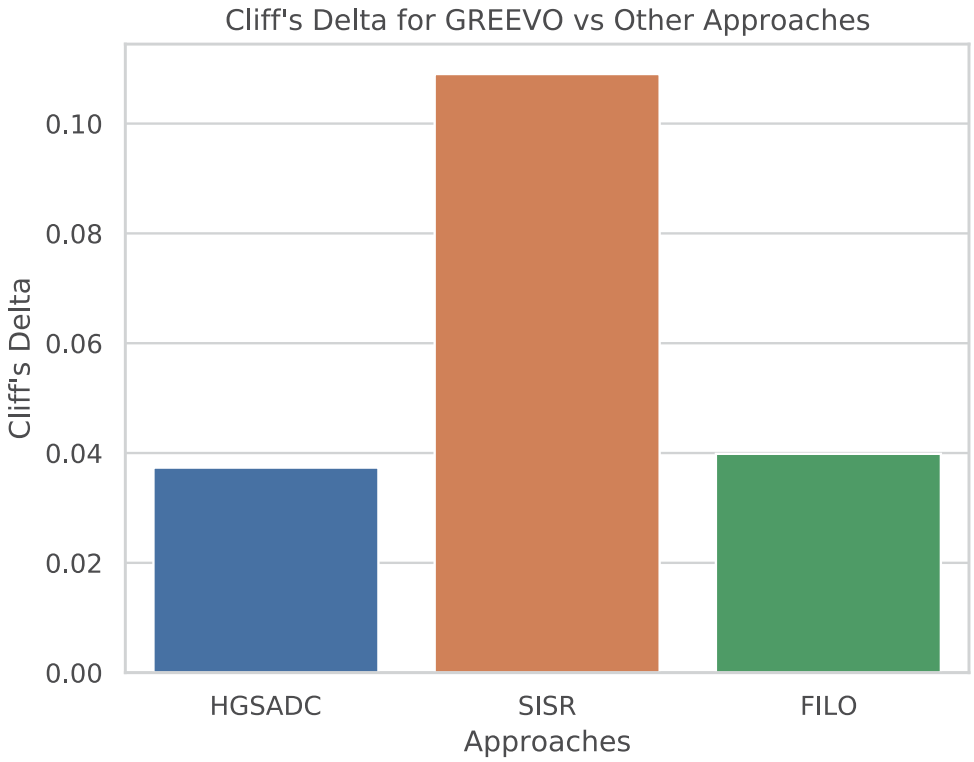



Figure 6.2: Kruskall-Wallis - Cliff's Delta Comparison

a general tendency for GREEVO to outperform HGSADC. Similarly, in the comparison with FILO, a positive value for Cliff's Delta (0.0399) indicates a small yet favorable association with GREEVO's performance. Notably, the comparison with SISR demonstrates a larger value for Cliff's Delta (0.1090), pointing to a stronger positive effect. This suggests that GREEVO consistently exhibits better overall performance when compared to SISR, with a more pronounced positive association. Across all comparisons, the positive associations support GREEVO, indicating its propensity for superior overall performance compared to HGSADC, SISR, and FILO.

## 6.2 OVERALL PERFORMANCE ANALYSIS

The statistical tests show that there is a difference between the approaches. Upon examining Figure 6.3, it becomes evident that the efficacy of GREEVO is well-founded. This visual representation offers a more comprehensive understanding of the distribution of data.

Although HGSADC and GREEVO share the same median, their respective data distributions can be quite disparate. GREEVO's distribution is more condensed, implying less variability and a lower upper quartile.

Still, in Figure 6.3, it illustrates that GREEVO outperforms HGSADC, boasting a lower upper quartile. This suggests that while half of the data (the median) may be comparable to HGSADC, GREEVO generally excels in the higher range of its data points (i.e., 50-75% of its data).

Table 6.4 shows various statistics related to the solutions. These statistics include the average (AVG), the difference between the average and the best-known solution (AVG GAP), the best solution found (BEST), and the difference between the best solution and the best-known

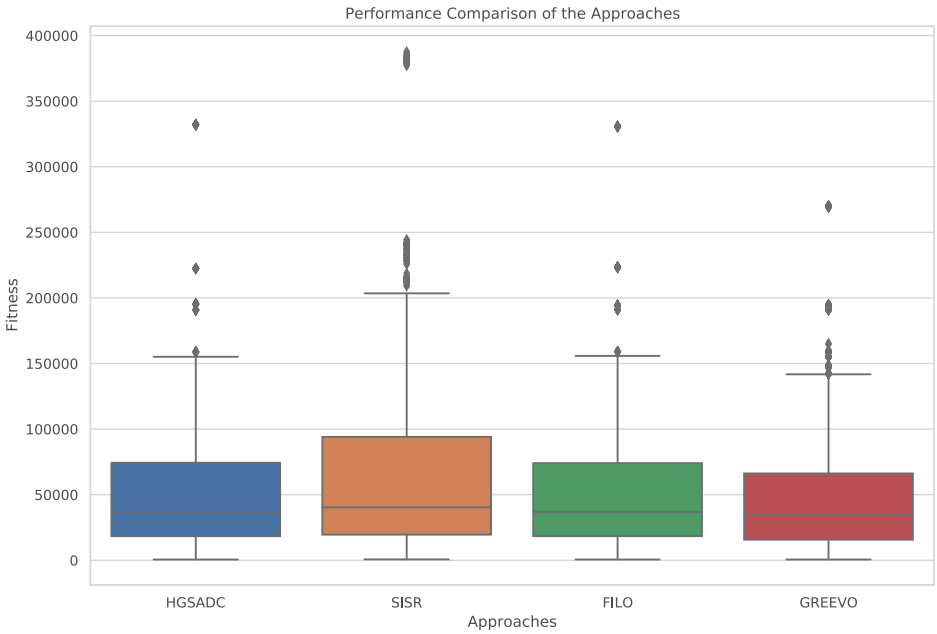



Figure 6.3: Overwall Performance Comparison of the Approaches

solution (BEST GAP). The bold values in the table are either equal to or lower than the best-known solution. The underlined values represent the lowest values among all the approaches. Lastly, the approach that performed the best statistically (determined using the Kruskal-Wallis test with post-hoc Dunn's test and Bonferroni correction) is highlighted in dark gray.

Based on the data presented in Table 6.4, it can be observed that HGSADC had the lowest AVG GAP (**0.25**) compared to the other methods assessed. However, GREEVO achieved the BEST GAP (**0.11**). HGSADC outperformed the others in **44** out of 120 cases, while GREEVO surpassed them in **29** out of 120 and FILO in **4** out of 120. In terms of the lowest values achieved, GREEVO had **61** out of 120, HGSADC had **60**, and FILO had **33**. Finally, concerning BKS, both GREEVO and HGSADC achieved **16** out of 120, while FILO attained **13**.

Table 6.4: Results of experiments with Golden et al. (Golden et al., 1998) and Uchoa et al. (Uchoa et al., 2017) CVRP benchmarks

