University of the Aegean
School of Business
Department of Financial and Management Engineering

Emergency evacuation planning in natural disasters: Models and solution approaches

Evangelia Baou

Dissertation Committee:

Professor Ioannis Minis, Supervisor

Professor Agapios Platis, Examination Committee Member

Vasileios Zeimpekis, Ph.D., Examination Committee Member

Chios, 2017

Acknowledgements

Firstly, I would like to thank my Professor Ioannis Minis for giving me the opportunity to undertake the present diploma thesis, for his time, as well for the supervision. I would also like to thank Mr Vasileios Koutras, Lecturer of the University of the Aegean, for his guidance and for his contributions during the past year; without his support I would not be able to complete my diploma thesis.

Furthermore, I am grateful to Vasileios Zeimpekis, PhD Candidate of the University of the Aegean, for his advises throughout the project. Finally, I am grateful to all members of the Design, Operations & Production Systems Lab (DeOPSys) of the Financial and Management Engineering (FME) Department of the University of the Aegean, and mainly to Mrs. Christina Arampantzi.

Last but not least, I would like to thank my parents, Panagiotis and Maria, my sister, Vasia, and all my lovely friends for their continued psychological support and encouragement.

Εκτενής Ελληνική Περίληψη

(Summary in Greek)

Εισαγωγή

Ως έκτακτη κατάσταση ορίζεται συμβάν εκτός κανονικών συνθηκών που ενδέχεται να προκαλέσει βλάβη στην οποία είναι δύσκολο να ανταπεξέλθουν και να προσαρμοστούν οι επηρεαζόμενοι. Οι καταστροφές διαφέρουν στο επίπεδο βλάβης που προκαλούν και έχουν σημαντικές επιπτώσεις, όπως απώλειες ανθρώπινων ζωών, υλικές και περιβαλλοντικές βλάβες κ.α. Τις τελευταίες δεκαετίες η συχνότητα καταστροφών αυξάνεται παγκοσμίως. Πιο συγκεκριμένα, την τελευταία δεκαετία οι καταστροφές έχουν προκαλέσει οικονομικές ζημιές ύψους 1.4 τρισεκατομμυρίων δολαρίων, επηρεάζοντας 1.7 δισεκατομμύρια ανθρώπους εκ των οποίων οι 0.7 εκατομμύρια έχασαν τη ζωή τους [5].

Οι καταστροφές χωρίζονται σε τρείς βασικές κατηγορίες: φυσικές, τεχνολογικές και κοινωνικές. Ο κύκλος αντιμετώπισής τους αποτελείται από τέσσερις φάσεις. Οι πρώτες δύο, 'μετριασμός' (mitigation) και 'ετοιμότητα' (preparedness), υλοποιούνται πριν ξεσπάσει μια καταστροφή ενώ οι δύο τελευταίες, 'απόκριση' (response) και 'ανάκτηση' (recovery), έπονται της καταστροφής.

Σε καταστάσεις έκτατης ανάγκης και όταν πρέπει να διασωθεί πληθυσμός, σχεδιάζεται η εκκένωση περιοχών και η μεταφορά του πληθυσμού τους σε ασφαλή μέρη, με τον καλύτερο δυνατό τρόπο, ώστε να αποφευχθούν απώλειες ανθρώπινων ζωών. Ο σχεδιασμός εκκένωσης χαρακτηρίζεται ως μέτρο της δεύτερης φάσης ('ετοιμότητα') και η εκκένωση υλοποιείται κατά τη διάρκεια της τρίτης φάσης ('απόκριση'). Ο σχεδιασμός εκκένωσης αποτελεί συνήθη τακτική για την αντιμετώπιση των καταστάσεων εκτάκτου ανάγκης ενώ η εκκένωση ορίζεται ως η διαδικασία κατά την οποία άνθρωποι, οι οποίοι απειλούνται από μια καταστροφή, μετακινούνται από τις απειλούμενες περιοχές σε μέρη με μεγαλύτερη ασφάλεια.

Παρά το γεγονός ότι οι επηρεαζόμενοι θα μπορούσαν να χρησιμοποιήσουν τα δικά τους οχήματα κατά τη διάρκεια μιας εκκένωσης, η ατομική εκκένωση μπορεί να προκαλέσει επιπλοκές και καθυστερήσεις με συνέπεια την απώλεια ανθρώπινων ζωών. Για το λόγο αυτό, οι αρχές αναλαμβάνουν τον σχεδιασμό συντονισμένης εκκένωσης με δημόσια μέσα.

Η εκκένωση αποτελεί πολύπλοκη διαδικασία της οποίας η επιτυχία εξαρτάται από πολλούς παράγοντες, όπως τα δρομολόγια, στρατηγικές ελέγχου της κυκλοφοριακής συμφόρησης, συμπεριφορά των εκκενωθέντων κλπ.

Ορισμός προβλήματος

Στην παρούσα διπλωματική εργασία αναπτύσσεται μεθοδολογικό πλαίσιο για τον σχεδιασμό της εκκένωσης περιοχών και την μεταφορά του πληθυσμού σε καταφύγια σε περίπτωση φυσικής καταστροφής. Αναλυτικότερα, μοντελοποιείται το Πρόβλημα Εκκένωσης Πληθυσμού χρησιμοποιώντας Ετερογενή Στόλο οχημάτων (ΠΕΠΕΣ). Το μαθηματικό μοντέλο που αναπτύσσεται για το συγκεκριμένο πρόβλημα στοχεύει στον καθορισμό των δρομολογίων τα οποία ελαχιστοποιούν τον χρόνο εκκένωσης. Στο πρόβλημά μας, ο χρόνος εκκένωσης ορίζεται ως ο χρόνος που ο τελευταίος κάτοικος φθάνει σε ασφαλές καταφύγιο.

Στο ΠΕΠΕΣ, χρησιμοποιείται στόλος οχημάτων με διαφορετικά χαρακτηριστικά, όσον αφορά την χωρητικότητά τους. Τα οχήματα καλούνται να συλλέξουν τους κατοίκους συγκεκριμένων πληθυσμιακών συγκεντρώσεων, οι οποίες βρίσκονται υπό απειλή (π.χ. δασική πυρκαγιά), και να τους μεταφέρουν σε ασφαλέστερο μέρος (καταφύγιο). Η ζήτηση κάθε τοποθεσίας είναι γνωστή εξ αρχής και το καταφύγιο είναι απεριόριστης χωρητικότητας.

Για την ανάπτυξη του κατάλληλου μαθηματικού μοντέλου, αρχικά, εντοπίστηκαν σημαντικές ομοιότητες και διαφορές με τα ήδη υπάρχοντα προβλήματα της σχετικής βιβλιογραφίας. Αξιοποιώντας τα αποτελέσματα της βιβλιογραφίας και λαμβάνοντας υπόψη τα ιδιαίτερα χαρακτηριστικά του υπό μελέτη προβλήματος, αναπτύχθηκε νέο μοντέλο ΜΑΓΠ.

Δεδομένα προβλήματος

Η διατύπωση του μαθηματικού μοντέλου προϋποθέτει την γνώση των δεδομένων του προβλήματος. Πιο συγκεκριμένα, για κάθε πληθυσμιακή συγκέντρωση (χωριό), τα δεδομένα εισόδου του προβλήματος περιλαμβάνουν τον ακριβή αριθμό κάτοικων που πρόκειται να μεταφερθούν καθώς και τις κατηγορίες αυτών. Στο πρόβλημά μας, οι κάτοικοι χωρίζονται σε τρείς κατηγορίες οι οποίες δημιουργήθηκαν με βάση τις ανάγκες τους κατά την διάρκεια της μεταφοράς τους. Η πρώτη κατηγορία περιλαμβάνει άτομα τα οποία δεν χρειάζονται ειδική μεταχείριση κατά την μεταφορά τους. Η δεύτερη κατηγορία περιλαμβάνει άτομα με προβλήματα κινητικότητας τα οποία μεταφέρονται σε αναπηρικό αμαξίδιο και κατά την διάρκεια της εκκένωσης θα χρειαστούν οχήματα με ειδικά χαρακτηριστικά. Τέλος, η τρίτη κατηγορία αφορά άτομα τα οποία θα μεταφερθούν με ασθενοφόρο.

Επιπλέον δεδομένα εισόδου αποτελούν οι ακριβείς τοποθεσίες των πληθυσμιακών συγκεντρώσεων και οι αντίστοιχο οδικό δίκτυο (αποστάσεις και χρόνοι διαδρομής). Όσον αφορά τα οχήματα, η χωρητικότητά τους δίδεται σχετικά με τις παραπάνω κατηγορίες εκκενωθέντων.

Προσεγγιστική Επίλυση Προβλήματος

Η βέλτιστη επίλυση του μαθηματικού μοντέλου είναι δύσκολο να επιτευχθεί σε εύλογο χρόνο για προβλήματα πρακτικού μεγέθους. Για την αντιμετώπιση αυτών των δυσκολιών, αναπτύσσονται ευρετικοί αλγόριθμοι οι οποίοι παρέχουν αποδοτικές προσεγγιστικές λύσεις σε λογικά χρονικά πλαίσια.

Στην παρούσα διπλωματική εργασία αναπτύσσονται δυο ευρετικοί αλγόριθμοι για το ΠΕΠΕΣ. Οι δυο ευρετικοί αλγόριθμοι εντοπίζουν δρομολόγια χρησιμοποιώντας τον διαθέσιμο στόλο οχημάτων με σκοπό την εκκένωση του πληθυσμού που βρίσκεται στα σημεία συγκέντρωσης και την μεταφορά του στο καταφύγιο, με αντικειμενικό στόχο την ελαχιστοποίηση του συνολικού χρόνου εκκένωσης του πληθυσμού υπό περιορισμούς σχετικούς με τη χωρητικότητα των διαθέσιμων οχημάτων, του διαθέσιμου οδικού δικτύου κα. Να σημειωθεί ότι, το σχέδιο εκκένωσης και δρομολόγησης που προκύπτει από τους δύο αλγορίθμους, εκκενώνει ολόκληρο τον πληθυσμό. Αρχικά, οι προτεινόμενοι αλγόριθμοι εφαρμόζονται σε απλούστερη περίπτωση του προβλήματος εκκένωσης στην οποία ουδείς από τους εκκενωθέντες αντιμετωπίζει πρόβλημα κινητικότητας. Στη συνέχεια, οι δυο αλγόριθμοι συγκρίνονται όσον αφορά στο συνολικό χρόνο εκκένωσης και αυτός με τον ελάχιστο χρόνο εκκένωσης επιλέγεται να εφαρμοστεί σε πολύπλοκη εκδοχή του προβλήματος εκκένωσης, στην οποία οι εκκενωθέντες εμπίπτουν σε πολλαπλές κατηγορίες.

Για την σύγκριση των αλγορίθμων, αναπτύχθηκε γεννήτρια προβλημάτων και οι αλγόριθμοι μελετήθηκαν σχετικά με βασικές παραμέτρους του προβλήματος. Στο πρώτο σενάριο ο αριθμός των πληθυσμιακών συγκεντρώσεων παραμένει σταθερός και ο αριθμός των οχημάτων μεταβάλλεται, ενώ στο δεύτερο σενάριο ο αριθμός των πληθυσμιακών συγκεντρώσεων μεταβάλλεται και ο αριθμός των οχημάτων παραμένει σταθερός. Για κάθε συνδυασμό οχημάτων- πληθυσμιακών συγκεντρώσεων δημιουργήθηκαν 100 προβλήματα, τα οποία λύθηκαν και με τους δύο αλγορίθμους. Ακολούθως, υπολογίστηκε η μέση τιμή του χρόνου εκκένωσης για κάθε συνδυασμό προβλημάτων για κάθε αλγόριθμο.

Από τα αποτελέσματα συνεπάγεται ότι ο ευρετικός αλγόριθμος 1 (H1) έχει καλύτερη απόδοση όσον αφορά στον χρόνο εκκένωσης, σε σχέση με τον ευρετικό αλγόριθμο 2 (H2). Αυτο συμβαίνει διότι ο ευρετικός αλγόριθμος 1 αξιοποιεί περισσότερα οχήματα για την ολοκλήρωση της εκκένωσης σε αντίθεση με τον ευρετικό αλγόριθμο 2. Συνεπώς ο ευρετικός αλγόριθμος 1 ολοκληρώνει την εκκένωση σε λιγότερο χρόνο και για το λόγο αυτό, επιλέγεται να εφαρμοστεί στην πιο περίπλοκη περίπτωση (μελέτη περίπτωσης).

Μελέτη Περίπτωσης

Η μελέτη περίπτωσης που παρουσιάζεται σε αυτή την διπλωματική εργασία εστιάζει σε ρεαλιστική φυσική καταστροφή. Πιο συγκεκριμένα μελετούμε δασική πυρκαγιά η οποία εξελίσσεται δυναμικά στην επαρχία του Teruel στην Ισπανία. Οι απειλούμενες περιοχές είναι τρία μικρά χωριά, το Tramacastiel, το Rubiales και το El Campillo με πληθυσμό 44, 31 και 40 κατοίκους αντίστοιχα, κοντά στην επαρχία του Teruel.