| Instance | SISR | | | | FILO | | | | HGSADC | | | | GREEVO | | | | BKS |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | AVG | AVG GAP | BEST | BEST GAP | AVG | AVG GAP | BEST | BEST GAP | AVG | AVG GAP | BEST | BEST GAP | AVG | AVG GAP | BEST | BEST GAP | |
| **Golden_1** | 8366.96 | 48.79 | 7746.50 | 37.75 | 5636.97 | 0.24 | 5626.81 | 0.06 | 5647.60 | 0.43 | 5644.44 | 0.37 | 5647.37 | 0.42 | 5636.23 | 0.23 | 5623.47 |
| **Golden_2** | 12769.92 | 51.94 | 12056.00 | 43.45 | 8447.92 | 0.52 | 8447.92 | 0.52 | 8447.92 | 0.52 | 8447.92 | 0.52 | 8448.10 | 0.52 | 8447.92 | 0.52 | 8404.61 |
| **Golden_3** | 15149.51 | 37.75 | 13655.08 | 24.16 | 11036.22 | 0.35 | 11036.22 | 0.35 | 11044.26 | 0.42 | 11036.20 | 0.35 | 11037.34 | 0.36 | 11036.22 | 0.35 | 10997.80 |
| **Golden_4** | 16340.25 | 20.25 | 15744.07 | 15.86 | 13624.56 | 0.26 | 13624.52 | 0.26 | 13624.50 | 0.26 | 13624.50 | 0.26 | 13626.85 | 0.28 | 13624.52 | 0.26 | 13588.60 |
| **Golden_5** | 6844.31 | 5.93 | 6584.93 | 1.92 | 6460.98 | 0.00 | **6460.98** | 0.00 | 6460.98 | 0.00 | **6460.98** | 0.00 | 6460.98 | 0.00 | **6460.98** | 0.00 | 6460.98 |
| **Golden_6** | 9489.98 | 12.97 | 9362.62 | 11.46 | 8412.90 | 0.15 | 8412.90 | 0.15 | 8412.57 | 0.15 | 8404.20 | 0.05 | 8413.23 | 0.15 | 8412.90 | 0.15 | 8400.33 |
| **Golden_7** | 12201.74 | 20.78 | 11791.18 | 16.71 | 10195.59 | 0.92 | 10195.59 | 0.92 | 10181.10 | 0.78 | 10132.90 | 0.30 | 10195.59 | 0.92 | 10195.59 | 0.92 | 10102.70 |
| **Golden_8** | 16956.88 | 45.74 | 15436.80 | 32.67 | 11828.78 | 1.66 | 11828.78 | 1.66 | 11726.77 | 0.79 | 11663.50 | 0.24 | 11824.67 | 1.63 | 11664.76 | 0.25 | 11635.30 |
| **Golden_9** | 732.24 | 26.31 | 696.27 | 20.11 | 581.21 | 0.26 | **579.71** | 0.00 | 580.59 | 0.15 | **579.70** | 0.00 | 581.57 | 0.32 | **579.70** | 0.00 | 579.70 |
| **Golden_10** | 929.08 | 26.33 | 890.08 | 21.03 | 738.93 | 0.48 | 737.21 | 0.24 | 738.14 | 0.37 | 736.81 | 0.19 | 738.90 | 0.47 | 736.14 | 0.10 | 735.43 |
| **Golden_11** | 1161.20 | 27.33 | 1127.05 | 23.58 | 914.98 | 0.33 | 913.82 | 0.20 | 914.45 | 0.27 | 913.20 | 0.13 | 916.23 | 0.47 | 913.89 | 0.21 | 911.98 |
| **Golden_12** | 1530.46 | 39.05 | 1470.00 | 33.55 | 1105.92 | 0.48 | 1103.29 | 0.24 | 1105.44 | 0.43 | 1102.52 | 0.17 | 1106.25 | 0.51 | 1102.69 | 0.18 | 1100.67 |
| **Golden_13** | 1204.73 | 40.54 | 1102.82 | 28.66 | 859.49 | 0.27 | **857.19** | 0.00 | 857.84 | 0.08 | **857.19** | 0.00 | 859.57 | 0.28 | **857.19** | 0.00 | 857.19 |
| **Golden_14** | 1581.82 | 46.39 | 1395.77 | 29.17 | 1081.11 | 0.05 | **1080.55** | 0.00 | 1080.55 | 0.00 | **1080.55** | 0.00 | 1081.58 | 0.10 | **1080.55** | 0.00 | 1080.55 |
| **Golden_15** | 1946.01 | 45.52 | 1720.10 | 28.63 | 1341.27 | 0.30 | 1339.29 | 0.15 | 1340.34 | 0.23 | 1338.45 | 0.09 | 1342.57 | 0.40 | 1338.82 | 0.12 | 1337.27 |
| **Golden_16** | 2304.69 | 43.03 | 2124.24 | 31.84 | 1617.33 | 0.38 | 1611.78 | 0.03 | 1616.36 | 0.32 | 1612.72 | 0.09 | 1621.09 | 0.61 | 1614.32 | 0.19 | 1611.28 |
| **Golden_17** | 930.65 | 31.49 | 879.51 | 24.27 | 707.83 | 0.01 | **707.76** | 0.00 | 707.76 | 0.00 | **707.76** | 0.00 | 707.93 | 0.02 | **707.76** | 0.00 | 707.76 |
| **Golden_18** | 1451.11 | 45.82 | 1358.90 | 36.56 | 997.09 | 0.20 | **995.13** | 0.00 | 995.46 | 0.03 | **995.13** | 0.00 | 997.24 | 0.21 | **995.13** | 0.00 | 995.13 |
| **Golden_19** | 1912.89 | 40.08 | 1793.02 | 31.30 | 1366.32 | 0.05 | **1365.60** | 0.00 | 1365.88 | 0.02 | **1365.60** | 0.00 | 1366.76 | 0.08 | **1365.60** | 0.00 | 1365.60 |
| **Golden_20** | 2615.54 | 43.90 | 2461.59 | 35.43 | 1819.46 | 0.10 | 1817.89 | 0.02 | 1818.95 | 0.07 | 1817.64 | 0.00 | 1820.00 | 0.13 | 1818.64 | 0.06 | 1817.59 |
| **X-n101-k25** | 28095.71 | 1.83 | 27598.10 | 0.03 | 27598.10 | 0.03 | 27598.10 | 0.03 | 27598.10 | 0.03 | 27598.10 | 0.03 | 27598.10 | 0.03 | 27598.10 | 0.03 | 27591 |
| **X-n106-k14** | 26746.26 | 1.46 | 26533.35 | 0.65 | 26372.05 | 0.04 | **26362.24** | 0.00 | 26376.80 | 0.06 | **26362.20** | 0.00 | 26378.47 | 0.06 | **26362.24** | 0.00 | 26362 |
| **X-n110-k13** | 15197.13 | 1.51 | 15059.27 | 0.59 | 14978.22 | 0.05 | 14978.22 | 0.05 | 14978.20 | 0.05 | 14978.20 | 0.05 | 14978.22 | 0.05 | 14978.22 | 0.05 | 14971 |
| **X-n115-k10** | 12900.45 | 1.20 | 12751.64 | 0.04 | 12751.64 | 0.04 | 12751.64 | 0.04 | 12751.60 | 0.04 | 12751.60 | 0.04 | 12751.64 | 0.04 | 12751.64 | 0.04 | 12747 |
| **X-n120-k6** | 13795.12 | 3.47 | 13716.47 | 2.88 | 13329.42 | -0.02 | **13329.42** | -0.02 | 13329.40 | -0.02 | **13329.40** | -0.02 | 13329.43 | -0.02 | **13329.42** | -0.02 | 13332 |
| **X-n125-k30** | 57203.46 | 3.00 | 56386.37 | 1.53 | 55702.30 | 0.29 | 55546.20 | 0.01 | 55546.20 | 0.01 | 55546.20 | 0.01 | 55648.96 | 0.20 | 55546.20 | 0.01 | 55539 |
| **X-n129-k18** | 29768.87 | 2.86 | 29091.26 | 0.52 | 28955.11 | 0.05 | 28946.53 | 0.02 | 28946.50 | 0.02 | 28946.50 | 0.02 | 28950.80 | 0.04 | 28946.53 | 0.02 | 28940 |
| **X-n134-k13** | 11545.40 | 5.77 | 11234.19 | 2.91 | 10936.04 | 0.18 | 10919.85 | 0.04 | 10919.80 | 0.03 | 10919.80 | 0.03 | 10928.02 | 0.11 | 10919.85 | 0.04 | 10916 |
| **X-n139-k10** | 13902.55 | 2.30 | 13727.62 | 1.01 | 13595.65 | 0.04 | 13595.65 | 0.04 | 13595.70 | 0.04 | 13595.70 | 0.04 | 13595.65 | 0.04 | 13595.65 | 0.04 | 13590 |
| **X-n143-k7** | 16491.19 | 5.04 | 16322.28 | 3.96 | 15722.14 | 0.14 | **15697.06** | -0.02 | 15697.10 | -0.02 | **15697.10** | -0.02 | 15713.42 | 0.09 | **15697.06** | -0.02 | 15700 |
| **X-n148-k46** | 44623.44 | 2.71 | 44082.26 | 1.46 | 43455.34 | 0.02 | 43452.95 | 0.01 | 43452.90 | 0.01 | 43452.90 | 0.01 | 43487.05 | 0.09 | 43452.95 | 0.01 | 43448 |
| **X-n153-k22** | 22250.22 | 4.85 | 21841.66 | 2.93 | 21259.00 | 0.18 | 21227.04 | 0.03 | 21227.00 | 0.03 | 21227.00 | 0.03 | 21242.24 | 0.10 | 21227.04 | 0.03 | 21220 |
| **X-n157-k13** | 17250.76 | 2.22 | 16982.72 | 0.63 | 16885.13 | 0.05 | 16885.13 | 0.05 | 16885.10 | 0.05 | 16885.10 | 0.05 | 16885.21 | 0.05 | 16885.13 | 0.05 | 16876 |
| **X-n162-k11** | 14534.88 | 2.81 | 14296.24 | 1.12 | 14165.62 | 0.20 | 14147.71 | 0.07 | 14138.60 | 0.00 | **14138.60** | 0.00 | 14152.03 | 0.10 | **14138.58** | 0.00 | 14138 |
| **X-n167-k10** | 21752.24 | 5.81 | 21176.97 | 3.02 | 20557.38 | 0.00 | **20557.38** | 0.00 | 20557.40 | 0.00 | **20557.40** | 0.00 | 20560.51 | 0.02 | **20557.38** | 0.00 | 20557 |
| **X-n172-k51** | 48061.80 | 5.38 | 46808.23 | 2.63 | 45612.06 | 0.01 | 45612.06 | 0.01 | 45612.10 | 0.01 | 45612.10 | 0.01 | 45711.55 | 0.23 | 45612.06 | 0.01 | 45607 |
| **X-n176-k26** | 50019.61 | 4.62 | 49273.39 | 3.06 | 47943.20 | 0.27 | **47812.19** | 0.00 | 47812.20 | 0.00 | **47812.20** | 0.00 | 47857.88 | 0.10 | **47812.19** | 0.00 | 47812 |
| **X-n181-k23** | 26816.13 | 4.88 | 26311.36 | 2.90 | 25576.25 | 0.03 | 25575.68 | 0.03 | 25575.70 | 0.03 | 25575.70 | 0.03 | 25577.57 | 0.03 | 25575.68 | 0.03 | 25569 |
| **X-n186-k15** | 25585.90 | 5.97 | 24746.35 | 2.49 | 24162.14 | 0.07 | 24155.51 | 0.04 | 24154.30 | 0.04 | 24154.30 | 0.04 | 24164.84 | 0.08 | 24154.29 | 0.04 | 24145 |
| **X-n190-k8** | 18059.07 | 6.35 | 17732.88 | 4.43 | 16993.81 | 0.08 | 16984.50 | 0.03 | 16991.58 | 0.07 | 16984.50 | 0.03 | 17058.70 | 0.46 | 16985.90 | 0.03 | 16980 |
| **X-n195-k51** | 46411.10 | 4.94 | 45611.26 | 3.13 | 44272.84 | 0.11 | 44229.31 | 0.01 | 44230.86 | 0.01 | 44229.30 | 0.01 | 44277.81 | 0.12 | 44229.31 | 0.01 | 44225 |

| | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **X-n200-k36** | 61832.53 | 5.56 | 61009.49 | 4.15 | 58827.73 | 0.43 | 58630.68 | 0.09 | 58582.60 | 0.01 | 58581.50 | 0.01 | 58720.16 | 0.24 | 58581.48 | 0.01 | 58578 |
| **X-n204-k19** | 20631.29 | 5.45 | 20248.02 | 3.49 | 19572.48 | 0.04 | 19567.69 | 0.01 | 19567.70 | 0.01 | 19567.70 | 0.01 | 19606.89 | 0.21 | 19567.69 | 0.01 | 19565 |
| **X-n209-k16** | 32658.92 | 6.53 | 31963.43 | 4.26 | 30692.89 | 0.12 | 30664.35 | 0.03 | 30659.90 | 0.01 | 30659.70 | 0.01 | 30688.15 | 0.10 | 30659.66 | 0.01 | 30656 |
| **X-n214-k11** | 12147.19 | 11.89 | 11913.72 | 9.74 | 10892.41 | 0.34 | 10871.36 | 0.14 | 10870.03 | 0.13 | 10864.40 | 0.08 | 10892.89 | 0.34 | 10866.82 | 0.10 | 10856 |
| **X-n219-k73** | 120182.84 | 2.20 | 119254.77 | 1.41 | 117601.49 | 0.01 | 117601.10 | 0.01 | 117602.16 | 0.01 | 117601.00 | 0.01 | 117622.60 | 0.02 | 117601.10 | 0.01 | 117595 |
| **X-n223-k34** | 43071.87 | 6.52 | 41952.49 | 3.75 | 40512.64 | 0.19 | 40446.28 | 0.02 | 40437.90 | 0.00 | **40437.90** | 0.00 | 40517.71 | 0.20 | **40437.95** | 0.00 | 40437 |
| **X-n228-k23** | 27514.40 | 6.89 | 26863.29 | 4.36 | 25805.45 | 0.25 | 25753.22 | 0.04 | 25746.32 | 0.02 | 25745.60 | 0.01 | 25769.87 | 0.11 | 25745.58 | 0.01 | 25742 |
| **X-n233-k16** | 20926.21 | 8.82 | 20449.67 | 6.34 | 19318.21 | 0.46 | 19239.08 | 0.05 | 19239.83 | 0.05 | 19239.10 | 0.05 | 19306.02 | 0.40 | 19239.08 | 0.05 | 19230 |
| **X-n237-k14** | 28919.67 | 6.94 | 28189.49 | 4.24 | 27053.65 | 0.04 | 27049.01 | 0.03 | 27049.12 | 0.03 | 27049.00 | 0.03 | 27059.49 | 0.06 | 27049.01 | 0.03 | 27042 |
| **X-n242-k48** | 87218.49 | 5.40 | 85562.39 | 3.40 | 82909.72 | 0.19 | 82810.04 | 0.07 | 82815.64 | 0.08 | 82757.20 | 0.01 | 82969.48 | 0.26 | 82785.40 | 0.04 | 82751 |
| **X-n247-k50** | 39472.95 | 5.90 | 38514.16 | 3.33 | 37480.90 | 0.56 | 37279.79 | 0.02 | 37294.15 | 0.05 | 37279.80 | 0.02 | 37477.27 | 0.55 | 37279.79 | 0.02 | 37274 |
| **X-n251-k28** | 41714.59 | 7.83 | 40718.96 | 5.26 | 38792.72 | 0.28 | 38693.33 | 0.02 | 38700.94 | 0.04 | 38690.40 | 0.02 | 38784.20 | 0.26 | 38690.43 | 0.02 | 38684 |
| **X-n256-k16** | 19904.01 | 5.65 | 19554.73 | 3.80 | 18887.44 | 0.26 | 18887.44 | 0.26 | 18847.61 | 0.05 | 18846.40 | 0.04 | 18900.94 | 0.33 | 18846.39 | 0.04 | 18839 |
| **X-n261-k13** | 29248.62 | 10.13 | 28814.62 | 8.50 | 26637.25 | 0.30 | 26570.95 | 0.05 | 26577.32 | 0.07 | 26564.90 | 0.03 | 26716.91 | 0.60 | 26564.88 | 0.03 | 26558 |
| **X-n266-k58** | 80696.93 | 6.91 | 79276.24 | 5.03 | 75793.98 | 0.42 | 75562.92 | 0.11 | 75615.36 | 0.18 | 75534.30 | 0.07 | 75804.68 | 0.43 | 75534.28 | 0.07 | 75478 |
| **X-n270-k35** | 37944.71 | 7.52 | 36910.27 | 4.59 | 35364.79 | 0.21 | 35304.38 | 0.04 | 35301.47 | 0.03 | 35292.90 | 0.01 | 35360.21 | 0.20 | 35300.78 | 0.03 | 35291 |
| **X-n275-k28** | 23595.51 | 11.06 | 23143.15 | 8.93 | 21254.56 | 0.05 | **21245.20** | 0.00 | 21245.20 | 0.00 | **21245.20** | 0.00 | 21268.75 | 0.11 | **21245.20** | 0.00 | 21245 |
| **X-n280-k17** | 36637.87 | 9.36 | 35859.59 | 7.03 | 33651.69 | 0.44 | 33544.95 | 0.13 | 33543.98 | 0.12 | **33503.50** | 0.00 | 33600.43 | 0.29 | **33503.46** | 0.00 | 33503 |
| **X-n284-k15** | 22637.93 | 11.99 | 22184.74 | 9.74 | 20278.68 | 0.32 | 20253.24 | 0.19 | 20262.28 | 0.23 | 20235.10 | 0.10 | 20335.93 | 0.60 | 20248.90 | 0.17 | 20215 |
| **X-n289-k60** | 105161.37 | 10.52 | 101672.56 | 6.85 | 95580.61 | 0.45 | 95422.64 | 0.29 | 95323.51 | 0.18 | 95243.20 | 0.10 | 95526.36 | 0.39 | 95199.16 | 0.05 | 95151 |
| **X-n294-k50** | 51951.52 | 10.16 | 50469.86 | 7.02 | 47297.49 | 0.29 | 47229.74 | 0.15 | 47205.63 | 0.09 | 47172.30 | 0.02 | 47353.34 | 0.41 | 47206.43 | 0.10 | 47161 |
| **X-n298-k31** | 38931.70 | 13.73 | 37199.85 | 8.67 | 34315.85 | 0.25 | 34244.55 | 0.04 | 34254.00 | 0.07 | 34243.50 | 0.04 | 34312.73 | 0.24 | 34243.50 | 0.04 | 34231 |
| **X-n303-k21** | 24087.07 | 10.82 | 23700.58 | 9.04 | 21832.35 | 0.44 | 21777.26 | 0.19 | 21762.77 | 0.12 | 21753.70 | 0.08 | 21831.19 | 0.44 | 21757.68 | 0.10 | 21736 |
| **X-n308-k13** | 28587.85 | 10.55 | 27793.69 | 7.48 | 26135.17 | 1.07 | 25889.99 | 0.12 | 25901.95 | 0.17 | 25875.20 | 0.06 | 26016.78 | 0.61 | 25875.16 | 0.06 | 25859 |
| **X-n313-k71** | 101750.89 | 8.20 | 100394.15 | 6.75 | 94379.53 | 0.36 | 94240.89 | 0.21 | 94135.53 | 0.10 | 94055.40 | 0.01 | 94394.31 | 0.37 | 94095.73 | 0.06 | 94043 |
| **X-n317-k53** | 83136.22 | 6.10 | 82130.68 | 4.82 | 78377.73 | 0.03 | 78372.77 | 0.02 | 78378.51 | 0.03 | 78372.80 | 0.02 | 78381.86 | 0.03 | 78372.77 | 0.02 | 78355 |
| **X-n322-k28** | 34389.86 | 15.27 | 32791.90 | 9.91 | 29960.09 | 0.42 | 29893.11 | 0.20 | 29870.34 | 0.12 | 29845.30 | 0.04 | 29966.20 | 0.44 | 29867.20 | 0.11 | 29834 |
| **X-n327-k20** | 31100.14 | 12.96 | 30235.81 | 9.82 | 27672.16 | 0.51 | 27592.12 | 0.22 | 27565.96 | 0.12 | 27541.80 | 0.04 | 27661.56 | 0.47 | 27574.51 | 0.15 | 27532 |
| **X-n331-k15** | 34135.18 | 9.75 | 33816.55 | 8.73 | 31129.33 | 0.09 | 31122.67 | 0.07 | 31123.44 | 0.07 | 31122.70 | 0.07 | 31127.95 | 0.08 | 31122.67 | 0.07 | 31102 |
| **X-n336-k84** | 153555.80 | 10.38 | 148914.81 | 7.05 | 139633.48 | 0.38 | 139271.81 | 0.12 | 139381.00 | 0.19 | 139199.00 | 0.06 | 140032.52 | 0.66 | 139229.22 | 0.08 | 139111 |
| **X-n344-k43** | 46701.75 | 11.06 | 45621.95 | 8.49 | 42227.63 | 0.42 | 42072.38 | 0.05 | 42098.68 | 0.12 | 42067.50 | 0.04 | 42235.08 | 0.44 | 42068.15 | 0.04 | 42050 |
| **X-n351-k40** | 31076.26 | 20.00 | 30388.71 | 17.35 | 26043.27 | 0.57 | 25979.69 | 0.32 | 25960.35 | 0.25 | 25938.30 | 0.16 | 26043.65 | 0.57 | 25956.08 | 0.23 | 25896 |
| **X-n359-k29** | 58436.24 | 13.46 | 57213.91 | 11.08 | 51672.05 | 0.32 | 51523.79 | 0.04 | 51685.58 | 0.35 | 51576.20 | 0.14 | 51773.52 | 0.52 | 51592.19 | 0.17 | 51505 |
| **X-n367-k17** | 25471.09 | 11.65 | 24657.94 | 8.08 | 22852.73 | 0.17 | 22833.66 | 0.09 | 22833.70 | 0.09 | 22833.70 | 0.09 | 22868.34 | 0.24 | 22833.66 | 0.09 | 22814 |
| **X-n376-k94** | 154803.26 | 4.80 | 153317.72 | 3.79 | 147740.60 | 0.02 | 147729.45 | 0.01 | 147731.29 | 0.01 | 147729.00 | 0.01 | 147744.75 | 0.02 | 147729.45 | 0.01 | 147713 |
| **X-n384-k52** | 74543.75 | 13.07 | 73012.01 | 10.75 | 66179.49 | 0.38 | 66057.78 | 0.20 | 66085.43 | 0.24 | 66019.50 | 0.14 | 66180.53 | 0.38 | 66013.31 | 0.13 | 65928 |
| **X-n393-k38** | 44100.38 | 15.26 | 42412.81 | 10.85 | 38332.66 | 0.19 | 38295.41 | 0.09 | 38272.82 | 0.03 | 38267.30 | 0.02 | 38366.38 | 0.28 | 38269.55 | 0.02 | 38260 |
| **X-n401-k29** | 75133.38 | 13.57 | 73489.43 | 11.09 | 66302.61 | 0.22 | 66262.39 | 0.16 | 66272.08 | 0.18 | 66216.80 | 0.09 | 66351.58 | 0.30 | 66228.78 | 0.11 | 66154 |
| **X-n411-k19** | 22584.41 | 14.57 | 21824.97 | 10.72 | 19856.55 | 0.73 | 19785.43 | 0.37 | 19746.95 | 0.18 | 19738.50 | 0.13 | 19907.03 | 0.99 | 19728.83 | 0.09 | 19712 |
| **X-n420-k130** | 116650.59 | 8.21 | 114253.47 | 5.99 | 107969.45 | 0.16 | 107851.97 | 0.05 | 107861.00 | 0.06 | 107823.00 | 0.02 | 108139.34 | 0.32 | 107822.99 | 0.02 | 107798 |
| **X-n429-k61** | 74025.42 | 13.10 | 72161.11 | 10.26 | 65643.86 | 0.30 | 65556.83 | 0.16 | 65530.12 | 0.12 | 65490.20 | 0.06 | 65652.43 | 0.31 | 65481.36 | 0.05 | 65449 |
| **X-n439-k37** | 41089.35 | 12.91 | 39916.52 | 9.69 | 36426.72 | 0.10 | 36408.64 | 0.05 | 36419.19 | 0.08 | 36408.60 | 0.05 | 36442.41 | 0.14 | 36408.64 | 0.05 | 36391 |
| **X-n449-k29** | 66769.71 | 20.89 | 62851.37 | 13.79 | 55507.59 | 0.50 | 55398.93 | 0.30 | 55434.94 | 0.37 | 55316.40 | 0.15 | 55786.15 | 1.00 | 55328.05 | 0.17 | 55233 |
| **X-n459-k26** | 30778.19 | 27.50 | 29495.78 | 22.19 | 24233.28 | 0.39 | 24195.40 | 0.23 | 24191.49 | 0.22 | 24163.00 | 0.10 | 24239.77 | 0.42 | 24181.81 | 0.18 | 24139 |
| **X-n469-k138** | 240784.39 | 8.55 | 237291.73 | 6.97 | 223373.95 | 0.70 | 222953.38 | 0.51 | 222391.74 | 0.26 | 222200.00 | 0.17 | 222804.38 | 0.44 | 222002.24 | 0.08 | 221824 |
| **X-n480-k70** | 99815.49 | 11.59 | 97673.18 | 9.19 | 89725.76 | 0.31 | 89581.63 | 0.15 | 89605.15 | 0.17 | 89500.30 | 0.06 | 89784.08 | 0.37 | 89489.89 | 0.05 | 89449 |

| | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **X-n491-k59** | 83470.62 | 25.55 | 77955.07 | 17.26 | 66772.12 | 0.43 | 66691.48 | 0.31 | 66715.36 | 0.35 | 66599.60 | 0.18 | 67088.37 | 0.91 | 66687.55 | 0.31 | 66483 |
| **X-n502-k39** | 73813.30 | 6.63 | 73239.49 | 5.80 | 69285.20 | 0.09 | 69256.63 | 0.04 | 69280.65 | 0.08 | 69259.90 | 0.05 | 69284.19 | 0.08 | 69260.20 | 0.05 | 69226 |
| **X-n513-k21** | 28375.25 | 17.25 | 27216.19 | 12.46 | 24321.80 | 0.50 | 24267.82 | 0.28 | 24231.46 | 0.13 | 24224.40 | 0.10 | 24352.10 | 0.62 | 24224.37 | 0.10 | 24201 |
| **X-n524-k153** | 163478.39 | 5.75 | 161441.03 | 4.43 | 155159.85 | 0.37 | 154684.86 | 0.06 | 154928.87 | 0.22 | 154771.00 | 0.12 | 155242.54 | 0.42 | 154637.60 | 0.03 | 154593 |
| **X-n536-k96** | 111908.47 | 17.99 | 108852.79 | 14.77 | 95578.49 | 0.77 | 95418.87 | 0.60 | 95168.40 | 0.34 | 95043.70 | 0.21 | 95450.43 | 0.64 | 95026.56 | 0.19 | 94846 |
| **X-n548-k50** | 98688.92 | 13.83 | 96389.99 | 11.18 | 86785.33 | 0.10 | 86726.88 | 0.03 | 86829.12 | 0.15 | 86733.80 | 0.04 | 86818.05 | 0.14 | 86722.71 | 0.03 | 86700 |
| **X-n561-k42** | 50603.82 | 18.46 | 49303.70 | 15.42 | 42887.03 | 0.40 | 42804.54 | 0.20 | 42788.50 | 0.17 | 42747.80 | 0.07 | 42935.33 | 0.51 | 42762.91 | 0.11 | 42717 |
| **X-n573-k30** | 55207.03 | 8.95 | 54194.23 | 6.95 | 50890.81 | 0.43 | 50802.32 | 0.26 | 50899.18 | 0.45 | 50831.50 | 0.31 | 50929.97 | 0.51 | 50783.27 | 0.22 | 50673 |
| **X-n586-k159** | 213839.20 | 12.36 | 209108.68 | 9.87 | 191192.34 | 0.46 | 190899.00 | 0.31 | 190820.48 | 0.27 | 190430.00 | 0.06 | 191222.00 | 0.48 | 190608.80 | 0.15 | 190316 |
| **X-n599-k92** | 127872.98 | 17.91 | 125163.44 | 15.41 | 108830.70 | 0.35 | 108698.83 | 0.23 | 108808.61 | 0.33 | 108662.00 | 0.19 | 108911.97 | 0.43 | 108606.95 | 0.14 | 108451 |
| **X-n613-k62** | 79272.75 | 33.15 | 76390.55 | 28.31 | 59820.43 | 0.48 | 59653.66 | 0.20 | 59785.56 | 0.42 | 59591.40 | 0.09 | 59936.62 | 0.67 | 59637.30 | 0.17 | 59535 |
| **X-n627-k43** | 76704.00 | 23.39 | 73563.71 | 18.34 | 62384.27 | 0.35 | 62312.63 | 0.24 | 62600.62 | 0.70 | 62463.10 | 0.48 | 62474.42 | 0.50 | 62284.57 | 0.19 | 62164 |
| **X-n641-k35** | 75960.99 | 19.28 | 74127.39 | 16.40 | 63909.55 | 0.36 | 63810.66 | 0.20 | 64065.41 | 0.60 | 63893.40 | 0.33 | 64075.37 | 0.62 | 63796.70 | 0.18 | 63682 |
| **X-n655-k131** | 119688.41 | 12.09 | 115744.67 | 8.40 | 106841.12 | 0.06 | 106821.28 | 0.04 | 106856.94 | 0.07 | 106820.00 | 0.04 | 106858.83 | 0.07 | 106831.91 | 0.05 | 106780 |
| **X-n670-k130** | 163663.57 | 11.84 | 159791.03 | 9.20 | 147928.81 | 1.09 | 147274.86 | 0.64 | 147089.29 | 0.52 | 146692.00 | 0.25 | 147611.88 | 0.87 | 146725.77 | 0.27 | 146332 |
| **X-n685-k75** | 98957.39 | 45.09 | 84741.98 | 24.25 | 68564.02 | 0.53 | 68382.22 | 0.26 | 68508.27 | 0.44 | 68366.00 | 0.24 | 68599.90 | 0.58 | 68340.81 | 0.20 | 68205 |
| **X-n701-k44** | 98740.87 | 20.53 | 96712.75 | 18.05 | 82272.44 | 0.43 | 82047.02 | 0.15 | 82606.98 | 0.83 | 82335.10 | 0.50 | 82495.16 | 0.70 | 82239.45 | 0.39 | 81923 |
| **X-n716-k35** | 52844.26 | 21.84 | 51734.47 | 19.28 | 43625.75 | 0.58 | 43530.93 | 0.36 | 43662.49 | 0.67 | 43566.30 | 0.45 | 43943.52 | 1.32 | 43533.59 | 0.37 | 43373 |
| **X-n733-k159** | 161642.34 | 18.69 | 157973.11 | 16.00 | 136547.08 | 0.26 | 136376.86 | 0.14 | 136558.97 | 0.27 | 136409.00 | 0.16 | 136874.77 | 0.51 | 136362.09 | 0.13 | 136187 |
| **X-n749-k98** | 98596.05 | 27.60 | 96478.84 | 24.86 | 77724.93 | 0.59 | 77591.10 | 0.42 | 78033.83 | 0.99 | 77853.80 | 0.76 | 77869.14 | 0.78 | 77577.96 | 0.40 | 77269 |
| **X-n766-k71** | 140021.85 | 22.38 | 133515.83 | 16.69 | 115137.31 | 0.63 | 114872.42 | 0.40 | 114968.77 | 0.48 | 114836.00 | 0.37 | 115159.30 | 0.65 | 114684.13 | 0.23 | 114417 |
| **X-n783-k48** | 98532.97 | 36.12 | 95794.30 | 32.34 | 72737.77 | 0.49 | 72607.04 | 0.31 | 73238.21 | 1.18 | 72965.80 | 0.80 | 73195.86 | 1.12 | 72461.49 | 0.10 | 72386 |
| **X-n801-k40** | 92780.14 | 26.57 | 88569.86 | 20.82 | 73468.20 | 0.22 | 73387.21 | 0.11 | 73791.21 | 0.66 | 73558.50 | 0.35 | 73534.16 | 0.31 | 73399.00 | 0.13 | 73305 |
| **X-n819-k171** | 184709.49 | 16.82 | 181073.00 | 14.52 | 159080.51 | 0.61 | 158846.17 | 0.46 | 158844.29 | 0.46 | 158579.00 | 0.29 | 159026.16 | 0.57 | 158438.55 | 0.20 | 158121 |
| **X-n837-k142** | 232020.34 | 19.76 | 225605.28 | 16.45 | 194417.33 | 0.35 | 194179.69 | 0.23 | 195294.52 | 0.80 | 194842.00 | 0.57 | 194800.65 | 0.55 | 194122.08 | 0.20 | 193737 |
| **X-n856-k95** | 110196.80 | 23.87 | 105414.67 | 18.49 | 89118.72 | 0.17 | 89044.16 | 0.09 | 89135.83 | 0.19 | 89082.50 | 0.13 | 89188.66 | 0.25 | 89107.47 | 0.16 | 88965 |
| **X-n876-k59** | 120809.73 | 21.66 | 117901.76 | 18.73 | 99721.07 | 0.43 | 99611.07 | 0.31 | 100286.96 | 0.99 | 99972.70 | 0.68 | 99888.63 | 0.59 | 99675.51 | 0.38 | 99299 |
| **X-n895-k37** | 70276.08 | 30.48 | 66645.55 | 23.74 | 54189.53 | 0.61 | 54025.46 | 0.31 | 54437.51 | 1.07 | 54198.20 | 0.63 | 54312.47 | 0.84 | 54047.62 | 0.35 | 53860 |
| **X-n916-k207** | 381595.70 | 15.92 | 377083.76 | 14.55 | 330726.84 | 0.47 | 330183.72 | 0.31 | 332061.81 | 0.88 | 331615.00 | 0.74 | 331561.07 | 0.72 | 329719.59 | 0.16 | 329179 |
| **X-n936-k151** | 156852.11 | 18.19 | 149776.58 | 12.86 | 133761.07 | 0.79 | 133302.77 | 0.44 | 133686.10 | 0.73 | 133413.00 | 0.53 | 133923.16 | 0.91 | 133076.02 | 0.27 | 132715 |
| **X-n957-k87** | 109296.44 | 27.88 | 102382.00 | 19.79 | 85613.96 | 0.17 | 85574.19 | 0.13 | 85823.56 | 0.42 | 85619.30 | 0.18 | 85664.20 | 0.23 | 85564.87 | 0.12 | 85465 |
| **X-n979-k58** | 152377.86 | 28.07 | 146902.05 | 23.47 | 119368.17 | 0.33 | 119256.83 | 0.24 | 120179.45 | 1.01 | 119814.00 | 0.70 | 119724.52 | 0.63 | 119192.71 | 0.18 | 118976 |
| **X-n1001-k43** | 99916.63 | 38.09 | 93586.42 | 29.34 | 72829.85 | 0.66 | 72704.50 | 0.48 | 73533.63 | 1.63 | 73199.00 | 1.17 | 73041.33 | 0.95 | 72604.58 | 0.34 | 72355 |
| **AVG GAP AVG** | | 16.37 | | | | 0.32 | | | | **0.25** | | | | 0.38 | | | |
| **AVG GAP BEST** | | | | 12.39 | | | | 0.16 | | | | 0.14 | | | | **0.11** | |

## 6.3 CONVERGENCE AND DOMINANCE ANALYSIS

In this experiment, we aim to demonstrate how GREEVO compares to other approaches regarding convergence. We also want to ensure that the Q-learning HH algorithm was not biased towards a single technique. To achieve this, we conducted an independent run using instance X-n1001-43 from Uchoa et al. (2017). We tested each approach, including HGSADC, FILO, SISR, and GREEVO, for 15 minutes.

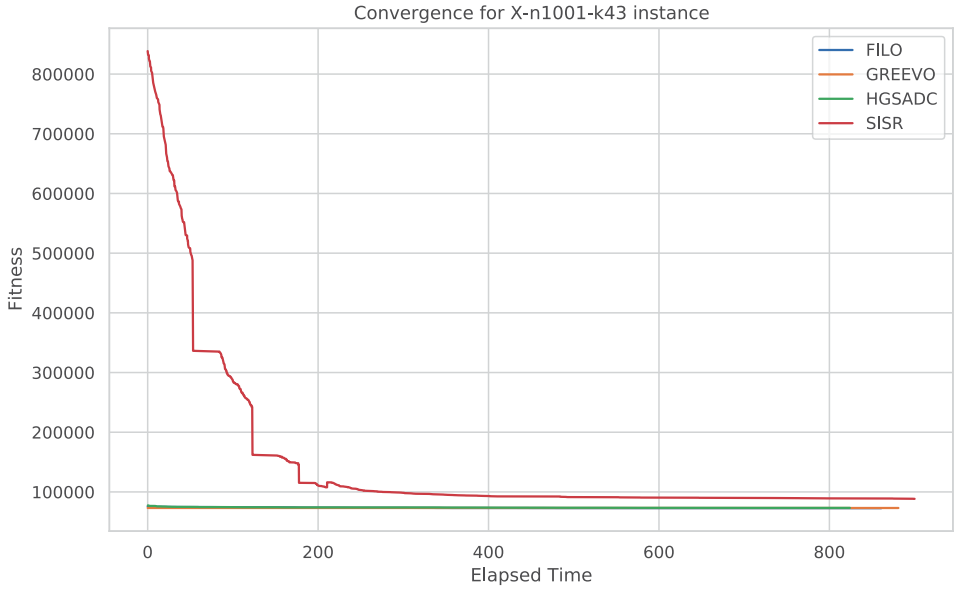



Figure 6.4: Convergence for X-n1001-k43 instance

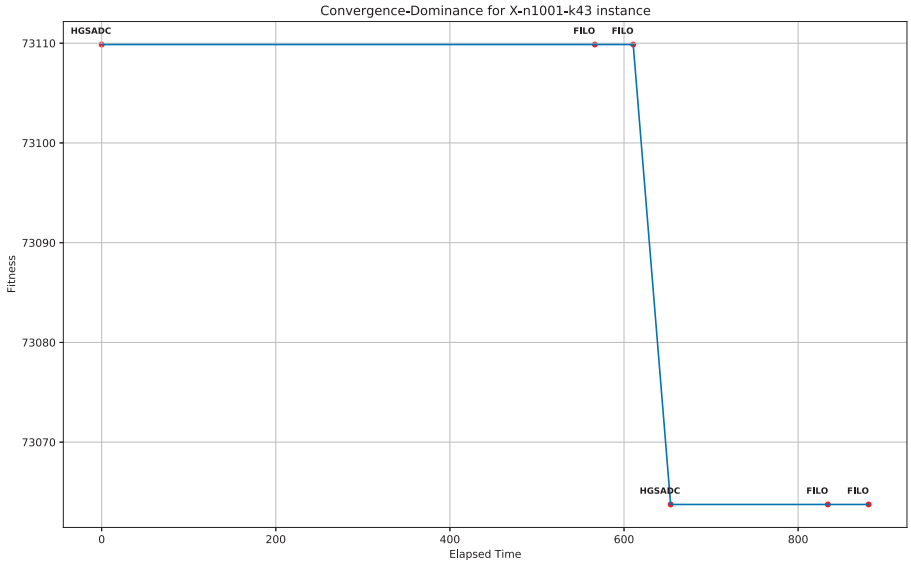



Figure 6.5: Convergence-Dominance for X-n1001-k43 instance

Figure 6.4 displays the convergence of each approach. From it, we can observe that GREEVO has a convergence rate similar to that of HGSADC and FILO. Additionally, these approaches start from almost the same initial solution/population.

The observations made from Figure 6.5 reveal that Q-learning employs multiple techniques during its evolutionary process. Additionally, even if one of the techniques does not contribute to the overall fitness of the solution, it presents an alternative solution with a different order of customer and route, which potentially helps the subsequent technique in enhancing the overall solution.

## 6.4 CASE STUDY

We were inspired by Tsai et al. (2021), which proposed a case study that involves an evacuation plan according to the City-Assisted Evacuation (CAE) guide for Hurricanes in New Orleans (NOHSEP, 2023) and an impact evaluation on it when considering the social distance. However, we will not consider the social distance aspect. Therefore, this case study characterized the pre-hurricane evacuation procedure as a Capacitated Vehicle Routing Problem (CVRP). In this context, a singular vehicle is assigned to each neighborhood, systematically initiating the process from a local rescue center, traversing various residences to collect individuals, and concluding the route by returning to the rescue center upon reaching its maximum capacity. This sequence continues until the vehicle successfully picks up every resident listed on the special needs registry within the designated neighborhood.

In structuring the problem, they use the city of New Orleans, Louisiana, USA, as the reference case. In their study, they used the dataset of Augerat (1995). The first 20 locations were selected from the A-n36-k5 dataset as Dataset 1, while Dataset 2 included all 35 locations in the A-n36-k5 dataset. For Datasets 3 and 4, they used the A-n53-k7 and A-n69-k9 datasets, respectively, which contain 52 and 68 houses. To account for the possibility that the depot is located outside the neighborhood was included predetermined transit times (0, 0.5, 1.0, and 2.0 hours) to and from a central depot.

New Orleans experienced massive losses and disruption in Hurricane Katrina (2005) (Dyson and Elliott, 2010) and has implemented comprehensive plans to prepare for a future major hurricane. Figure 6.6 shows the 72-hour evacuation timeline announced by the City of New Orleans, which specifies a window of 42 hours to collect evacuees from their houses and transport them to the Smoothie King Center, which serves as a transfer and processing center. From there, evacuees will board a bus to state or federal shelters in other cities (NOHSEP, 2023). Finally, the city government will bring evacuees back to their homes or local shelters once it becomes safe to return to New Orleans. As of February 23, 2021, the special needs registry for New Orleans included approximately 4000 individuals (Tsai et al., 2021).

To estimate the number of people in each house, Tsai et al. (2021) used the average household size (2014-2018), which is 2.44 persons per household in New Orleans, based on the American Community Survey (ACS) of the U.S. Census Bureau (U.S. Census Bureau, 2020), assuming a normal distribution with $mean = 2.44$ and standard $deviation = 0.5$. If the household size is not an integer, it is rounded to the nearest integer. So, the household size ranges from 1-4 people. For simplicity, they assume that if one person in a house is on the special needs registry, then all of the people in that house are also on the registry. An evacuation fleet was defined as a fleet of unlimited buses with a capacity of up to 64 passengers, as stated by Tsai et al. (2021). We also will use a range of 4 different speeds for the fleet, 5-20 km/h (ITF, 2018).

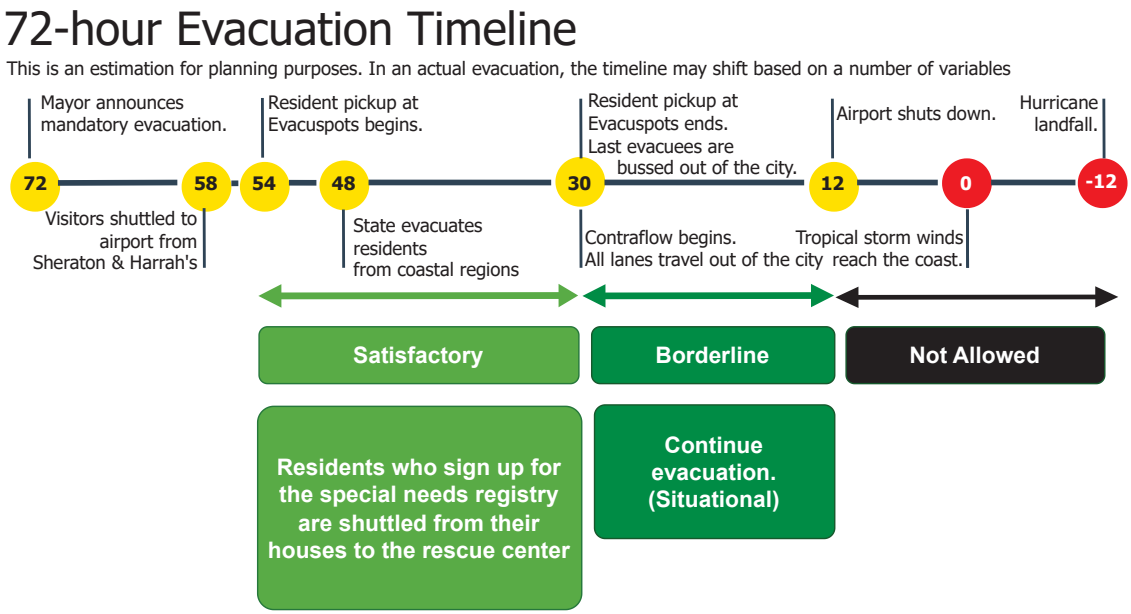



Figure 6.6: Evacuation plan of New Orleans (extracted from Tsai et al. (2021))

### 6.4.1 Result Analysis

Table 6.5 displays the number of evacuees in each neighborhood. We ran each defined dataset twice for five minutes. Furthermore, Figure 6.7 illustrates the average time that the vehicle with the largest path would take to evacuate all evacuees in each neighborhood based on the maximum speed for the fleet.