Χρησιμοποιώντας τον ευρετικό αλγόριθμο 1, στοχεύουμε στην ανάπτυξη του κατάλληλου πλάνου εκκένωσης για την επαρχία του Teruel. Ειδικότερα, εστιάζουμε στην δημιουργία προσεγγιστικών λύσεων για τρία διαφορετικά σενάρια. Το πρώτο σενάριο, αφορά την εκκένωση του Tramacastiel και την μεταφορά των εκκενωθέντων στο Villel, μια μικρή πόλη στην επαρχία του Teruel. Το δεύτερο σενάριο, αφορά την εκκένωση και των τριών χωριών και την μεταφορά των κατοίκων τους σε ασφαλές καταφύγιο στην πόλη του Teruel. Τέλος, το τρίτο σενάριο αντιμετωπίζει την εκκένωση των τριών αυτών χωριών σε περίπτωση όπου η δασική πυρκαγιά εξελίσσεται σύμφωνα με τις καιρικές συνθήκες. Σε αυτό το σενάριο, η πυρκαγιά απειλεί αρχικά το Tramacastiel και έπειτα ταυτόχρονα το Rubiales και το El Campillo.

Στην μελέτη περίπτωσης του Teruel αντιμετωπίστηκε η πρόκληση μεταφοράς συγκεκριμένων κατηγοριών εκκενωθέντων με προβλήματα κινητικότητας που απαιτούν χρήση ειδικών οχημάτων. Παρατηρήθηκε ότι τα άτομα με κινητικά προβλήματα επηρεάζουν σημαντικά τον χρόνο εκκένωσης και ότι η χρήση περισσότερων οχημάτων, ειδικά προσαρμοσμένων στις ανάγκες τέτοιων ατόμων, μπορεί να μειώσει σε μεγάλο βαθμό τον χρόνο εκκένωσης.

Abstract

The thesis presents and solves the Population Evacuation using Heterogeneous Fleet Problem (PEHFP), which concerns evacuation planning of certain pick-up locations and the transportation of the evacuees to safe shelters. To address PEHFP, a Mixed Integer Linear Programming (MILP) mathematical formulation and two heuristic algorithms have been developed. Initially, the heuristic algorithms are tested on a simple scenario of the evacuation problem, in which none of the evacuees faces mobility constraints. Then, the heuristic algorithm with the minimum evacuation time is applied to the more complex scenario, in which some of the evacuees are characterized by a physical disability. The selected heuristic algorithm is applied to a case study that focuses on developing an evacuation plan to deal with a forest fire in the Province of Teruel, Spain.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Description
BEP	Bus-based Evacuation Planning
H1	Heuristic Algorithm 1
H2	Heuristic Algorithm 2
IBEP	Integrated Bus Evacuation Problem
LRP	Location Routing Problems
MILP	Mixed Integer Linear Programming
MTVRP	Multi-Trip Vehicle Routing Problem
PEHFP	Population Evacuation using Heterogeneous Fleet Problem
VRPSF	Vehicle Routing Problem with Satellite Facilities

1. Introduction

1.1 Scope of the thesis

An emergency refers to an unexpected event that may cause damages which may not be dealt with the existing resources of the affected community. Disasters have both physical and human impacts such as human deaths and property losses and the level of such effects vary from disaster to disaster.

Although disasters can take several forms, they are classified to three major categories; natural, technological and social. The aforementioned types of disasters relate to a management cycle consisting of four phases. The first two phases, mitigation and preparedness, precede a disaster while the last two phases, response and recovery, occur post the disaster occurrence [1].

When a disaster strikes and sets people lives in danger, evacuation planning and transportation of population to safer places is of great importance so that human losses are avoided. Evacuation can be defined as the process in which affected people are relocated from threatened areas to safer places and consists a common and effective strategy to deal with emergency situations. The designing of evacuation plans is characterized as a mitigation measure while its execution takes place during the response phase.

The proposed approaches in evacuation planning vary and they have been developed under different aspects. Such aspects are traffic control strategies, identification of optimal evacuation routing plans in complex road networks, household behavior etc. [1]. Evacuation is a complex process consisting of various stages [2]. Due to the complexity of evacuation process, its effectiveness depends on several factors such as warning time, the traffic flow conditions etc. [3].

The necessity and importance of developing evacuation plans, has significantly increased due to the steep rise of the number of disasters during the last ten years. In particular, disasters caused economic damages of 1.4 trillion dollars in total and affected 1.7 billion people including 0.7 million fatalities. Roughly, 70% of deaths are caused by natural disasters such as earthquakes and tsunamis and 30% due to other types of disasters [5].

Despite the fact that a part of the population would use their own vehicles during the evacuation process, individual evacuation could lead to traffic congestion and impede operations. Moreover, due to the chaotic nature of a disaster it is hard for individuals to get access to reliable vehicles. Therefore, other forms of transportation such as public transportation resources are needed [4]. Authorities and evacuation planning managers are responsible for the development of evacuation plans, which aim to define optimal evacuation policies for the individuals/households from areas under risk and uncertainty [2].

This thesis deals with the development of a mathematical model and a solution method for the logistics problem under consideration; that is, planning the evacuation from certain pick-up locations and the transportation of the evacuees to shelters in the minimum evacuation time, subject to related constraints.

The contribution of the approach proposed in this thesis is two-fold. Firstly, a novel approach of evacuation planning is introduced, which deals effectively with some unique features of the problem compared to the existing literature. The main differences of the proposed approach consist in considering that a) vehicles are of different types and, thus, capacities (heterogeneous fleet), b) evacuees are also of different type in terms of mobility characteristics, c) vehicles are allowed to make multiple trips in order to collect evacuees, and d) each pick-up location can be visited at least once. Secondly, most of evacuation plans are, usually, car-based which means that they cannot satisfy the needs of transit-dependent population such as elderly or people with mobility issues. In this thesis, the needs of transit-dependent population are taken under consideration.

To address the problem we developed a novel Mixed Integer Linear Programming (MILP) and two heuristic algorithms for the evacuation of population, a part of which deals with a form of disability. By comparing the two proposed heuristics, in evacuation time terms, helped us produce an effective solution approach.

We applied this approach to a real case study and obtained very encouraging results.

1.2 Problem Description

This thesis introduces, models, and solves the Population Evacuation using Heterogeneous Fleet Problem (PEHFP), thereinafter called PEHFP. PEHFP concerns planning the evacuation of population from assembly points, and transporting the evacuees to safe shelters. The related case study concerns the evacuation of one or more villages of the Province of Teruel, in case of a major forest fire, and the transportation of the evacuees into one or more shelters.

In this thesis, we present a new mathematical model for the above problem along with all appropriate assumptions, available data and information related to this problem. The model for PEHFP seeks to determine the set of routes that minimize the total evacuation time. Among the possibly multiple solutions with the minimum evacuation time, the one with the minimum operational cost (total travel time) is selected. The proposed model includes multiple constraints that concern key operational issues, such as routing constraints, timing constraints, capacity constraints and other constraints. In order to develop the proposed model, we initially identify certain similarities and differences of the problem under consideration with the existing evacuation problems in the relevant literature. Accordingly, leveraging the related modeling work of the literature and considering fully the special characteristics of the problems under study, we developed the novel MILP.

To solve this problem we developed two heuristic algorithms. Moreover, we compared the algorithms in terms of the total evacuation time, and selected the most superior one to deal with the population evacuation planning for the case study of Province of Teruel.

This complex case study deals with the multipoint-to-point evacuation of Tramacastiel, Rubiales, and El Campillo, three small villages with 44, 31 and 40 citizens respectively, and the transportation of the evacuees to the Sports Hall "Los Planos" in Teruel. The latter is a province of Aragon, in the northeast part of Spain. The main types of emergencies in the area are floods and forest fires. In particular, the frequency of forest fires in Spain is one forest fire every 2.3 years. In addition, the fact that these three villages are located inside a forest, makes their evacuation planning an issue of great importance.

1.3 Literature Review

Prior to modeling and developing efficient algorithms to solve PEHFP, an extended literature review was carried out in order to identify similar problems. In [6], the author introduces a model specifically designed for Bus-based Evacuation Planning (BEP) along with two mathematical programming formulations, which are used to develop a heuristic algorithm. Using these models, the author analyzes the differences in the structural properties of optimal solutions between this problem and traditional vehicle routing problems. The objective in [6] is to transport evacuees from pickup points to shelters in a minimal amount of time by using a fleet of capacitated and homogenous buses. The BEP model has a key feature: it is assumed that the demand is predefined and fixed during the evacuation process. In an extended version of BEP, called robust bus evacuation problem, the demand is assumed to be known at later evacuation stages. In [7] the authors consider a set of estimates for the demand. The decision about whether buses need to be dispatched immediately (based on the estimates of demand) or to wait (until exact demand information is available) must be taken. Moreover, once a bus is routed, its plan cannot be changed. The model aims to minimize the maximum travel time of the buses.

In [8] the evacuation of a carless population under a no-notice scenario, in which buses perform a single trip without returning to pick up the rest of the carless population, is considered. All buses are initially located at a depot and the optimal departure time to demand points, so that the minimum travel time of buses is achieved, is discussed. In addition, travel times on network links are produced by a simulation model as a function of time.

In [9] a binary integer programming model is developed. The objective is to maximize the number of carless evacuated people within a certain time horizon. It is assumed that buses are located at the demand points at the beginning of evacuation and they have to return to the same demand point. The area under threat is divided by the zip code, and pick up points are assigned inside each zip code. Finally, the demand of each zip code is a certain percentage of the population within that zip code.

In [10] the authors present a simplified version of BEP. A Branch and Bound framework is used to identify lower and upper bounds of evacuation time. In [11] the authors focus on using public transport in emergency evacuation, aiming to maximize the number of evacuees. In the related work, a constraint of single trips of vehicles is considered and, therefore, it is assumed that not all evacuees may be transported. In [12] the authors propose a two index MILP to address the evacuation problem and its variants, and they developed a hybrid solution framework. They present extensive experimental results indicating that the proposed framework provides efficient solutions in reasonable computational times.

In [13] an emergency evacuation strategy is presented, in which buses serve a set of pick-up requests and delivery points using a certain routing strategy aiming to minimize the exposed casualty time. The delivery nodes of this case are of limited capacity and include both train stations and shelters.

Interested readers may also refer to [14], [15], [16] and [17] for research advances in the area of evacuation planning and emergency response. Recently, the work in [18] introduced the Integrated Bus Evacuation Problem (IBEP) that extends the simplified model of [10] by determining both the pick-up and the shelter points for evacuating a region using buses. To address this problem, the authors developed a branch-and-price strategy and compared its efficiency using a commercial IP solver. In general, the case of evacuation upon advance notice of threat bears similarities with the Vehicle Routing Problem with Satellite Facilities (VRPSF) studied in [19].

Other known problems that share common attributes with BEP include the Multi-Trip Vehicle Routing Problem (MTVRP) [20], [21], [22], [23], in which only one depot is available for vehicles to replenish their load between trips, and the VRP with Intermediate

Facilities or with inter-depot routes [24], [25], in which the vehicles may visit intermediate depots for load replenishment along their trips.

A related, but more general, class of problems includes the Location Routing Problems (LRP), in which the appropriate number and location of distribution centers are determined simultaneously while optimizing the routing costs to serve a set of customers. An extensive review of LRP is provided in [26], and recent interesting cases are addressed in [27] and [28]. In [29] the authors address an LRP that considers depots and vehicles with limited capacities, as well as fixed costs to establish a depot or to use a vehicle. In [30] a stochastic optimization model to minimize the total evacuation time is developed. However, the assumption that the demand is under uncertainty is not appropriate.

From the existing literature, the problems that are closer to the PEHFP evacuation case are those discussed in [6] and especially in [12]. Notable differences of the problem introduced in [12] with PEHFP include the following:

- In [12] all vehicles are assumed to be of equal capacity, though in PEHFP the vehicles are of *different types* and, thus, *capacities* (heterogeneous fleet)
- In PEHFP each vehicle is allowed to make *multiple trips* in order to collect evacuees. This is not the case in [12]
- In [12], when a vehicle visits a pick-up location it has to pick up the entire demand. In the PEHFP problem this constraint is relaxed. Consequently in [12] each pick-up location is visited exactly once, while in the PEHFP problem *each pick-up location is visited at least once.*
- In PEHFP, the evacuation of different types of evacuees, as far as their mobility problems concerns, is considered. This is not included in [12].

1.4 Thesis Structure

The rest of this thesis is organized as follows: In Chapter 2, the problem description, along with the mathematical formulation for PEHFP are presented. In Chapter 3 two heuristic algorithms developed to deal with the PEHFP are presented and discussed. Furthermore, computational results for comparing and evaluating the performance of the proposed algorithms are provided. In Chapter 4 the selected heuristic algorithm is applied to a more complex scenario, in which different types of evacuees, who need different treatment as far as their transportation is concerned, are considered. Chapter 5 describes the implementation of the proposed algorithm to the case study of Province of Teruel and the related computational results. Finally, the conclusions of this thesis are summarized in Chapter 6.

2. Mathematical formulation for the Population Evacuation using Heterogeneous Fleet Problem (PEHFP)

In this chapter a MILP mathematical formulation is proposed for the Population Evacuation using Heterogeneous Fleet Problem (PEHFP). Furthermore, the inputs of the mathematical formulation are also presented.

2.1 PEFHP Description

In PEHFP a fleet of vehicles with different characteristics (as far as their capacity and their capability of transferring different types of evacuees with mobility problems) has to pick up citizens from certain locations under threat and transport them to safe locations (shelters). In the problem under consideration, the shelter is a single facility of unlimited capacity and the objective is to determine the set of routes that minimize the total evacuation time; among the possibly multiple solutions with the minimum evacuation time, the one with the minimum operational cost (total time spent all resources) is selected. Note that the evacuation time is defined by the point in time the last evacuee arrives to a shelter, and the total operation time is the sum of the operation times of all vehicles (till they return to the ending depots).

As for the available vehicle fleet, it consists of different capacity as mentioned before and the capacity of its vehicle is known in advance. Moreover, all vehicles start and finish their routes from/to different locations (depots).

2.2 Mathematical Formulation

Let $\{t\}$ be the shelter of unlimited capacity (a single node) in which all the evacuees will be transferred to, and let $K = \{1, .., u\}$ be the set of available vehicles, assuming that u is their total number, each of capacity$Q_k, k \in K$. All vehicles start and finish their routes from/to different locations (depots), and thus we define two sets for the vehicle starting and ending locations - sets S and E, where $S = \{s^k | k \in K\}$ is the set of originating locations and $E = \{e^k \big| k \in K\}$ is the set of the ending locations. Each of these locations may be considered to be a single parking space. The locations are used in order to address the requirement to separate the total vehicle operation time from the evacuation time.

Let C be the set of all nodes representing the evacuee locations, hereafter called pick-up nodes. Additionally, let $d_i \in \mathbb{N}^+, i \in C$ be the number of evacuees waiting at pick-up node i. Moreover, let $V^k = \left\{v_1^k, v_2^k, \dots, v_{|V^k|}^k\right\}, k \in K$ be an ordered set containing the possible trips of each vehicle k, assuming that $\left|V^k\right| = \left\lceil\frac{\sum_{i\in C} di}{Q_k}\right\rceil, k \in K$, i.e. the maximum number of trips required to pick-up all evacuees by (utilizing the full capacity of) vehicle k. Let also $\bar{V} = \bigcup_{k\in K} V^k$, be the set of all possible trips. Note that we use ancillary parameter $Q^v = Q_k, v \in V^k, k \in K$ to denote that the capacity of the trips is equal to the capacity of the corresponding vehicle making the trip.

We formalize now the definition of directed graph $G(N, A)$, in which $N = \{t\} \cup S \cup E \cup C$ is the set of nodes, A is the arc set connecting the nodes of N and $\bar{A} = A_S \cup A_C \cup A_t \cup A_E$ is a set of triplets, with each triplet comprising an arc and a trip. Thus, let

- $A_S = \left\{\left(s^k, j, v_1^k\right) \middle| j \in C \cup \{e^k\}, k \in K\right\}$ be triplets containing the arcs starting from the originating location of each vehicle k, and the corresponding first trip. The first trip may be directed to a pick-up location, or to the ending location. The latter is used to model idle vehicles (if any)
- $A_C = \{(i, j, v) | i \in C, j \in (C\backslash\{i\}) \cup \{t\}, v \in V^k \backslash \{v_{|V^k|}^k\}, k \in K\}$ be triplets containing: a) arcs connecting each pick-up location i ∈ C to all other pick-up locations and to the shelter, and b) all trips besides the last trip that is dedicated to the return of the vehicle to its ending location (from the shelter or from the originating location for possible idle vehicles)
- $A_t = \{(t, j, v) | j \in C, v \in V^k \backslash \{v_1^k, v_{|V^k|}^k\}, k \in K\}$ be triplets containing arcs departing from the shelter to all pick-up locations by all trips besides the first and the last one
- $A_E = \{(t, e^k, v_{|V^k|}^k) | k \in K\}$ be triplets comprising of arcs connecting the shelter with the ending location of each vehicle by its last trip

Additionally, we define a set of pairs comprising trips related to certain nodes of the directed graph. Thus, we define the set $\bar{N} = N_S \cup N_C \cup N_E$, where:

- $N_S = \{\left(s^k, v_1^k\right) | k \in K\}$ contains only the first trip of each vehicle
- $N_C = \{(i, v) | i \in C \cup \{t\}, v \in V^k \backslash \{v_{|V^k|}^k\}, k \in K\}$ contains all trips, except the last trip of each vehicle, that may arrive to the pick-up location and to the shelter
- $N_E = \{(e^k, v) | v \in \{v_1^k, v_{|V^k|}^k\}, k \in K\}$ contains the last trip of each vehicle that arrives at the corresponding ending location. Note that an idle vehicle will be directed from the originating location to its ending location at its first trip, though a non-idle vehicle will make its last trip to its ending location.

Let $t_{ij}^{v}, (i,j,v) \in \bar{A}$ be the minimum travel time between nodes i and j by trip v. Let also:

- $w_i^v, (i,v) \in \bar{N}$ be the time that trip v arrives to node i
- $q_i^v, (i,v) \in \{\bar{N} | i \in C \cup E\}$ be the number of evacuees onboard the vehicle of trip v just before its arrival to node i
- $d_i^v, (i,v) \in \{\bar{N} | i \in C\}$ be the number of evacuees picked-up form node i during trip v
- $x_{ij}^{v}, (i,j,v) \in \bar{A}$ be assigned the value 1 if arc $(\mathrm{i},\mathrm{j}) \in \mathrm{A}$ is traversed by trip v, and 0 otherwise
- T_{evac} be the duration of the evacuation, i.e. the time span defined by the start of the evacuation until the time the last evacuee arrives to a shelter

Then the objective function of the PEHFP is defined as follows:

$$\min TC = T_{evac} + \frac{1}{L} \sum_{(i,j,v) \in \bar{A}} t_{ij}^{v} x_{ij}^{v} \tag{2.1}$$

where the second term is the total vehicle operation time (cost) and L ensures that the first term of (2.1) dominates lexicographically the second term: $L > \sum_{(i,j,v) \in \bar{A}} t_{ij}^{v}$. In particular, in case there are more than one optimal solutions, in terms of evacuation time, the one with less total “cost” is selected.

Optimization of (2.1) is subject to:

Routing constraints

$$\sum_{j \in N | (s^k, j, v_1^k) \in \bar{A}} x_{s^k j}^{v_1^k} = 1, \quad k \in K \tag{2.2}$$

$$\sum_{v \in \bar{V}, j \in N | (i,j,v) \in \bar{A}} x_{ij}^{v} \geq 1, \quad i \in C \tag{2.3}$$

$$\sum_{i \in N | (i,t,v_n^k) \in \bar{A}} x_{it}^{v_n^k} = \sum_{j \in N | (t,j,v_{n+1}^k) \in \bar{A}} x_{tj}^{v_{n+1}^k}, \quad n = 1, \dots, |V^k| - 1, k \in K \tag{2.4}$$

$$x_{s^k e^k}^{v_1^k} + \sum_{i \in N | (i,t,v) \in \bar{A}} x_{it}^{v} = 1, \quad v \in V^k \backslash \{v_{|V^k|}^k\}, k \in K \tag{2.5}$$

$$x_{s^k e^k}^{v_1^k} + x_{te^k}^{v_{|V^k|}^k} = 1, \quad k \in K \tag{2.6}$$

$$\sum_{i \in N | (i,h,v) \in \bar{A}} x_{ih}^v = \sum_{j \in N | (h,j,v) \in \bar{A}} x_{hj}^v, \quad h \in C, v \in V^k \backslash \{v_{|V^k|}^k\}, k \in K \tag{2.7}$$

Timing constraints

$$T_{evac} \geq w_t^v, \quad v \in \{v_{|V^k|-1}^k | k \in K\} \tag{2.8}$$

$$w_i^v + t_{ij}^v - B(1 - x_{ij}^v) \leq w_j^v, \quad (i,j,v) \in \bar{A}, i \in S \cup C \tag{2.9}$$

$$w_t^{v_n^k} + t_{tj}^{v_{n+1}^k} - B(1 - x_{tj}^{v_{n+1}^k}) \leq w_j^{v_{n+1}^k}, \quad \left(t, j, v_{n+1}^k\right) \in \bar{A}, \ \ n = 1, \dots, \left|V^k\right| - 1, \ \ k \in K, j \in C \cup \{e^k\} \tag{2.10}$$

$$0 \leq w_j^v \leq B \sum_{(i,j,v) \in \bar{A}} x_{ij}^v, \quad (j,v) \in \bar{N} \backslash N_S \tag{2.11}$$

Capacity constraints

$$q_i^v + d_i^v - B\left(1 - x_{ij}^v\right) \leq q_j^v, \quad (i,j,v) \in \bar{A}, i \in C \tag{2.12}$$

$$q_j^v \leq B\left(1 - x_{ij}^v\right), \quad (i,j,v) \in \bar{A}, i \in \{t\} \cup S, j \in C \tag{2.13}$$

$$\sum_{k \in K} q_{e^k}^{v_{|V^k|}^k} + \sum_{k \in K} w_{s^k}^{v_1^k} = 0 \tag{2.14}$$

$$0 \leq q_j^v \leq Q^v \sum_{(i,j,v) \in \bar{A}} x_{ij}^v, \quad (j,v) \in \bar{N} \backslash N_S \cup N_E \tag{2.15}$$

Other constraints

$$\sum_{v \in \bar{V} | (i,v) \in \bar{N}} d_i^v = d_i, \quad i \in C \tag{2.16}$$

$$d_i^v \in \mathbb{N}_0, \quad i \in C, (i,v) \in \bar{N} \tag{2.17}$$

$$x_{ij}^v \in \{0,1\}, \quad (i,j,v) \in \bar{A} \tag{2.18}$$

Regarding the **routing constraints**: Constraint (2.2) indicates that the first vehicle trip should depart from the related originating depot. Constraint (2.3) ensures that all pick-up locations should be visited at least once. Constraint (2.4) ensures that when a vehicle trip arrives to the shelter, the next vehicle trip should depart from it. Constraint (2.5) indicates that trips of non-idle vehicles should arrive at the shelter, or idle vehicles should head directly to the ending location. Constraint (2.6) ensures that the first or the last trip should arrive at the ending depot. Constraint (2.7) ensures that if a vehicle arrives to an evacuee location, it should also depart from the location within the same trip.

Regarding the **timing constraints**: Inequality (2.8) ensures that the evacuation time should be greater than the last visit to the shelter. Constraint (2.9) defines the change of the arriving time at each node within the same trip. Constraint (2.10) defines the change of arriving time between successive trips (through the shelter). Constraint (2.11) ensures that the time of arrival to any node, other that the starting location, will be greater or equal to zero, with $B \gg 1$, and, specifically, it will be equal to zero if the location is not visited. Constraint (2.14) denotes that the first trip of each vehicle starts at time equal to zero (and that each vehicle arrives at the ending location empty).

Regarding the **capacity constraints**: Constraint (2.12) defines the change of load for each trip, where $B \gg 1$. Constraint (2.13) ensures that every vehicle trip departs empty after a visit to the shelter and departs empty from the starting location. Constraint (2.14) denotes that each vehicle arrives at the ending position empty (and that the vehicle leaves its starting position at time equals to zero). Inequality (2.15) ensures that at any node, other than the starting locations, the number of evacuees aboard the vehicles will not exceed the vehicle's (trip) capacity nor will it be negative.

Regarding the **rest of the constraints**: Constraint (2.16) ensures that all evacuees should be picked-up from all pick-up locations by one or more vehicle trips. Finally, constraint (2.17) defines the nature of the variable that represents the number of evacuees picked up. Constraint (2.18) defines the binary nature of the arc variables at each trip v.

2.3 Inputs for PEHFP

For PEHFP, the formulation of the appropriate mathematical programming model assumes prior knowledge of the population at each village to be evacuated, including enabled and disabled citizens, since these citizen categories have different transportation needs. Specifically, for each village (pick-up point), problem inputs include the number of enabled evacuees, the number of disabled evacuees using wheelchairs who will be transported by vehicles with certain technical characteristics, and the number of disabled evacuees who need to be transported by ambulances. Note that for each village we assume that there will be a single pick-up point (assembly point) already been identified. This assumption does not present significant restrictions, since the intra village distances and travel times for citizens that require home pick ups are significantly shorter than the inter-village or the village to city distances and travel times. Note that the proposed approach does not address the case of citizens that evacuate using their own means of transport.

Further input data include the exact village locations (pick-up points), and the corresponding road network connecting all villages and the city (shelter). Note that the road network may offer the opportunity for more than one route between any two locations. Consequently, all network nodes and arcs should be provided, along with the corresponding distances and travel times.

Vehicle-related information includes the location of the starting point of each vehicle and the connecting road network, vehicle capacities and other characteristics. The latter concerns vehicles which may be used for transportation of enabled evacuees, wheel chair users, or citizens in need of special care (ambulance users).

Finally, additional input data include the exact shelter location. In the proposed approach we assume that the capacity of each shelter is unlimited.

3. Solution approach for PEHFP

Mathematical programming problems like the one presented in Section 2 for PEHFP are difficult to solve to optimality. In fact, such problems become harder to solve as complexity increases due to problem size. Consequently, trying to obtain an optimal solution for practical, complex, problems in reasonable time usually is not feasible. To overcome such difficulties, heuristic and other algorithms are developed in order to obtain efficient, near optimal solutions in reasonable time.