Table 6.5: Neighborhood Datasets

| Dataset | Neihgborhood Size | Maximium Demand |
|---|---|---|
| A-n21-k5 | 20 | 64 |
| A-n36-k5 | 35 | 100 |
| A-n53-k7 | 52 | 64 |
| A-n69-k9 | 68 | 100 |

For instance, one bus was allocated for A-n21-k5, two for A-n36-k5, three for A-n53-k7, and two for A-n69-k9. In most neighborhoods, the speed of 5 km/h is not sufficient to evacuate all evacuees in the time window available according to the New Orleans evacuation plan Tsai et al. (2021)). However, if we consider a speed of 10 km/h, with the allocated vehicles, it would be feasible to evacuate all evacuees. Lastly, if we consider 15 km/h, which is close to the average speed reported by ITF (2018) (16.1 km/h), it would be possible to conduct the evacuation within the satisfactory time window. Therefore, given the unpredictability of evacuation scenarios, it may be necessary to apply more resources to avoid any loss of life, and with the solution provided by GREEVO that could be done without excessively expending resources.

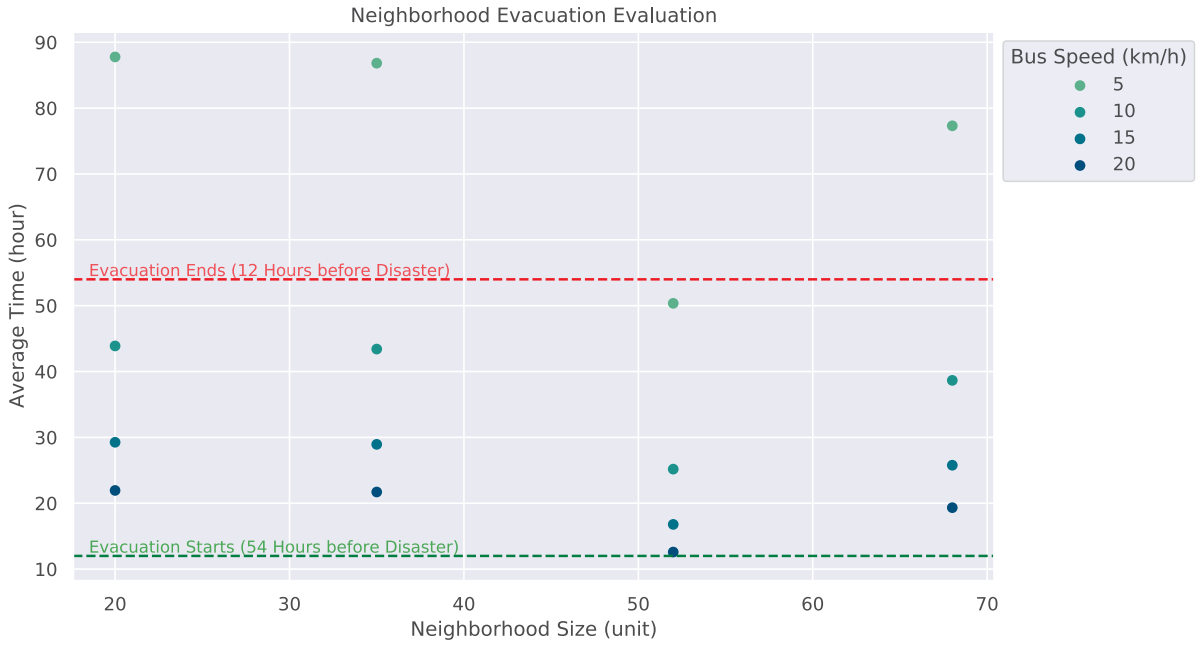



Figure 6.7: Neighborhood Evacuation Evaluation

## 6.5 CONCLUDING REMARKS

The experiments conducted to evaluate the efficiency of GREEVO in comparison to established approaches such as HGSADC, SISR, and FILO have provided valuable insights. These insights are not limited to the algorithm's performance across various benchmark instances and datasets but also include a possible way of using Hyper-heuristics with an ensemble method to tackle combinatorial optimization problems.

The statistical analysis, including the Friedman test and Kruskal-Wallis test with post-hoc comparisons, revealed that although GREEVO performs better when compared to the other algorithms, there is still room for improvement, especially related to consistency. This improvement may help approximate the average results obtained in the classical problem datasets.

The overall performance analysis, including a comprehensive examination of various statistics related to the solutions, indicated that GREEVO performed well in terms of average gaps, best solutions found, and overall competitiveness against other algorithms. The results suggest that GREEVO can be a valuable tool for addressing complex routing problems in practical scenarios.

The case study involving the evacuation plan for hurricanes in New Orleans provided a practical application of GREEVO, showcasing its effectiveness in solving a real-world Capacitated Vehicle Routing Problem. The results of the case study, particularly in the context of the New Orleans evacuation plan, demonstrated that GREEVO could provide feasible solutions for evacuating residents within specified time windows, considering different fleet speeds and neighborhood sizes. Regrettably, we were unable to conduct a comparative study between GREEVO and the disaster approaches referenced in Chapter 4 due to the unavailability of data, which was either not provided by the author or the referenced source.

It would be interesting if GREEVO could solve problems with more restrictions, such as time windows, multi-depot, and split delivery. However, due to the need to adapt the techniques present in both the ensemble (FILO and SISR) and the initialization (GRASP) of the population, it would require more time and testing to ensure that they would be able to solve the problems

either with all restrictions at the same time or separately. Unfortunately, this extension of the work has not been possible until now.

In the future, research efforts could be directed toward exploring and developing new policies for Q-learning. This could involve designing reward functions that align with the specific objectives of the problem, such as reducing costs or meeting time constraints. Moreover, conducting case studies in diverse contexts that involve more restrictions. Furthermore, it would be useful to investigate the scalability of GREEVO for larger problem instances and evaluate its performance under different constraints. Such efforts would contribute to a more comprehensive understanding of the applicability of Q-Learning and GREEVO in problem-solving contexts.

In conclusion, the combination of experimental evaluations and practical case studies positions GREEVO as a promising solution for addressing routing problems, with potential applications in various domains, including logistics, transportation, and emergency evacuation planning.

# 7 CONCLUSION AND FUTURE WORK

The study introduces GREEVO, a novel ensemble approach for solving the Capacitated Vehicle Routing Problem (CVRP). GREEVO combines elements of the Greedy Randomized Adaptive Search Procedure (GRASP), ensemble techniques, and a Q-learning hyper-heuristic. Typically, hyper-heuristics are used to select low-level heuristics, but in this study, we propose a new use for them by selecting high-level heuristics. This combination allows GREEVO to adapt to the dynamic search landscape during optimization, making it effective and efficient in providing competitive state-of-the-art results for CVRP.

In the past and nowadays, researchers have tried to use machine learning either on the dataset, searching for insights, or in the evolutionary process without achieving significant results (Queiroga et al., 2022; Sadana et al., 2024; Greif et al., 2024). GREEVO presents a way of using machine learning by applying an HH with reinforcement learning to select outstanding techniques (motivated by the machine learning statement: garbage in - garbage out) on an ensemble.

Experimental evaluations and practical case studies demonstrate GREEVO's promising performance across benchmark instances and datasets. Regarding GREEVO's performance, it outperforms HGSADC with a lower upper quartile and a more condensed distribution. This is further supported by the BEST GAP, which is the lowest among all approaches. However, there is room for improvement, particularly in terms of consistency. The comprehensive analysis of GREEVO's overall performance underscores its potential as a valuable tool for addressing complex routing problems in practical scenarios.

A case study involving the evacuation plan for hurricanes in New Orleans demonstrates GREEVO's effectiveness in providing feasible solutions for evacuating residents within specified time windows, considering different fleet speeds, and neighborhood sizes. Although the designed evacuation scenario only considers the CVRP, it demonstrates GREEVO's potential to provide efficient evacuation solutions while saving resources. Unfortunately, our efforts to undertake a comparative analysis between GREEVO and the disaster management methodologies outlined in Chapter 4 were impeded by the unavailability of pertinent data. This data, crucial for such a comparison, was either not furnished by the author or was inaccessible from the referenced sources. As a result, a comprehensive evaluation of GREEVO in relation to existing disaster management approaches could not be executed.

Further research should aim at improving GREEVO's consistency across a wider range of problem instances and datasets. Additionally, exploring GREEVO's suitability for problems with extra constraints, like time windows, multi-depot, and split delivery, is a promising direction for future work. The AVG GAP obtained by GREEVO provides insights into how the hyper-heuristic was applied, which includes refining reward functions and developing new selection policies for Q-learning that align more closely with specific problem objectives, as well as evaluating the scalability of GREEVO for larger problem instances. Moreover, testing different population management strategies and assessing the algorithm's adaptability to other combinatorial optimization problems beyond CVRP can help enhance its versatility and effectiveness.

In conclusion, continued research and exploration in these areas can further enhance the capabilities of GREEVO, contributing to its evolution as a versatile and powerful solution for addressing various combinatorial optimization challenges in different domains.

# REFERENCES


Abdallah, M. B. and Ennigrou, M. (2020). Hybrid Multi-agent Approach to Solve the Multi-depot Heterogeneous Fleet Vehicle Routing Problem with Time Window (MDHFVRPTW). *Advances in Intelligent Systems and Computing*, 923:376–386.

Accorsi, L. and Vigo, D. (2021). A fast and scalable heuristic for the solution of large-scale capacitated vehicle routing problems. *Transportation Science*, 55:832–856.

Aitsi-Selmi, A., Murray, V., Wannous, C., Dickinson, C., Johnston, D., Kawasaki, A., Stevance, A. S., and Yeung, T. (2016). Reflections on a Science and Technology Agenda for 21st Century Disaster Risk Reduction: Based on the Scientific Content of the 2016 UNISDR Science and Technology Conference on the Implementation of the Sendai Framework for Disaster Risk Reduction 2015–2030. *International Journal of Disaster Risk Science*, 7(1).

Altay, N. and Green, W. G. (2006). OR/MS research in disaster operations management. *European Journal of Operational Research*, 175(1):475–493.

Amiri, A. and Salari, M. (2019). Time-constrained maximal covering routing problem. *OR Spectrum*, 41(2):415–468.

Amit, S. N. K. B., Shiraishi, S., Inoshita, T., and Aoki, Y. (2016). Analysis of satellite images for disaster detection. In *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 5189–5192. IEEE.

Atmojo, P. S. and Sachro, S. S. (2017). Disaster Management: Selections of Evacuation Routes Due to Flood Disaster. In *Procedia Engineering*, volume 171, pages 1478–1485.

Augerat, P. (1995). *Approche Polyèdrale du Problème de Tournées de Véhicules*. Phd thesis, Institut National Polytechnique de Grenoble.