In this thesis two (related) heuristic approaches for PEHFP are presented. In both approaches, the algorithms schedule routes for the available vehicles in order to evacuate the population waiting at the pick-up locations and transport the evacuees to the shelter. The routing and pick up plan evacuate the entire population, minimizing operational time span, and respecting the capacities of the available vehicles, the traveling times between network nodes, and all other constraints. The heuristic algorithms are applied initially to an instance of the evacuation problem, in which none of the evacuees faces any mobility disabilities. The heuristics are compared in terms of total evacuation time and the one with the minimum evacuation time is then applied to the more complex case, in which some of the evacuees are characterized by a form of disability.

Both heuristics developed to solve this problem use the following input information.

- Number of available vehicles and the corresponding capacities
- Number of evacuees to be collected from each pick-up location (node)
- The network comprised by the vehicle starting points, the vehicle ending points, the shelter, and the pick-up locations, as well as all arcs feasibly connecting these nodes
- Travel times for all arcs in this network.

3.1 H1 for PEHFP for enabled population

For the first algorithm we create a list of the available vehicles ($List$) arranged in descending order with respect to their capacity. Thereafter, the vehicles in $List$ are routed simultaneously. In particular, the first vehicle of $List$ is routed to the node with the highest demand (always keeping a record of the corresponding traveling time), the second vehicle of $List$ to the second node with the highest demand (record traveling time), etc., until the $List$ is exhausted, or the demand of all pick-up locations is met. If the $List$ is exhausted and the demand is not met, the algorithm sorts the vehicle traveling times in ascending order and the vehicle with the minimum traveling time is routed after it completes its first pick up route. If the vehicle with minimum traveling time is at the shelter, it is routed to the node with the highest demand, otherwise it is routed to its nearest node.

The steps of the proposed algorithm to deal with PEHFP are the following:

Step 1. Sort the vehicles in descending order with respect to their initial capacity ($List$).

Step 2. Sort the demand of nodes in descending order ($Demand\_List$).

Step 3. Set the first vehicle k in $List$ as current vehicle (CV), delete it from $List$ and route it to the node with the highest demand.

Step 4. Update $CV's$ travel time ($Time\_List$), travel distance ($Total\_Distance$), capacity and update the demand of current node (CN).

Step 5. **If** $List$ is not exhausted and demand is not met go to **Step 2**

elseif demand is met

route the vehicles which are not at the shelter to the shelter, update their traveling time and their travel distance. Set T_{evac}= maximum element in $Time\_List$, set $Distance = sum(Total\_Distance)$ and **stop**.

elseif $List$ is exhausted and demand is not met

sort $Time\_List$ in ascending order and route the vehicle with the minimum travel time. Repeat steps 4-5.

end

In the following, the pseudo-code of the corresponding heuristic algorithm is given:

Step 1. Set $T_{evac} = 0, Time\_List = 0, Total\_Distance = 0$

Step 2. **While** $\sum_{c \in C} D_c > 0$

Step 3. **While** $List$ is not empty

3.1. Set the first vehicle $k \in K$ in $List$ as current vehicle ($CV = k$), delete it from $List$ and route it

3.2. Update $CV's$ travel time ($Time\_List$), travel distance ($Total\_Distance$), capacity and update the demand of current node CN

Step 4. Sort $Time\_List$ in ascending order

4.1 Set the vehicle $k \in K$ with the minimum travel time as current vehicle ($CV = k$) and route it to the nearest node if CV is at any demand point, otherwise route it to the node with the highest demand.

4.2 Repeat 3.2.

Step 5. Find the vehicles which are not at the shelter and route them to the shelter. Update their travel time ($Time\_List$) and their travel distance ($Total\_Distance$)

Step 6. Find $maxelement\{Time\_List\}$, set $T\_evac = maxelement\{Time\_List\}$ and $Distance = \sum_{k \in K} Total\_Distance(k)$

The detailed algorithm and the corresponding pseudo-code are given in Appendix A.

3.2 H2 for PEHFP for enabled population

In this second algorithm a list of vehicles ($List$) is uses also arranged in descending order with respect to their capacity. Thereafter, the first vehicle of $List$ is routed to the node with the highest demand (always keeping a record of the corresponding traveling time), and subsequently, if the capacity of the vehicle is not exhausted, it is routed to its nearest node. The process is continued in the same manner until its capacity is exhausted or the total demand is met. The vehicle returns to the depot. If the routing process of the first vehicle is completed and the total demand is not satisfied, then the algorithm continues with the second vehicle of the $List$ following the same process until the $List$ is empty. In case that the $List$ is empty and the demand is not satisfied, the algorithm identifies the vehicle that will return first to the shelter and continues performing the process described above until the total demand is met.

The corresponding steps of the second algorithm are the following:

Step 1. Sort the vehicles in descending order with respect to their initial capacity ($List$).

Step 2. If $List$ is not empty and demand is not met, set the first vehicle k in $List$ as current vehicle (CV) and delete it from $List$.

Elseif $List$ is empty

Sort $Time\_List$ in ascending order and set the vehicle with the minimum travel time as CV.

Step 3. Route CV from its starting point (or from the shelter) to the node with the highest demand and then to its nearest node until its residual capacity is equal to zero or the demand is met. Record its travel time ($Time\_List$), travel distance ($Total\_Distance$), capacity and update the demand of each node that CV services.

Step 4. If demand is not met, repeat steps 2-4

Else

Route the vehicles which are not at the shelter to the shelter. Update their travel time and their travel distance. Set T_{evac}= maximum element in $Time\_List$, set $Distance = sum(Total\_Distance)$ and **stop**.

end

Accordingly, the pseudo-code for the second algorithm is given below:

Step 1. Set $T_{evac} = 0, Time\_List = 0, Total\_Distance = 0$

Step 2. **While** $\sum_{c \in C} D_c > 0$

Step 3. **If** $List$ is not empty

3.1 Set the first vehicle $k \in K$ in $List$ as current vehicle ($CV = k$) and delete it from $List$

3.2 Route CV from its starting point (or from the shelter) to the node with the highest demand and then to its nearest node until its capacity is equal to zero or the demand is met. Record its travel time ($Time\_List$) , travel distance ($Total\_Distance$) , capacity and update the demand of each node that CV services.

3.3 **Else**

3.4 Sort the $Arrival\_List$ in ascending order and set the first vehicle k in $Arrival\_List$ as current vehicle($CV = k$). Route it according to step 3.2

Step 4. Find the vehicles which are not at the shelter and route them to the shelter. Update their travel time ($Time\_List$) and their travel distance ($Total\_Distance$)

Step 5. Find maxelement$\{Time_List\}$, set $T\_evac = maxelement\{Time\_List\}$ and $Distance = \sum_{k \in K} Total\_Distance(k)$

The detailed version of the second algorithm and the corresponding pseudo-code are given in Appendix B.

4. Comparison between H1 and H2 and application to the general problem

In order to evaluate and validate the proposed heuristic algorithms, a set of sample evacuation problems is generated and solved. To this end, a problem generator has been developed.

Both heuristics have been tested under two different scenarios that vary key parameters. In the first one, the number of pick up nodes is fixed and the number of vehicles varies, while in the second scenario, the number of nodes increases and the number of vehicles is fixed. For each vehicle-node combination, 100 problems have been generated and solved by both heuristics. Subsequently, the mean evacuation time is computed for each heuristic per vehicle-node combination.

The problem generator has been provided with the following inputs:

- Number of vehicles
- Number of nodes
- Vehicle capacities
- Node demand
- Travel times from each vehicle starting point to each node
- Travel times between nodes
- Travel times from shelter to each node
- Coordinates of each vehicle's starting point
- Coordinates of shelter
- Coordinates of nodes
- Average speed of each vehicle

Additionally, note that the following probability distributions have been used for the generated data:

- Vehicle capacity is generated from a Normal distribution $N(10,4)$.
- Node demand is generated from a Normal distribution $N(25,25)$.
- The coordinates of the shelter, nodes and vehicle starting points are generated from a Uniform distribution $U(0,100)$.
- Distances are calculated using the Euclidean norm.
- For each problem, the mean speed of all vehicles is generated from a Uniform distribution $U(45,55)$.

- The travel times are calculated as $\frac{s}{v}$, where s is the distance between nodes in km and v is the mean speed assumed for vehicles, generated from a Uniform distribution $U(45{,}55)$ km/h for each vehicle.

4.1. Scenario 1:Fixed number of operating vehicles

Under the first scenario, the number of vehicles v is fixed while the number of nodes i varies. Initially, for $i = 1$ node and $v = 5$ vehicles the generator creates 100 different problems. Both heuristics 1 and 2 are used to calculate the evacuation time and the total distance covered by all vehicles for each problem. Subsequently, the corresponding mean values are computed. The same process is followed for $i = 2, 3, \ldots, 15$.

The results obtained are presented in Table 4.1, and in Figures 4.1, 4.2, 4.3 and 4.4. In Figure 4.1, the mean evacuation time for both algorithms is shown with respect to the number of nodes (increasing as expected). Similarly, in Figure 4.2, the mean total distance covered by all operating vehicles during the evacuation problem increases with the number of nodes for both heuristics. Due to its nature, H1 utilizes more vehicles for meeting the demand. Thus, H1 is better, in terms of evacuation time, in comparison to H2, which is better in terms of total covered distance, since it uses fewer vehicles.

Table 4.1. *Mean total evacuation times, mean total distances and percentage differences for heuristic algorithms 1 and 2 for v = 5 vehicles and i nodes*

i	Mean T_{evac1} *in min*	Mean T_{evac2} *in min*	Mean *Total_Dist*$_1$ *in km*	Mean *Total_Dist*$_2$ *in km*	Percentage Difference of T_{evac} (H2-H1)	Percentage Difference of *Total_Dist* (H2-H1)
1	151.7	151.7	215.47	215.4	0.00%	0.00%
2	184.6	211.5	557.68	439.8	12.70%	-26.79%
3	221.1	235.8	652.62	636.9	6.24%	-2.45%
4	294.4	322.7	894.5	852.2	8.77%	-4.95%
5	333.5	365.4	1090.1	1037.4	8.73%	-5.08%
6	399.7	425.1	1335.7	1289.5	5.98%	-3.57%
7	442.5	476.2	1525.5	1460.2	7.07%	-4.47%
8	478.6	511.7	1681.4	1625.5	6.47%	-3.43%
9	523.9	564.2	1852.5	1782.0	7.15%	-3.95%
10	587.2	624.3	2075.2	1999.4	5.93%	-3.78%
11	631.3	665.4	2260.5	2166.7	5.12%	-4.32%
12	669.6	718.6	2445.6	2366.1	6.81%	-3.36%
13	720.2	769.2	2644.1	2545.3	6.35%	-3.88%
14	753.9	804.5	2781.6	2674.2	6.28%	-4.01%
15	814.2	855.9	3020.9	2917.2	4.86%	-3.55%

Moreover, in Figure 4.3, the percentage difference for the mean evacuation time between the two heuristics is shown. The percentage difference of the mean evacuation time decreases. This may be attributed to the fact that as the number of nodes increases the population to be evacuated also increases and, consequently, in both algorithms more routes are needed in order to evacuate the entire population. Since more routes are executed for meeting the demand, the evacuation time increases in both algorithms and their percentage difference, in term of total evacuation time, reduces. Thus the predominance of H1fades out as the number of demand points increases.

In Figure 4.4, the corresponding percentage difference of the mean total distance is given which is stabilized. Since the same number of vehicles serves more nodes, the covered distance in both algorithms is increased and their percentage difference is stabilized.

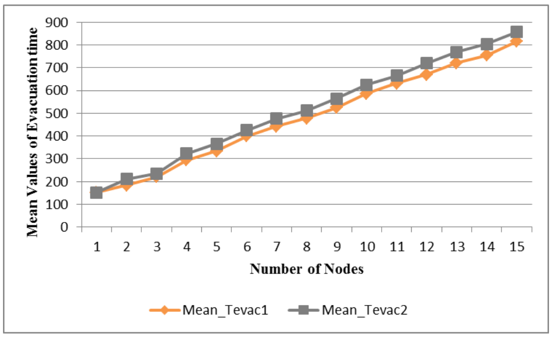

Figure 4.1 *Average evacuation time for heuristics 1 &2 for v = 5 vehicles and i= 1, …, 15 nodes*

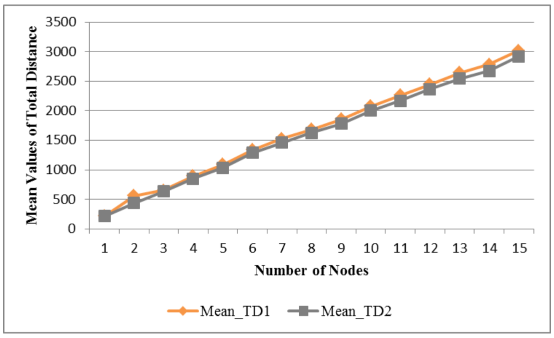

Figure 4.2. *Mean Total Distance Of Heuristics 1&2 for v = 5 vehicles and i= 1, …, 15 nodes*

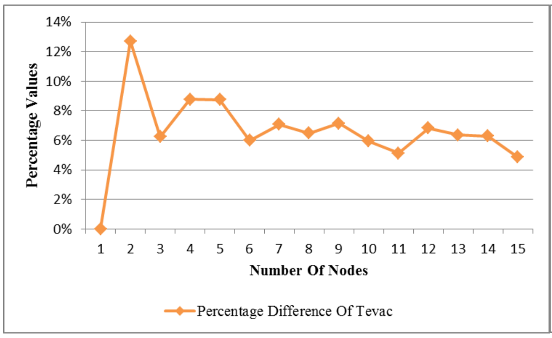

Figure 4.3 *Percentage difference of average evacuation time for heuristic 2 vs. heuristic 1 (v = 5 vehicles and i= 1, …, 15 nodes)*

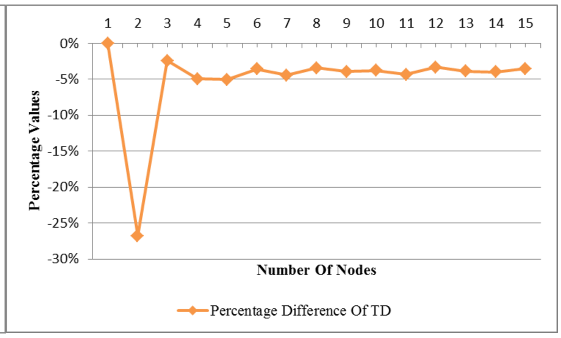

Figure 4.4 *Percentage difference of average total distance for heuristic 2 vs. heuristic 1(v = 5 vehicles and i= 1, …, 15 nodes)*

4.2. Scenario 2: Fixed number of pick up nodes

In the second scenario, the number of nodes is maintained constant while the number of vehicles v varies. Initially, for $v = 1$ vehicles and $i = 5$ nodes, 100 different problems are generated and heuristics 1 and 2 are used to determine the evacuation time and the total distance for each problem. The mean values are computed and recorded for each heuristic. This process is repeated for $i = 5$ and $v = 2, 3, \ldots, 15$.

Table 4.2 includes the results obtained. These results are also presented in Figures 4.5, 4.6, 4.7 and 4.8. In Figure 4.5 the mean evacuation times for both algorithms are presented with respect to the number of vehicles used. As expected, the evacuation time reduces with the number of operating vehicles since the same number of pick-up locations are served by more vehicles. Similarly, in Figure 4.6, the mean total distance for both algorithms reduces with the number of vehicles for both heuristics.

Table 4.2 *Mean total evacuation times, mean total distances and their percentage difference for heuristic algorithms 1 and 2 for k vehicles and 5 nodes*

k	**Mean T_{evac1} in min**	**Mean T_{evac2} in min**	**Mean $Total\_Dist_1$ in km**	**Mean $Total\_Dist_2$ in km**	**Percentage Difference of T_{evac} (H2-H1)**	**Percentage Difference of $Total\_Dist$ (H2-H1)**
1	1332.337	1332.337	1057.9542	1057.954	0.00%	0,00%
2	676.618	690.912	1051,.615	1034.285	2.07%	-1,68%
3	489.120	512.858	1077.149	1050.720	4.63%	-2,52%
4	391.300	416.897	1069.378	1035.949	6.14%	-3,23%
5	337.887	368.899	1081.782	1049.800	8.41%	-3,05%
6	308.832	349.316	1121.711	1065.205	11.59%	-5,30%
7	275.471	309.127	1084.043	1023.118	10.89%	-5,95%
8	241.683	272.031	1062.912	1029.039	11.16%	-3,29%
9	224.658	261.642	1079.255	1037.393	14.14%	-40,35%
10	202.767	248.686	1066.8726	1006.079	18.46%	-6,04%
11	194.623	245.510	1126.600	978.794	20.73%	-15,10%
12	195.703	244.126	1085.523	992.689	19.84%	-9,35%
13	201.397	252.357	1060.189	1010.959	20.19%	-0,49%
14	195.217	250.606	1061.407	1003.759	22.10%	-5,74%
15	199.372	248.468	1036.897	957.072	19.76%	-8,34%

Finally, in Figure 4.7, the percentage difference of the mean evacuation time between the two heuristics is presented, while in Figure 4.8, the corresponding percentage difference of the mean total distance is provided. Due to its nature, H1 utilizes more vehicles for meeting the demand in comparison to H2. Therefore, as the number of the available vehicles increase, H1 uses more vehicles and manages to complete the evacuation process earlier than H2. Consequently, the percentage difference of the mean evacuation time between the two heuristics increases.

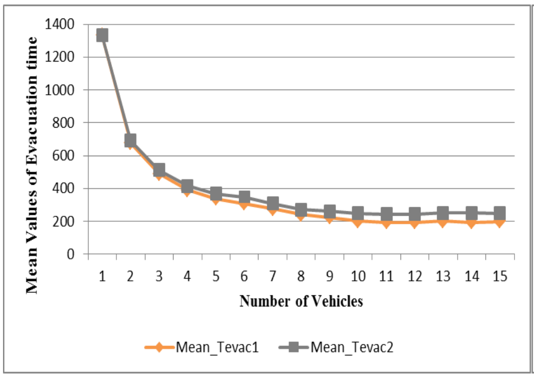

Figure 4.5 *Mean Values Of Heuristics 1&2 for v = 1, …, 15 vehicles and i = 5 nodes*

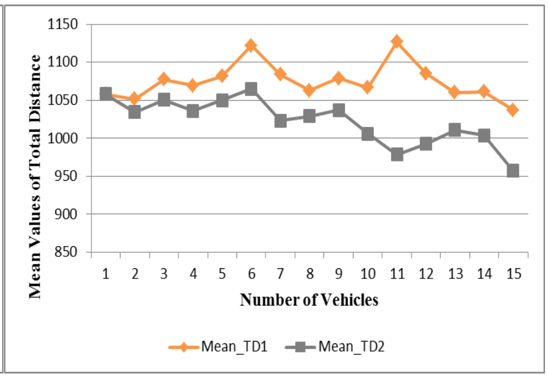

Figure 4.6 *Mean Values Of Total Distance Of Heuristics 1&2 for v = 1, …, 15 vehicles and i = 5 nodes*

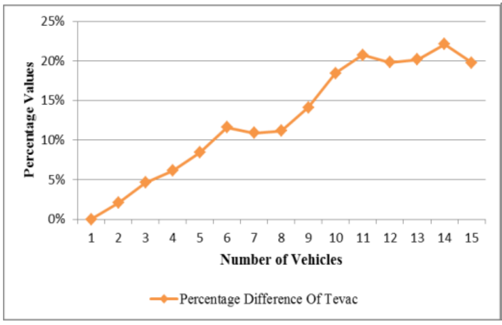

Figure 4.7 *Percentage difference of average evacuation time for heuristic 2 vs. heuristic 1 (v = 1, …, 15 vehicles and i = 5 nodes)*

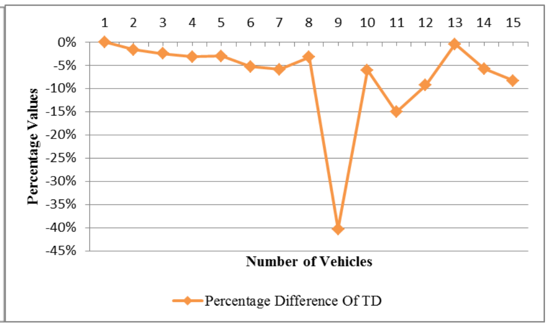

Figure 4.8 *Percentage difference of average total distance for heuristic 2 vs. heuristic 1 (v = 1, …, 15 vehicles and i = 5 nodes)*

4.3. Comparison of the two heuristics

Considering Figures 4.2 and 4.6, it is deduced that heuristic algorithm 1 (H1) is superior in terms of T_{evac}than heuristic algorithm 2 (H2). The reason lies in the fact that H2 is greedy in terms of distance and attempts to fully load each vehicle during each trip. In H1 the vehicles operate in parallel. Thus, H1 tends to minimize the total traveling time. Contrarily, as shown in Figures 4.3 and 4.7, H2 is superior with respect to the total distance covered by all vehicles, since less vehicles operate in parallel. To confirm this statement, the following example is presented. In this example let $v = 7$ vehicles, $i = 5$ nodes.

Input Data:

Table 4.3 *Travel time between nodes*

From / To	**1**	**2**	**3**	**4**	**5**
1	0	72.1139	41.4709	22.9490	70.1249
2	72.1139	0	73.2866	54.0694	29.9720
3	41.4709	73.2866	0	29.1830	54.3531
4	22.9490	54.0694	29.1830	0	47.4931
5	70.1249	29.9720	54.3531	47.4931	0

Table 4.4 *Travel time from shelter to each node*

From / To	**1**	**2**	**3**	**4**	**5**
Shelter	45.2821	57.0145	77.7297	48.7042	74.067

Table 4.5 *Travel time from each vehicle's starting point to each node (Note: S^k indicates the starting point of vehicle k)*

To \ From	1	2	3	4	5
S^1	24.2659	89.7237	64.2596	46.6567	92.6210
S^2	49.0665	120.2944	78.1593	71.2473	118.7385
S^3	59.0027	32.7607	78.6444	51.0082	56.3768
S^4	59.1934	16.3588	56.9286	39.1825	20.4996
S^5	50.2553	95.8922	91.6881	68.1829	107.6253
S^6	36.2560	41.8239	59.3286	30.3415	53.5540
S^7	14.4474	67.2251	27.3765	13.2964	59.8673

Table 4.6 *Node demand*

i	1	2	3	4	5
Demand D_i	22	24	27	17	27

Table 4.7 *Vehicle's capacity*

k	1	2	3	4	5	6	7
Capacity Q_k	10	14	5	14	11	12	7

The resulting routes for the evacuation problem for both heuristics are presented in the following Table 4.8:

Table 4.8 *Routes of heuristics 1&2*

Heuristic 1	Heuristic2
Route1= S^2-3-shelter with vehicle 2	Route1= S^2-3-shelter with vehicle 2
Route2= S^4-5-shelter with vehicle 4	Route2= S^4-5-shelter with vehicle 4
Route3= S^6-2-shelter with vehicle 6	Route3= S^6-2-shelter with vehicle 6
Route4= S^5-1-shelter with vehicle 5	Route4= S^5-1-shelter with vehicle 5
Route5= S^1-4-shelter with vehicle 1	Route5= S^1-4-shelter with vehicle 1
Route6= S^7-3-shelter with vehicle 7	Route6= S^7-3-shelter with vehicle 7
Route7= S^3-5-shelter with vehicle 3	Route7= S^3-5-shelter with vehicle 3
Route8=shelter-2-shelter with vehicle 4	Route8=shelter-2-5-shelter with vehicle 4
Route9=shelter-1-shelter with vehicle 1	Route9=shelter-1-shelter with vehicle 1
Route10=shelter-5-shelter with vehicle 5	Route10=shelter-4-1-3-shelter with vehicle 5
Route11=shelter-4-shelter with vehicle 6	Route11=shelter-5-3-shelter with vehicle 6
Route12=shelter-3-shelter with vehicle 7	
Route13=shelter-1-shelter with vehicle 3	
T_{evac1}= 260.6 min	**T_{evac2}= 305min**

The first trips for all vehicles are identical for both heuristics. Let's examine further the second step of each algorithm. In both algorithms, vehicle 4 is the first vehicle that arrives at the shelter after its first trip at time $t^4 = l_{S^4,5} + l_{5,shelter} = 20.5 + 74.1 = 94.6\, min$, where $L_{i,j}$ is the travel time between nodes i and j and t^i is the traveling time of vehicle i.

According to H1, vehicle 4 leaves the shelter at time $t^4 = 94.566\, min$, visits node 2, picks up the residual demand (12 evacuees) and returns to the shelter although it has some residual capacity (2 seats). Vehicle 4 returns to the shelter at time $t^4 = t^4 + l_{shelter,2} + l_{2,shelter} = 208.6\, min$. Note that, at time $t^5 = l_{S^5,1} + l_{1,shelter} = 95.5\, min$, vehicle 5 leaves the shelter, visits node 5, picks up the residual demand (8 evacuees) and returns to the shelter to drop off the evacuees at time $t^5 = t^5 + l_{shelter,5} + l_{5,shelter} = 243.7\, min$. Thus, under H1, node 5 is completely evacuated at time $t^5 = 243.7\, min$.

According to H2**,** vehicle 4 leaves the shelter at time $t^4 = 94.566\, min$, visits node 2 and picks up the residual demand (12 evacuees). However, there is still free space onboard vehicle 4 (2 seats). Due to this reason, the vehicle visits the nearest node, which is node 5, to collect more evacuees. At node 5, it picks up 2 evacuees (residual capacity) and then returns to shelter at time $t^4 = t^4 + l_{shelter,2} + l_{2,5} + l_{5,shelter} = 255.6\, min$. But not all evacuees are picked up from node 5 and thus another vehicle needs to visit node 5 to collect the remaining evacuees. Consequently, the evacuation time of node 5 when H2 is applied, is higher than the corresponding evacuation time with H1. Therefore, H1 manages to completely evacuate node 5 faster than H2.

Note that this pattern is repeated through the following trips planned by the algorithms, and hence, the accumulated difference of evacuation time increases. Consequently, H1 manages to complete the evacuation process earlier than H2. On the other hand, under algorithm H2, less distance is covered to complete the evacuation process. In particular, vehicle 4 returns to the shelter after its first trip and then, in both algorithms, it is routed to the node with the highest demand. According to H2, vehicle 4 is routed to node 2. After it serves node 2, it is routed to the nearest node because its remaining capacity is greater than zero. In H1, vehicle 4 is routed to node 2 and, although its residual capacity is greater than zero, it is not routed to the next node because the vehicle with the minimum travel time (vehicle 1) has priority to be routed. In that way, H1 uses more vehicles for meeting the demand and the total distance covered by the vehicles of H1 is greater than the total distance of H2.