Bayram, V. (2016). Optimization models for large scale network evacuation planning and management: A literature review. *Surveys in Operations Research and Management Science*, 21(2):63–84.

Bergmann, B. and Hommel, G. (1988). Improvements of general multiple test procedures for redundant systems of hypotheses.

Birge, J. R. and Louveaux, F. (2011). *Introduction to Stochastic Programming*. Springer Publishing Company, Incorporated, 2nd edition.

Blum, C., Puchinger, J., Raidl, G. R., and Roli, A. (2011). Hybrid metaheuristics in combinatorial optimization: A survey. *Applied Soft Computing*, 11(6):4135–4151.

Bortfeldt, A. and Yi, J. (2020). The Split Delivery Vehicle Routing Problem with three-dimensional loading constraints. *European Journal of Operational Research*, 282(2):545–558.

Burke, E. K., Hyde, M., Kendall, G., Ochoa, G., Özcan, E., and Woodward, J. R. (2010). *A Classification of Hyper-heuristic Approaches*, pages 449–468. Springer US, Boston, MA.

Caric, T. and Gold, H. (2008). *Vehicle Routing Problem*. I-Tech.

Christiaens, J. and Berghe, G. (2020). Slack induction by string removals for vehicle routing problems. *Transportation Science*, 54(2):417–433.

Clarke, G. and Wright, J. W. (1964a). Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. *Operations Research*, 12(4):568–581.

Clarke, G. and Wright, J. W. (1964b). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12(4):568–581.

da Silva Junior, O. S., Leal, J. E., and Reimann, M. (2021). A multiple ant colony system with random variable neighborhood descent for the dynamic vehicle routing problem with time windows. *Soft Computing*, 25(4):2935–2948.

Dantas, A. and Pozo, A. (2022). The impact of state representation on approximate q-learning for a selection hyper-heuristic. In Xavier-Junior, J. C. and Rios, R. A., editors, *Intelligent Systems*, pages 45–60. Springer International Publishing.

Dantas, A., Rego, A. F. d., and Pozo, A. (2021). Using deep q-network for selection hyper-heuristics. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '21, page 1488–1492, New York, NY, USA. Association for Computing Machinery.

Dantzig, G. B. and Ramser, J. H. (1959). The truck dispatching problem. *Manage. Sci.*, 6:80–91.

Davis, L. (1985). Applying adaptive algorithms to epistatic domains. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'85, page 162–164, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Dewey, M. E. and Seneta, E. (2001). *Carlo Emilio Bonferroni*, pages 411–414. Springer New York, New York, NY.

Dixit, V. and Wolshon, B. (2014). Evacuation traffic dynamics. *Transportation Research Part C: Emerging Technologies*, 49:114–125.

Do Rego, A. F. and Pozo, A. (2021). Influence of a grasp initialization on the performance of the hybrid genetic search with adaptive diversity control. In *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 01–08.

Dror, M. and Trudeau, P. (1989). Savings by Split Delivery Routing. *Transportation Science*, 23(2):141–145.

Dyson, M. and Elliott, P. (2010). *Come Hell Or High Water: Hurricane Katrina and the Color of Disaster*. ReadHowYouWant.com, Limited.

Erdoğan, G. (2017). An open source spreadsheet solver for vehicle routing problems. *Computers & Operations Research*, 84:62–72.

Esposito Amideo, A., Scaparra, M. P., and Kotiadis, K. (2019). Optimising shelter location and evacuation routing operations: The critical issues. *European Journal of Operational Research*, 279(2):279–295.

FEMA, F. E. M. A. (2019). National response framework. https://www.fema.gov/media-library-data/1559136348938-063ec40e34931923814dd50df638b448/NationalResponseFrameworkFourthEdition.pdf. Accessed 2019.10.20.

Feo, T. A. and Resende, M. G. (1995). Greedy randomized adaptive search procedures. *Journal of global optimization*, 6(2):109–133.

Feo, T. A. and Resende, M. G. C. (1989). A probabilistic heuristic for a computationally difficult set covering problem. *Oper. Res. Lett.*, 8(2):67–71.

Flanagan, B. E., Gregory, E. W., Hallisey, E. J., Heitgerd, J. L., and Lewis, B. (2011). A social vulnerability index for disaster management. *Journal of Homeland Security and Emergency Management*, 8:1547–7355. DOI: https://doi.org/10.2202/1547-7355.1792.

Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 32(200):675–701.

Gendreau, M. and Potvin, J.-Y. (2010). *Handbook of Metaheuristics*. Springer Publishing Company, Incorporated, 2nd edition.

Gillett, B. E. and Miller, L. R. (1974). A heuristic algorithm for the vehicle-dispatch problem. *Operations Research*, 22(2):340–349.

Goerigk, M., Deghdak, K., and T'Kindt, V. (2015). A two-stage robustness approach to evacuation planning with buses. *Transportation Research Part B: Methodological*, 78:66–82.

Goldberg, D. E., Lingle, R., et al. (1985). Alleles, loci, and the traveling salesman problem. In *Proceedings of an international conference on genetic algorithms and their applications*, volume 154, pages 154–159. Lawrence Erlbaum, Hillsdale, NJ.

Golden, B., Assad, A., Levy, L., and Gheysens, F. (1984). The fleet size and mix vehicle routing problem. *Computers & Operations Research*, 11(1):49–66.

Golden, B. L., Wasil, E. A., Kelly, J. P., and Chao, I.-M. (1998). *The Impact of Metaheuristics on Solving the Vehicle Routing Problem: Algorithms, Problem Sets, and Computational Results*, pages 33–56. Springer US, Boston, MA.

Gong, W., Zhou, A., and Cai, Z. (2015). A multioperator search strategy based on cheap surrogate models for evolutionary optimization. *IEEE Transactions on Evolutionary Computation*, 19(5):746–758.

Greif, T., Bouvier, L., Flath, C. M., Parmentier, A., Rohmer, S. U., and Vidal, T. (2024). Combinatorial optimization and machine learning for dynamic inventory routing. *arXiv preprint arXiv:2402.04463*.

Gutierrez, A., Dieulle, L., Labadie, N., and Velasco, N. (2018). A hybrid metaheuristic algorithm for the vehicle routing problem with stochastic demands. *Computers & Operations Research*, 99:135–147.

Holland, J. H. (1975). Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. *anas*.

Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. MIT Press, Cambridge, MA, USA.

Hoos, H. H. and Stützle, T. (2007). *Handbook of Approximation Algorithms and Metaheuristics*. Chapman and Hall/CRC.

Huberman, B., Lukose, R., and Hogg, T. (1997). An economics approach to hard computational problems. *Science*, 275(5296):51–54.

IFRC (2019). What is a disaster? `https://www.ifrc.org/en/what-we-do/disaster-management/about-disasters/what-is-a-disaster/`.

ITF (2018). Rethinking urban mobility: A global outlook from cities. `https://www.itf-oecd.org/sites/default/files/docs/urban-mobility-report-2018.pdf`.

Jun, Y. and Kim, B.-I. (2012). New best solutions to vrpspd benchmark problems by a perturbation based algorithm. *Expert Syst. Appl.*, 39(5):5641–5648.

Kaneko, K. (2017). Map database design for route finding and natural disaster risk evaluation. *Proceedings - 2016 International Conference on Agents, ICA 2016*, pages 139–141.

Kim, K.-I. (2014). A16. Visualization of Ship Collision Risk Based on Near-miss Accidents. *Cuadernos de Desarrollo Rural*, 3:1–6.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598):671–680.

Kitajima, R., Kamimura, R., Uchida, O., and Toriumi, F. (2016). Neural potential learning for tweets classification and interpretation. *Proceedings of the 2015 7th International Conference of Soft Computing and Pattern Recognition, SoCPaR 2015*, pages 141–148.

Kropat, E. and Meyer-Nieberg, S. (2016). A multi-layered adaptive network approach for shortest path planning during critical operations in dynamically changing and uncertain environments. *Proceedings of the Annual Hawaii International Conference on System Sciences*, 2016-March:1369–1378.

Kruskal, W. H. and Wallis, W. A. (1952). Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association*, 47(260):583–621.

Labadie, N., Prins, C., and Prodhon, C. (2016). *Metaheuristics for Vehicle Routing Problems*. John Wiley & Sons, Inc., Hoboken, NJ, USA.

Le, V. M., Chevaleyre, Y., Vinh, H. T., and Zucker, J. D. (2015). Hybrid of linear programming and genetic algorithm for optimizing agent-based simulation. Application to optimization of sign placement for tsunami evacuation. *Proceedings - 2015 IEEE RIVF International Conference on Computing and Communication Technologies: Research, Innovation, and Vision for Future, IEEE RIVF 2015*, pages 138–143.

Li, D. and Zhou, H. (2018). Robust Optimization for Vehicle Routing Problem under Uncertainty in Disaster Response. In *2018 15th International Conference on Service Systems and Service Management, ICSSSM 2018*. Institute of Electrical and Electronics Engineers Inc.

Li, J., Li, Y., and Pardalos, P. M. (2014a). Multi-depot vehicle routing problem with time windows under shared depot resources. *Journal of Combinatorial Optimization*, 31:515–532.

Li, K., Fialho, A., Kwong, S., and Zhang, Q. (2014b). Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 18(1):114–130.

Li, M., Xu, J., Wei, L., Jia, X., and Sun, C. (2019). Modeling a Risk-Based Dynamic Bus Schedule Problem under No-Notice Evacuation Incorporated with Dynamics of Disaster, Supply, and Demand Conditions. *Journal of Advanced Transportation*, 2019.

Liu, R. and Jiang, Z. (2019). A hybrid large-neighborhood search algorithm for the cumulative capacitated vehicle routing problem with time-window constraints. *Applied Soft Computing*, 80:18–30.

London, R. G. (2018 (accessed November 4, 2019)). London resilience mass evacuation framework version 3.0. `https://www.london.gov.uk/sites/default/files/mass_evacuation_framework_2018_v3.0_0.pdf`.

Luke, S. (2013). *Essentials of Metaheuristics*. Lulu, second edition. Available for free at http://cs.gmu.edu/∼sean/book/metaheuristics/.

Maldonado, M., Alulema, D., Morocho, D., and Proano, M. (2017). System for monitoring natural disasters using natural language processing in the social network Twitter. In *Proceedings - International Carnahan Conference on Security Technology*, pages 1–6. IEEE.

Mallipeddi, R. and Lee, M. (2015). An evolving surrogate model-based differential evolution algorithm. *Applied Soft Computing Journal*, 34:770–787. cited By 51.

Mallipeddi, R., Suganthan, P., Pan, Q., and Tasgetiren, M. (2011a). Differential evolution algorithm with ensemble of parameters and mutation strategies. *Applied Soft Computing*, 11(2):1679–1696. The Impact of Soft Computing for the Progress of Artificial Intelligence.

Mallipeddi, R., Suganthan, P., Pan, Q., and Tasgetiren, M. (2011b). Differential evolution algorithm with ensemble of parameters and mutation strategies. *Applied Soft Computing Journal*, 11(2):1679–1696.

Mole, R. H. and Jameson, S. R. (1976). A sequential route-building algorithm employing a generalised savings criterion. *Journal of the Operational Research Society*, 27(2):503–511.

MunichRe (2019). Natcatservice.

Máximo, V. R. and Nascimento, M. C. (2021). A hybrid adaptive iterated local search with diversification control to the capacitated vehicle routing problem. *European Journal of Operational Research*.

Nagata, Y. and Bräysy, O. (2009). Edge assembly-based memetic algorithm for the capacitated vehicle routing problem. *Netw.*, 54(4):205–215.

Nagata, Y. and Bräysy, O. (2009). A powerful route minimization heuristic for the vehicle routing problem with time windows. *Operations Research Letters*, 37(5):333–338.

Nagata, Y., Bräysy, O., and Dullaert, W. (2010). A penalty-based edge assembly memetic algorithm for the vehicle routing problem with time windows. *Computers & Operations Research*, 37(4):724–737.

Nagata, Y. and Kobayashi, S. (1997). Edge assembly crossover: A high-power genetic algorithm fot the traveling salesman problem. In *Proceedings of the 7th International Conference on Genetic Algorithms, 1997*.

Nagata, Y. and Kobayashi, S. (2010). Guided ejection search for the pickup and delivery problem with time windows. In Cowling, P. and Merz, P., editors, *Evolutionary Computation in Combinatorial Optimization*, pages 202–213, Berlin, Heidelberg. Springer Berlin Heidelberg.

Nagata, Y. and Kobayashi, S. (2013). A powerful genetic algorithm using edge assembly crossover for the traveling salesman problem. *INFORMS J. on Computing*, 25(2):346–363.

Neal, D. M. (1997). Reconsidering the Phases of Disaster. *International Journal of Mass Emergencies and Disasters*, 15(2):239–264.

NOHSEP (2023). Hurricane - nola ready.

Oliver, I., Smith, D., and Holland, J. R. (1987). Study of permutation crossover operators on the traveling salesman problem. In *Genetic algorithms and their applications: proceedings of the second International Conference on Genetic Algorithms: July 28-31, 1987 at the Massachusetts Institute of Technology, Cambridge, MA*. Hillsdale, NJ: L. Erlhaum Associates, 1987.

P. Toth and D. Vigo, editor (2014). *Vehicle Routing: Problems, Methods, and Applications, Second Edition*. SIAM.

Pecin, D. G., Pessoa, A., Poggi, M., and Uchoa, E. (2017). Improved branch-cut-and-price for capacitated vehicle routing. *Mathematical Programming Computation*, 9:61–100.

Penna, P. H. V., Santos, A. C., and Prins, C. (2018). Vehicle routing problems for last mile distribution after major disaster. *Journal of the Operational Research Society*, 69(8):1254–1268.

Pisinger, D. and Ropke, S. (2007). A general heuristic for vehicle routing problems. *Comput. Oper. Res.*, 34(8):2403–2435.

Potvin, J.-Y. and Bengio, S. (1996). The vehicle routing problem with time windows part ii: Genetic search. *INFORMS J. on Computing*, 8(2):165–172.

Prins, C. (2004). A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & Operations Research*, 31(12):1985 – 2002.

Prins, C., Lacomme, P., and Prodhon, C. (2014). Order-first split-second methods for vehicle routing problems: A review. *Transportation Research Part C: Emerging Technologies*, 40:179–200.

Psaraftis, H. (1988). *Dynamic Vehicle Routing Problems*, pages 223–248. North-Holland.

Queiroga, E., Sadykov, R., Uchoa, E., and Vidal, T. (2022). 10,000 optimal CVRP solutions for testing machine learning based heuristics. In *AAAI-22 Workshop on Machine Learning for Operations Research (ML4OR)*.

Rabbouch, B., Saâdaoui, F., and Mraihi, R. (2019). Efficient implementation of the genetic algorithm to solve rich vehicle routing problems. *Operational Research*.

Sadana, U., Chenreddy, A., Delage, E., Forel, A., Frejinger, E., and Vidal, T. (2024). A survey of contextual optimization methods for decision-making under uncertainty. *European Journal of Operational Research*.

Salhi, S. and Sari, M. (1997). A multi-level composite heuristic for the multi-depot vehicle fleet mix problem. *European Journal of Operational Research*, 103(1):95–112.

Shahparvari, S. and Abbasi, B. (2017). Robust stochastic vehicle routing and scheduling for bushfire emergency evacuation: An Australian case study. *Transportation Research Part A: Policy and Practice*, 104:32–49.

Shahparvari, S., Abbasi, B., Chhetri, P., and Abareshi, A. (2019). Fleet routing and scheduling in bushfire emergency evacuation: A regional case study of the Black Saturday bushfires in Australia. *Transportation Research Part D: Transport and Environment*, 67:703–722.

Shi, Y., Boudouh, T., and Grunder, O. (2017). A hybrid genetic algorithm for a home health care routing problem with time window and fuzzy demand. *Expert Systems with Applications*, 72:160 – 176.

Subramanian, A., Penna, P. H. V., Uchoa, E., and Ochi, L. S. (2012). A hybrid algorithm for the heterogeneous fleet vehicle routing problem. *European Journal of Operational Research*, 221(2):285–295.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA.

Syswerda, G. (1991). Schedule optimization using genetic algorithms. In *The Genetic Algorithms Handbook*.

Sze, J. F., Salhi, S., and Wassan, N. (2016). A hybridisation of adaptive variable neighbourhood search and large neighbourhood search: Application to the vehicle routing problem. *Expert Systems with Applications*, 65:383–397.

Talebian Sharif, M. and Salari, M. (2015). A GRASP algorithm for a humanitarian relief transportation problem. *Engineering Applications of Artificial Intelligence*, 41(3):259–269.

Toth, P., Vigo, D., Toth, P., and Vigo, D. (2014). *Vehicle Routing: Problems, Methods, and Applications, Second Edition*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

Tsai, Y.-L., Rastogi, C., Kitanidis, P. K., and Field, C. B. (2021). Routing algorithms as tools for integrating social distancing with emergency evacuation. *Scientific Reports*, 11(1):19623.

Uchoa, E., Pecin, D., Pessoa, A., Poggi, M., Vidal, T., and Subramanian, A. (2017). New benchmark instances for the capacitated vehicle routing problem. *European Journal of Operational Research*, 257(3):845–858.

UNDRR (2017). Terminology. `https://www.unisdr.org/we/inform/terminology`.

UNDRR (2019). Why do we differentiate disasters from natural hazards? https://www.stopdisastersgame.org/.

U.S. Census Bureau (Year of the data, e.g., 2020). American community survey (acs).

Van Wassenhove, L. N. (2006). Humanitarian aid logistics: supply chain management in high gear. *Journal of the Operational Research Society*, 57(5):475–489.

Vidal, T., Crainic, T. G., Gendreau, M., Lahrichi, N., and Rei, W. (2012). A hybrid genetic algorithm for multidepot and periodic vehicle routing problems. *Operation Research*, 60(3):611–624.

Vidal, T., Crainic, T. G., Gendreau, M., and Prins, C. (2013). A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. *Computers and Operations Research*, 40:475–489.

Vieira, Y. E. M., de Mello Bandeira, R. A., and da Silva Júnior, O. S. (2021). Multi-depot vehicle routing problem for large scale disaster relief in drought scenarios: The case of the Brazilian northeast region. *International Journal of Disaster Risk Reduction*, 58(March):102193.

Vitali, J. L., Riff, M.-C., and Montero, E. (2017). Bus routing for emergency evacuations: The case of the great fire of valparaiso. In *2017 IEEE Congress on Evolutionary Computation (CEC)*, pages 2346–2353.

Wang, F., Liao, F., Li, Y., Yan, X., and Chen, X. (2021). An ensemble learning based multi-objective evolutionary algorithm for the dynamic vehicle routing problem with time windows. *Computers and Industrial Engineering*, 154.

Wang, X., Choi, T. M., Liu, H., and Yue, X. (2018). A novel hybrid ant colony optimization algorithm for emergency transportation problems during post-disaster scenarios. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 48(4):545–556.

Wang, Y. and Li, B. (2010). Multi-strategy ensemble evolutionary algorithm for dynamic multi-objective optimization. *Memetic Computing*, 2(1):3–24. cited By 54.

Watkins, C. J. C. H. and Dayan, P. (1992). Q-learning. *Machine Learning*, 8(3):279–292.

Wolpert, D. and Macready, W. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82.

Wu, G., Mallipeddi, R., Suganthan, P., Wang, R., and Chen, H. (2016). Differential evolution with multi-population based ensemble of mutation strategies. *Information Sciences*, 329:329–345.

Wu, G., Mallipeddi, R., and Suganthan, P. N. (2019). Ensemble strategies for population-based optimization algorithms – a survey. *Swarm and Evolutionary Computation*, 44:695–711.

Yao, T., Mandala, S. R., and Chung, B. D. (2009). Evacuation Transportation Planning Under Uncertainty: A Robust Optimization Approach. *Networks and Spatial Economics*, 9(2):171–189.

Yu, L. and Liu, X. (2014). Debris flow risk assessment using principal components analysis and rough set techniques. *Proceedings of the 33rd Chinese Control Conference, CCC 2014*, pages 7179–7182.

Zhang, S., Chen, M., Zhang, W., and Zhuang, X. (2020). Fuzzy optimization model for electric vehicle routing problem with time windows and recharging stations. *Expert Systems with Applications*, 145.

Zhao, S.-Z., Suganthan, P., and Zhang, Q. (2012). Decomposition-based multiobjective evolutionary algorithm with an ensemble of neighborhood sizes. *IEEE Transactions on Evolutionary Computation*, 16(3):442–446.

Zhao, X., Xu, W., Ma, Y., Qin, L., Zhang, J., and Wang, Y. (2017). Relationships Between Evacuation Population Size, Earthquake Emergency Shelter Capacity, and Evacuation Time. *International Journal of Disaster Risk Science*, 8(4):457–470.

Zhen, L., Ma, C., Wang, K., Xiao, L., and Zhang, W. (2020). Multi-depot multi-trip vehicle routing problem with time windows and release dates. *Transportation Research Part E: Logistics and Transportation Review*, 135:101866.