4.4. PEHFP for a population that comprises enabled and disabled evacuees

Based on the results of Section 4.3, the first heuristic algorithm (H1), which performs better in terms of evacuation time, is selected to be implemented in the case of multiple evacuee types.

In particular, we deal with three evacuee types in total. The type of evacuee depends on her/his mobility status. The first types of evacuees, enabled evacuees, are those who have been considered in the previous section. The second type of evacuees concerns people with partial disability who use a wheel chair. The third type concerns citizens with more severe disability who need to be transported on stretchers. In other words, in contrast with the enabled evacuees, due to their mobility problems, the last two categories need special transportation treatment.

Due to evacuees' special needs for transportation, vehicles with special characteristics are required, contrarily to the previous approach where vehicles pick up only enabled evacuees and they do not need to be specially equipped. Specifically, partially disabled evacuees need to be transferred by vehicles that are equipped with ramps so that wheel chairs can easily get onboard. Totally disabled evacuees can only be transferred by ambulances. Furthermore, we assume that both partially and totally disabled evacuees are accompanied by a relative or a doctor/nurse. The aforementioned difference is critical for the evacuation problem since the fleet to be used needs to include specific types of vehicles. Note that a vehicle that can transport partially disabled evacuees can also transport enabled evacuees, while a vehicle that can transfer enabled evacuees cannot necessarily transfer partially or totally disabled evacuees. In addition, a vehicle that can transfer totally disabled evacuees may transfer partially disabled evacuees.

Additionally, it is worth mentioning that another critical difference with the problem so far presented concerns the service time. In particular, the service time of a partially or totally disabled evacuee is higher compared to the service time for an enabled evacuee, a fact that clearly affects the total evacuation time since the pick-up and drop-off processes last longer. Taking into account the aforementioned constraint along with the fact that a disabled evacuee may need immediate medical help, indicates that disabled citizens should be evacuated first. This decision affects the solution approach, since vehicles that can transport disabled evacuees should be routed with a priority.

Consequently, we have divided the problem in two parts according to the aforementioned constraints concerning the order of citizens' evacuation. The first part deals with partially and totally disabled evacuees, and the second part deals with partially disabled and enabled evacuees.

In the first part of H1, a list of available ambulances arranged in descending order with respect to their capacity for the partially disabled, is initially created. Note that in case none of the ambulances is adapted for partially disabled evacuees, the ambulances are sorted in descending order with respect to their capacity for enabled evacuees. Thereafter, the ambulances in $List$ are routed simultaneously. In particular, the first ambulance of $List$ is routed to the node with the highest demand for totally disabled (always keeping a record of the corresponding traveling time), the second ambulance of $List$ to the second node with the highest demand for totally disabled (record traveling time), etc., until the $List$ is empty, or the demand of pick-up locations for totally disabled is met. If the $List$ is empty and the demand of totally disabled is not met, the algorithm sorts the traveling times in ascending order and the ambulance with the minimum traveling time is routed after it completes its first trip to the node with the highest demand for totally disabled.

According to the second part of H1, a list of the available vehicles ($List$) is initially created as follow: in case that there are vehicles adapted for partially disabled, they are sorted in descending order with respect to their capacity for partially disabled and then, the rest of the vehicles are sorted in descending order with respect to their capacity for enabled. Thereafter, the first vehicle of $List$ is routed to the node with the highest demand (If the vehicle is adapted for partially disabled, it is routed to the node with the highest demand for partially disabled, otherwise it is routed to the node with the highest demand for enabled) and always keeping a record of the corresponding traveling time, the second vehicle of $List$ to the second node with the highest demand (record traveling time) etc. This process is repeated until $List$ is exhausted or the demand for both partially disabled and enabled is met. If the $List$ is empty and the demand is not met, the algorithm sorts the vehicle travel times in ascending order and the vehicle with the minimum travel time is selected to be routed. In case that the vehicle with the minimum travel time is adapted for partially disabled, the demand for them is not met and the current node of this vehicle is any demand point, then, it is routed to its nearest node with nonzero demand for partially disabled, otherwise, if it is at the shelter, it is routed to the node with the highest demand for partially disabled. In case that the vehicle with the minimum travel time cannot serve partially disabled and its current node is any demand point, then, it is routed to its nearest node with nonzero demand for enabled, otherwise, it is routed to the node with the highest demand for enabled.

The steps of the proposed algorithm to deal with PEHFP for enabled and disabled population are the following:

Step 1. **If** the demand for totally disabled is higher than zero execute steps 2-5,

Else

got to step 6.

Step 2. Sort the ambulances in descending order with respect to their initial capacity ($List$).

Step 3. **If** $List$ is not empty

set the first vehicle in $List$ as current vehicle and delete it from $List$,

Else

sort the $Time\_List$ in ascending order and set as current vehicle (CV) the vehicle with the minimum travel time.

Step 4. **If** CV is at its starting point or at the shelter, sort the demand of nodes for totally disabled in descending order ($Demand\_List^3$) and route it to the node with the highest demand for totally disabled.

Else

route it to the nearest node with nonzero demand for partially disabled.

Step 5. Update $CV's$ travel time ($Time\_List$), travel distance ($Total\_Distance$) , capacity and update the demand of current node(CN). Go to Step 1.

Step 6. Sort the vehicles for enabled and partially disabled in descending order with respect to their initial capacity ($List$).

Step 7. **If** the demand for partially disabled **OR** the demand for enabled is greater than zero execute steps 8-11,

Else

got to step 12.

Step 8.

- **Case 1:** In case that $List$ is not empty, set the first vehicle in $List$ as current vehicle and delete it from $List$.

- **Case 2:** Otherwise, sort the $Time\_List$ in ascending order and set as current vehicle the one with the minimum travel time. Note that if the demand for enabled is met, only the travel times of vehicles adapted for partially disabled are sorted.

Step 9.

- **Case 1:** In case that CV is at its starting point or at the shelter, sort the demand of nodes in descending order with respect to their demand for partially disabled ($Demand\_List^2$) or with respect to their demand for enabled ($Demand\_List^1$) (It depends on either CV is adapted for partially disabled or not). Set as current node (CN) the node i with demand $D_i = Demand\_List^2(1)$(or $D_i = Demand\_List^1(1)$) and route CV to CN.
- **Case 2:** In case that CV is at any demand point, route it to its nearest node with nonzero demand for enabled or for partially disabled (It depends on either CV is adapted for partially disabled or not).

Step 10. Update $CV's$ travel time ($Time\_List$), travel distance ($Total\_Distance$), capacity and update the demand of current node(CN).

Step 11. In case that there are vehicles adapted for partially disabled and the demand for partially disabled is met, convert their remaining capacity for partially disabled into capacity for enabled. Go to step 7.

Step 12. Route the vehicles which are not at the shelter to the shelter, update their traveling time and their travel distance. Set T_{evac}= maximum element in $Time\_List$, set $Distance = sum(Total\_Distance)$ and **stop**.

In the following, a pseudo-code of the above heuristic algorithm is given:

Step 1. Set $T_{evac} = 0, Time\_List = 0, Total\_Distance = 0, List = \emptyset$

Step 2. **If** $\sum_{k\in K} str_k > 0$ **AND** $\sum_{c\in C} D_c^3 > 0$

2.1. Sort the vehicles with descending order with respect to their capacity for partially disabled

2.2. **Else**

2.3. Sort the vehicles in descending order with respect to their capacity for enabled

Step 3. Insert the sorted vehicles into $List$

Step 4. **While** $\sum_{c\in C} D_c^3 > 0$

Step 5. **While** $List$ is not empty

5.1. Set the first vehicle $k \in K$ in $List$ as current vehicle ($CV = k$), delete it from $List$. In case that it is at its starting point or at the shelter and it is a vehicle adapted for partially disabled, among the nodes with the greatest demand for totally disabled route it to the node with the highest demand for partially disabled. If it is not a vehicle adapted for partially disabled route it to the node with the highest demand for totally disabled. In case that CV is at any demand point route it to its nearest node.

5.2. Update $CV's$ travel time ($Time\_List$), travel distance ($Total\_Distance$), capacity and update the demand of current node CN.

Step 6. Sort $Time\_List$ in ascending order

Step 7. Set the vehicle $k \in K$ with the minimum travel time as current vehicle ($CV = k$), route it according to step 5.1 and repeat step 5.2.

Step 8. **While** $\sum_{p=1}^{2} \sum_{c \in C} D_c^p > 0$

Step 9. **If** $List$ is not empty set the first vehicle $k \in K$ in $List$ as current vehicle ($CV = k$)

9.1 **If** $r_{CV}^p = 1$ **OR** $node = \{t\}$

9.2 **If** $Dis\_veh \neq 0$ **AND** $x_{cv} = 1$ **AND** $\sum_{c \in C} D_c^2 > 0$

9.3 Set as CN the node with the highest demand for partially disabled

9.4 **Elseif** ($Dis\_veh \neq 0$ **AND** $x_{cv} = 0$) **OR** $Dis\_veh = 0$ **OR** ($Dis\_veh \neq 0$ **AND** $\sum_{c \in C} D_c^2 = 0$)

9.5 Set as CN the node with the highest demand for enabled

9.6 **Elseif** $node \neq \{t\}$

9.7 **If** $x_{cv} = 1$ **AND** $\sum_{c \in C} D_c^2 > 0$ **AND** $Q_{CV}^2 > 0$

9.8 Route vehicle CV to node l with $\min\{L_{CNl}, l \in C \setminus \{CN\}\}$ and demand $D_l^2 \neq 0$.

9.9 **Elseif** $x_{cv} = 0$ **OR** ($x_{cv} = 1$ **AND**($\sum_{c \in C} D_c^2 = 0$ **OR** $Q_{CV}^2 = 0$))

9.10 Route vehicle CV to node l with $\min\{L_{CNl}, l \in C \setminus \{CN\}\}$ and demand $D_l^1 \neq 0$.

Step 10. Repeat step 5.2

Step 11. **Elseif** $List$ is empty, Repeat step 6

Step 12. **If** $\sum_{c \in C} D_c^2 = 0$

12.1 Set as CV the vehicle with the minimum travel time and the highest capacity for enabled

12.2 **Elseif** $\sum_{c \in C} D_c^1 = 0$

12.3 Set as CV the vehicle with the minimum travel time and the highest capacity for partially disabled

12.4 **Elseif** $\sum_{c \in C} D_c^2 > 0$ **AND** $\sum_{c \in C} D_c^1 > 0$

12.5 Set as CV the first vehicle in $Time\_List$

Step 13. Find the vehicles which are not at the shelter and route them to the shelter. Update their travel time ($Time\_List$) and their travel distance ($Total\_Distance$)

Step 14. Find maxelement$\{Time_List\}$, set $T\_evac = maxelement\{Time\_List\}$ and $Distance = \sum_{k\in K} Total\_Distance(k)$

The corresponding pseudo-code for enabled and disabled population is given in Appendix C.

5. Case Study

The case study considered in this thesis focuses on a forest fire in the Province of Teruel, which evolves dynamically. The aim is to develop an appropriate population evacuation plan for the Province of Teruel, by using heuristic algorithm H1, enhanced to address the evacuee (and vehicle) types. In particular, we focus in obtaining near-optimal solutions for three different case scenarios.

- The first one (Scenario A) concerns the evacuation of the small village of Tramacastiel at the province of Teruel and the transportation of evacuees to Villel (point-to- point PEHFP).
- The second scenario (Scenario B) deals with the evacuation of Tramacastiel, Rubiales and El Campillo (small villages) and the transportation of the evacuees to a safe shelter at the city of Teruel (multipoint-to-point PEHFP).
- Finally, the third scenario (Scenario C) deals with evacuating the three aforementioned villages in case of a forest fire that evolves according to weather condition changes. More specifically, under Scenario C, the fire initially threatens the village of Tramacastiel and its evacuation is ordered by the local authorities. Later the fire evolves and threatens both the villages of Rubiales and El Campillo. An order to evacuate these villages is then given by local authorities.

In order to apply the heuristic algorithm presented in Section 4.4 to the aforementioned evacuation scenarios, three categories of data need to be provided: (a) Evacuees and demand, (b) Network, (c) Available Vehicles.

Regarding the evacuees to be picked up, the total population of each village (Tramacastiel, Rubiales, El Campillo) should be provided. Additionally, in order to use the appropriate vehicles for the transportation of the evacuees, for each village the number of enabled evacuees, the number of wheel chair users and the number of evacuees to be transported by ambulances is required.

Regarding the nodes of the network, the pick-up locations (villages), and their exact location should be provided. The same holds for the starting and ending locations of each vehicle, as well as of the shelter. For the network arcs, input data required include the distances a) between the originating points of available vehicles and the pick-up locations, b) between each pick-up location and the shelter, c) between the pick-up locations, d) between the shelter and the ending locations of each vehicle. Note that when possible, any alternative arcs should be also be provided.

For each of the available vehicles input information should include the capacity per type of evacuee (enabled, wheel chair users, totally disabled).

All the aforementioned necessary input data are provided in Appendix D.

It is also important to note that the total evacuation time depends on the circumstances under which the physical disaster evolves. For instance, there may significant traffic along the road network used by the proposed solution, resulting in an increase of the total evacuation time. If one of the operating vehicles becomes incapacitated (for any reason), then the load and the exact location of the vehicle should be known in order to decide on how to overcome such a difficulty, e.g. either by sending another vehicle to take over the mission of the failed one, or to reach the location of the accident and transfer its load. To deal with such unplanned situations, redundant vehicles should be also available.

5.1 Scenario A: PEHFP solution for point-to point evacuation

The evacuation of Tramacastiel and the transportation of all types of evacuees to Villel is a small scale evacuation problem. Note that Villel can be considered as a safe assembly point for inhabitants of Tramacastiel during an emergency.

Figure 5.1 *Pick-up point Tramacastiel and shelters of Villel and Teruel*

Table 5.1 *List of evacuees of Tramacastiel*

Village	Village ID	Enabled Evacuees	Disabled Evacuees (with total disability)	Disabled Evacuees (with partial disability)
Tramacastiel	100	37	1	6
Teruel (Shelter)	1000			
Villel (Shelter)	2000			

Table 5.1 provides the population of Tramacastiel village that need to be evacuated in Scenario A. The evacuees are categorized as follows: a) enabled evacuees that will be transported via buses, 4x4 vehicles, and vans, b) disabled evacuees with total disability that will be transported via ambulances and emergency mobile units and, c) disabled evacuees with partial disability that will be transported via vans or ambulances (if needed).The results of evacuation planning for Scenario A are shown in Table 5.2. The Table provides each route to be operated indicating which vehicle operates the route, the starting point, the pick-up location and the delivery location, and the exact number of evacuees collected at each route per type of evacuee. According to Table 5.2, only one ambulance and three vehicles for enabled and partially disabled evacuees are adequate for evacuating Tramacastiel. The total evacuation time is 97 min and the total distance covered by all vehicles to accomplish the evacuation plan is 202.4 km; 4 vehicles were employed during the evacuation operation.

Table 5.2 *Emergency plan for the evacuation of Tramacastiel to Villel*

Route No	Operating Vehicle ID	Type of Vehicle	Node Sequence	Route Start Time	Route End Time	Number Of Collected Evacuees		
						Enabled	**Totally Disabled**	**Partially Disabled**
Routes Operated by Ambulances								
1	43	Collective Ambulance (PR)	Teruel-Tramacastiel-Villel	0	97	1	1	2
Routes Operated by Fleet for Enable and Partially Disabled Evacuees								
1	74	Minibus (PR)	Teruel-Tramacastiel-Villel	0	97	22	0	3
2	44	Collective Ambulance (PR)	Teruel-Tramacastiel-Villel	0	73	1	0	1
3	53	Bus(PR)	Teruel-Tramacastiel-Villel	0	65	13	0	0
Total Evacuation Time = 97 min						**Total Distance = 202.4 km**		

*PR = Private Vehicle

The routes for the solution of the proposed algorithm of Table 5.2 are given also on a map in Figure 5.2, which shows the vehicle starting locations in Teruel, along with the pick-up location in Tramacastiel and the shelter in Villel.

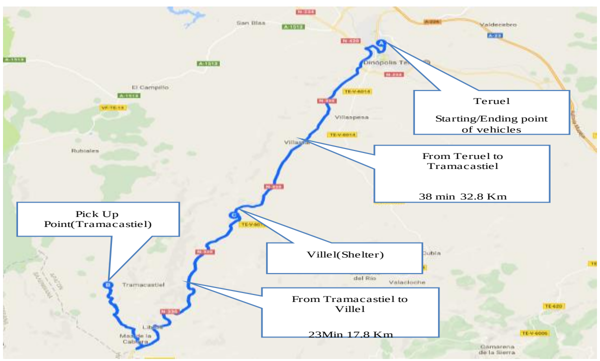

Figure 5.2 *Evacuation routes for PEHFP: Tramacastiel-Villel*

5.2 Scenario B: PEHFP solution for multipoint-to point evacuation

Scenario B addressed the case when a simultaneous evacuation of all three small villages Tramacastiel, Rubiales and El Campillo is required. The evacuees are to be transported to a safe shelter at the city of Teruel. The plan can be applied during an emergency when the entire Province of Teruel is threatened by a physical disaster. Table 5.3 presents the necessary data in terms of the number of evacuees per village.

The results of this large scale evacuation scenario are given in Table 5.4. According to Table 5.4, the algorithm uses 8 vehicles in total, each one operating just one route. The total evacuation time is 112 min, less than 2 hours, and the total distance is 511.6 km. Note that 11 vehicles were available for the evacuation operation.

Figure 5.3 *Pick-up points Tramacastiel, Rubiales, El Campillo and shelter of Teruel*

Table 5.3 *List of evacuees of pick-up points*

Village	Village ID	Enabled Evacuees	Disabled Evacuees (with total disability)	Disabled Evacuees (with partial disability)
Tramacastiel	100	37	1	6
Rubiales	200	26	1	4
El Campillo	300	33	1	6
Teruel (Shelter)	1000			

Table 5.4 *Emergency evacuation plan for Tramacastiel, Rubialles and El Campillo to Teruel*

Route No	Operating Vehicle ID	Type of Vehicle	Node Sequence	Route Start Time	Route End Time	Number Of Collected Evacuees		
						Enabled	Totally Disabled	Partially Disabled
Routes Operated by Ambulances								
1	43	Colective Ambulance(PR)	Teruel-El Campillo-Teruel	0	68	1	1	2
2	44	Colective Ambulance(PR)	Teruel -Tramacastiel-Teruel	0	112	1	1	2
3	45	Colective Ambulance(PR)	Teruel -Rubialles-Teruel	0	90	1	1	2
Routes Operated by Fleet for Enabled and Partially Disabled Evacuees								
1	74	Minibus(PR)	Teruel - Tramacastiel - Teruel	0	112	22	0	3
2	46	Colective Ambulance(PR)	Teruel - El Campillo - Teruel	0	56	1	0	2
3	47	Colective Ambulance(PR)	Teruel - El Campillo - Teruel	0	56	1	0	2
4	48	Colective	Teruel - Rubialles - Teruel	0	78	1	0	2

Route No	Operating Vehicle ID	Type of Vehicle	Node Sequence	Route Start Time	Route End Time	Number Of Collected Evacuees		
						Enabled	Totally Disabled	Partially Disabled
5	49	Ambulance(PR) Colective Ambulance(PR)	Teruel - Tramacastiel - Teruel	0	88	1	0	1
6	53	Bus(PR)	Teruel - El Campillo - Teruel	0	36	30	0	0
7	54	Bus(PR)	Teruel - Rubialles - Teruel	0	58	24	0	0
8	64	Bus(PR)	Teruel - Tramacastiel - Teruel	0	80	13	0	0
Total Evacuation Time = 112 min						**Total Distance = 511.6 km**		

In Figure 5.4, the routes of the vehicles are provided along with their starting locations in Teruel, the pick-up locations in Tramacastiel, Rubialles and El Campillo, as well as the shelter in Teruel.

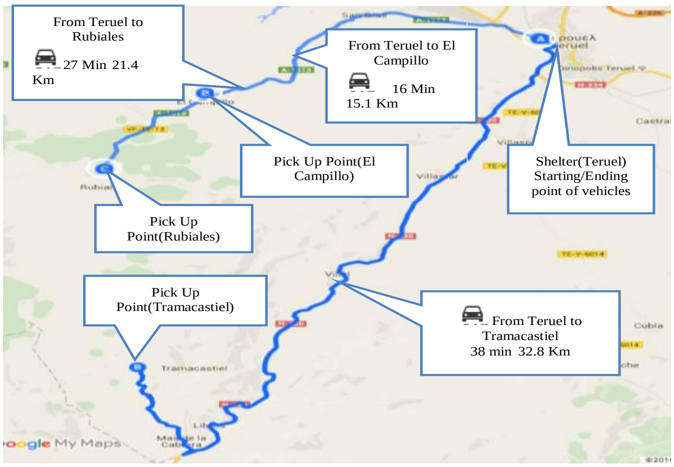

Figure 5.4 *Evacuation routes for PEHFP: Tramacastiel, Rubiales, ElCampillo-Teruel*

5.3 Scenario C: PEHFP solution for multi-point-to point evacuation

Scenario C includes firstly the evacuation of Tramacastiel and the transportation of the evacuees to Teruel; thereafter, having available the entire fleet, Rubiales and El Campillo are evacuated, and the evacuees are transported to Teruel. In table 5.5 the population to be evacuated is shown.

The results provided by the proposed algorithm for the evacuation of Tramacastiel and transportation of the evacuees to Teruel are given in Table 5.6. The total evacuation time is 112 min, and the total distance is 262.4 km; 4 vehicles were employed during evacuation.

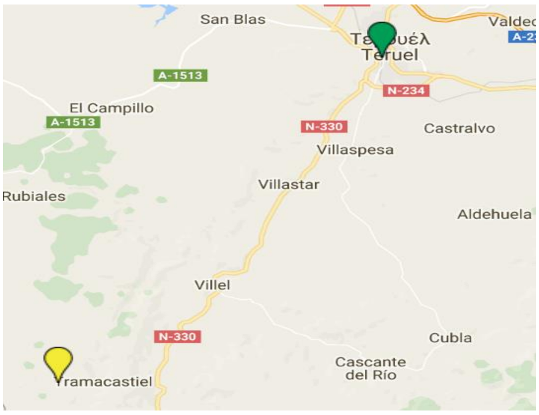

Figure 5.5 *Pick-up point Tramacastiel and shelter of Teruel*

Table 5.5 *List of evacuees of pick-up points*

Village	Village ID	Enabled Evacuees	Disabled Evacuees (with total disability)	Disabled Evacuees (with partial disability)
Tramacastiel	100	37	1	6
Teruel (Shelter)	1000			

Table 5.6 *Emergency evacuation plan for Pilot Test Event: Tramacastiel to Teruel*

Route No	Operating Vehicle ID	Type of Vehicle	Node Sequence	Route Start Time	Route End Time	Number Of Collected Evacuees		
						Enabled	Totally Disabled	Partially Disabled
Routes Operated by Ambulances								
1	43	Colective Ambulance (PR)	Teruel-Tramacastiel-Teruel	0	112	1	1	2
Routes Operated by Fleet for Enabled and Partially Disabled Evacuees								
1	74	Minibus (PR)	Teruel - Tramacastiel - Teruel	0	112	22	0	3
2	44	Colective Ambulance (PR)	Teruel - Tramacastiel - Teruel	0	88	1	0	1
3	53	Bus (PR)	Teruel - Tramacastiel - Teruel	0	80	13	0	0
Total Evacuation Time = 112 min						**Total Distance = 262.4 km**		

***PR = Private Vehicle**

The solution presented in Table 5.6 is given in Figure 5.6, which shows the vehicle routes, the starting locations in Teruel, the pick-up locations in Tramacastiel, as well as the shelter in Teruel.

Table 5.7 *List of evacuees of Rubiales and El Campillo*

Village	Village ID	Enabled Evacuees	Disabled Evacuees (with total disability)	Disabled Evacuees (with partial disability)
Rubiales	200	26	1	4
El Campillo	300	33	1	6
Teruel (Shelter)	1000			

Figure 5.6. *Evacuation routes for PEHFP: Tramacastiel -Teruel*

After Tramacastiel, according to the dynamic scenario, the villages of Rubialles and El Campillo are threatened by the evolving forest fire. Consequently, having available the entire fleet of vehicles, we need to plan a new evacuation schedule for the transportation of inhabitants of these two villages to Teruel. Table 5.7 presents the number of evacuees in Rubialles and El Campillo.

The results are shown in Table 5.8. The total evacuation time, after the evacuation of Tramacastiel, is 90 min, and the total distance is 249.2 km; 7 vehicles were employed during the evacuation operation.

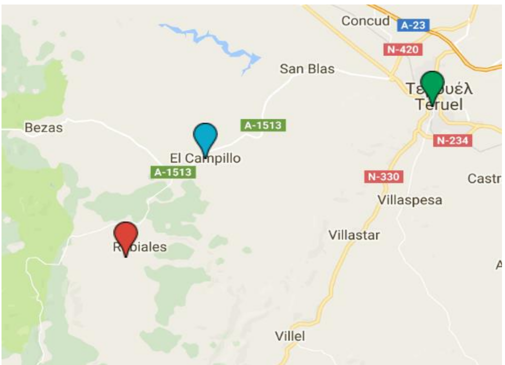

Figure 5.7. *Pick-up points Rubiales, El Campillo and shelter of Teruel*

Table 5.8 *Emergency evacuation plan for Pilot Test Event: Rubiales and El Campillo to Teruel*

Route No	Operating Vehicle ID	Type of Vehicle	Node Sequence	Route Start Time	Route End Time	Number Of Collected Evacuees		
						Enabled	Totally Disabled	Partially Disabled
Routes Operated by Ambulances								
1	43	Colective Ambulance (PR)	Teruel -El Campillo-Teruel	0	68	1	1	2
2	44	Colective Ambulance (PR)	Teruel -Rubialles-Teruel	0	90	1	1	2
Routes Operated by Fleet for Enabled and Partially Disabled Evacuees								
1	74	Minibus (PR)	Teruel - El Campillo - Teruel	0	68	22	0	3

Route No	Operating Vehicle ID	Type of Vehicle	Node Sequence	Route Start Time	Route End Time	Number Of Collected Evacuees		
						Enabled	Totally Disabled	Partially Disabled
2	45	Colective Ambulance (PR)	Teruel - Rubialles-Teruel	0	66	1	0	2
3	46	Colective Ambulance (PR)	Teruel - El Campillo - Teruel	0	44	1	0	1
4	53	Bus (PR)	Teruel - Rubialles- Teruel	0	58	24	0	0
5	54	Bus (PR)	Teruel - El Campillo - Teruel	0	36	9	0	0
Total Evacuation Time = 90 min						**Total Distance = 249.2 km**		

***PR = Private Vehicle**

The routes for evacuating Rubialles and El Campillo, and transporting the evacuees to the shelter in Teruel are shown in Figure 5.8. The Figure shows the vehicle starting locations in Teruel, the pick-up locations in Rubialles and El Campillo and the shelter in Teruel.

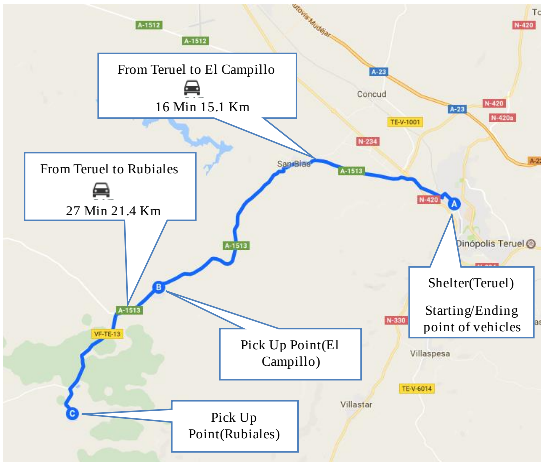

***Figure 5.8**. Evacuation routes for PEHFP: Rubialles, El Campillo -Teruel*

6. Conclusions

The last decades due to the increasing frequency of both natural and man-made disasters, evacuation planning of affected populations is of great importance. Evacuation planning is a complex process and its effectiveness depends on several factors, such as warning time, response time, etc. Many researchers have developed mathematical models, algorithms and simulation programs in order to develop effective evacuation plans, which can be applied to various disaster events such as floods, fires etc.

In this thesis the Population Evacuation using Heterogeneous Fleet Problem (PEHFP) is proposed. PEHFP deals with the evacuation of population that characterized by different types of evacuees as far as their mobility status concerns. In particular, we deal with three types of evacuees. The first types of evacuees, enabled evacuees, are those who do not need any special transportation treatment. The second type of evacuees concerns people with partial disability who use a wheel chair. The third type concerns citizens with more severe disability who need to be transported on stretchers. To describe PEHFP a mathematical programming model has been developed. The objective is to minimize the total time needed for evacuating the population from a set of pick-up points under all related constraints.

Trying to obtain an optimal solution based on this mathematical problem in reasonable time is not feasible for problems of practical size. Thus, two heuristic algorithms were developed to solve this problem. The heuristic algorithms obtain near optimal solutions in reasonable time and they are applied initially to instances of the evacuation problem, in which all evacuees are able and do not face and mobility challenges. The proposed heuristics have been compared in terms of total evacuation time and it proved that H1 minimizes the evacuation time in contrast to H2. H1 utilizes more vehicles for meeting the demand and therefore, H1 manages to complete the evacuation process earlier. Consequently, H1 is chosen to be applied to the more complex case, in which some of the evacuees are characterized by one of two forms of physical disability (which need particular treatment).

Finally, the proposed algorithm was applied to a case study which deals with the evacuation of three small villages when a forest fire occurs. The results obtained provide a route schedule for each vehicle that is needed for the evacuation. The route schedule includes the starting location of each vehicle, the pick-up points visited, the number of evacuees that are collected per type of evacuee and the exact time needed for each vehicle to collect the evacuees and transport them to a safe shelter. The case study illustrates the practicality of the proposed algorithm to provide efficient solutions to practical PEHF problems.

Comparing with the existing literature in the area of evacuation planning, the proposed PEHFP takes into account heterogeneous fleet, multiple trips, multiple visits at each pick-up location, and, importantly, treats different types of evacuees. We have observed that evacuees with mobility disabilities have a great impact on the total evacuation time and that using more vehicles adapted for disabled evacuees can lead to significant reduction of total evacuation time.

It should be noted that both the proposed mathematical model and heuristic algorithms can be used for any type of disaster, provided that the appropriate inputs are available.

Further research may be done in planning the evacuation process. Uncertainties concerning the availability of road links may be included in the model. For instance,

- Uncontrolled fires are able to damage road links, making some parts or roads inaccessible. In such cases, alternative routes must be provided
- Future work may also include the development of more advanced heuristics, or metaheuristics, to deal with PEHFP
- Ways of overcoming difficulties of incapacitated vehicles may also be investigated.

References

[1].Alexander, D. E. (2002). Principles of emergency planning and management. *Oxford University Press on Demand*.

[2].Stepanov, A., & Smith, J. M. (2009). Multi-objective evacuation routing in transportation networks. *European Journal of Operational Research*, *198*(2), 435-446.

[3].Pel, A. J., Bliemer, M. C., & Hoogendoorn, S. P. (2012). A review on travel behaviour modelling in dynamic traffic simulation models for evacuations. *Transportation*, *39*(1), 97-123.

[4].Liu, Y., & Yu, J. (2011). Emergency Evacuation Planning for Highly Populated Urban Zones: A Transit-Based Solution and Optimal Operational Strategies. *Edited by Burak Eksioglu*, 59.

[5]. unisdr.org, (2000). The United Nations Office for Disaster Risk Reduction. Available at: https://www.unisdr.org/we/inform/disaster-statistics [Accessed 2 March 2017].

[6].Bish, D. R. (2011). Planning for a bus-based evacuation. *OR spectrum*, *33*(3), 629-654.

[7].Goerigk, M., & Grün, B. (2012). The robust bus evacuation problem. *University of Kaiserslautern, preprint, KLUEDO*.

[8].Perkins, J.A., Dabipi, I.K. and Han, L.D., (2001). *Modeling Transit Issues Unique to Hurricane Evacuations: North Carolina's Small Urban and Rural Areas* (No. Final Report).

[9].Margulis, L., Charosky, P., Fernandez, J. and Centeno, M.A., (2006) June. Hurricane evacuation decision-support model for bus dispatch. In *Fourth LACCEI International Latin American and Caribbean Conference for Engineering and Technology (LACCET '2006),"Breaking Frontiers and Barriers in Engineering: Education, Research, and Practice* (pp. 21-23).

[10].Goerigk, M., Grün, B. and Heßler, P., 2013. Branch and bound algorithms for the bus evacuation problem. *Computers & Operations Research*, *40*(12), pp.3010-3020.

[11].Sayyady, F., & Eksioglu, S. D. (2010). Optimizing the use of public transit system during no-notice evacuation of urban areas. *Computers & Industrial Engineering*, *59*(4), 488-495.

[12].Dikas, G. and Minis, I., 2016. Solving the bus evacuation problem and its variants. *Computers & Operations Research*, *70*, pp.75-86.

[13].Zheng, H., 2014. Optimization of bus routing strategies for evacuation. *Journal of advanced transportation*, 48(7), pp.734-749.

[14].Hamacher, H.W. and Tjandra, S.A., 2001. Mathematical modelling of evacuation problems: A state of art. *Fraunhofer-Institut für Techno-und Wirtschaftsmathematik, Fraunhofer* (ITWM).

[15].Altay, N., & Green, W. G. (2006). OR/MS research in disaster operations management. *European journal of operational research*, *175*(1), 475-493.

[16].Bretschneider, S. (2012). Mathematical models for evacuation planning in urban areas (Vol. 659). *Springer Science & Business Media.*

[17].Murray-Tuite, P., & Wolshon, B. (2013). Evacuation transportation modeling: An overview of research, development, and practice. *Transportation Research Part C: Emerging Technologies*, *27*, 25-45.

[18].Goerigk, M., Grün, B., & Heßler, P. (2014). Combining bus evacuation with location decisions: A branch-and-price approach. *Transportation Research Procedia*, *2*, 783-791.

[19].Bard, J. F., Huang, L., Dror, M., & Jaillet, P. (1998). A branch and cut algorithm for the VRP with satellite facilities. *IIE transactions*, *30*(9), 821-834.

[20]. Taillard, É. D., Laporte, G., & Gendreau, M. (1996). Vehicle routing with multiple use of vehicles. *Journal of the Operational research society*, *47*(8), 1065-1070.

[21].Brandao, J. C. S., & Mercer, A. (1998). The multi-trip vehicle routing problem. *Journal of the Operational research society*, *49*(8), 799-805.

[22].Petch, R. J., & Salhi, S. (2003). A multi-phase constructive heuristic for the vehicle routing problem with multiple trips. *Discrete Applied Mathematics*, *133*(1), 69-92.

[23].Salhi, S., & Petch, R. J. (2007). A GA based heuristic for the vehicle routing problem with multiple trips. *Journal of Mathematical Modelling and Algorithms*, *6*(4), 591-613.

[24].Crevier, B., Cordeau, J. F., & Laporte, G. (2007). The multi-depot vehicle routing problem with inter-depot routes. *European Journal of Operational Research*, *176*(2), 756-773.

[25].Tarantilis, C. D., Zachariadis, E. E., & Kiranoudis, C. T. (2008). A hybrid guided local search for the vehicle-routing problem with intermediate replenishment facilities. *INFORMS Journal on Computing*, *20*(1), 154-168.

[26].Nagy, G., & Salhi, S. (2007). Location-routing: Issues, models and methods. *European Journal of Operational Research*, *177*(2), 649-672.

[27].Belenguer, J. M., Benavent, E., Prins, C., Prodhon, C., & Calvo, R. W. (2011). A branch-and-cut method for the capacitated location-routing problem. *Computers & Operations Research*, *38*(6), 931-941.

[28].Escobar, J. W., Linfati, R., & Toth, P. (2013). A two-phase hybrid heuristic algorithm for the capacitated location-routing problem. *Computers & Operations Research*, *40*(1), 70-79.

[29].Prins, C., Prodhon, C., & Calvo, R. W. (2006). Solving the capacitated location-routing problem by a GRASP complemented by a learning process and a path relinking. *4OR: A Quarterly Journal of Operations Research*, *4*(3), 221-238.

[30].He, S., Zhang, L., Song, R., Wen, Y., & Wu, D. (2009). Optimal transit routing problem for emergency evacuations. In *Transportation Research Board 88th Annual Meeting* (No. 09-0931).

Appendix A. PEHFP: Algorithm and Pseudo code for H1

A.1 Notation

$G(N, A)$ is a directed graph where N is the set of all nodes related to the problem, and A is the set of arcs that connect the nodes.

Nodes and vehicles

- Let $\{t\} \subset N$ be the shelter
- Let $C \subset N$ be the set of all nodes representing the evacuee locations, called *pick-up locations*. In particular: $C = \{1,2, \dots, m\}$.
- Let $K = \{1,2, \dots, v\}$ be the set of available vehicles
- Let $S^k \subset N,\ k \in K$ be the originating location of vehicle k. In particular: $S^k = \{\mathrm{s}^1, \mathrm{s}^2, \dots, \mathrm{s}^v\}$
- Let $E^k \subset N,\ k \in K$ be the ending location of vehicle k. In particular: $E^k = \{e^1, e^2, \dots, e^v\}$

Arcs (travel times)

- Let l_{ij} be the traveling time from node i to node j, $i, j \in N, i \neq j$. In particular:

$$\mathbf{L} = \begin{cases} \begin{pmatrix} l_{s^1 1} & \cdots & l_{s^1 m} \\ \vdots & \ddots & \vdots \\ l_{s^v 1} & \cdots & l_{s^v m} \end{pmatrix}, & i \in S^k, j \in C \\ \begin{pmatrix} l_{ts^1} & \dots & l_{ts^v} \end{pmatrix}, & i \in \{t\}, j \in S^k \\ \begin{pmatrix} 0 & l_{12} & \dots & l_{1m} \\ l_{21} & 0 & \dots & l_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ l_{m1} & l_{m2} & \dots & 0 \end{pmatrix}, & i, j \in C, i \neq j \end{cases}$$

Arcs (distances)

- Let p_{ij} *b*e the travel distance from node i to node j, $i, j \in N, i \neq j$. In particular:

$$\mathbf{P} = \begin{cases} \begin{pmatrix} p_{s^1 1} & \cdots & p_{s^1 m} \\ \vdots & \ddots & \vdots \\ p_{s^v 1} & \cdots & p_{s^v m} \end{pmatrix}, & i \in S^k, j \in C \\ (p_{ts^1} \quad \cdots \quad p_{ts^v}), & i \in \{t\}, j \in S^k \\ \begin{pmatrix} 0 & p_{12} & \cdots & p_{1m} \\ p_{21} & 0 & \cdots & p_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ p_{m1} & p_{m2} & \cdots & 0 \end{pmatrix}, & i, j \in C, i \neq j \end{cases}$$

Other

- Let D_i be the demand of each pick up location $i \in C$
- Let Q_k be the capacity of vehicle $k \in K$
- Let $ICV = \{Q_k, k \in K\}$ be the set of initial vehicles' capacities
- Let $List$ be the list of all the available vehicles $k \in K$ arranged in descending order with respect to capacity.
- Let $Time\_List$ be the list of the traveling times of the vehicles. Note that initially all the elements of $Time\_List$ are zero.
- Let $Capacity\_List$ be a list of vehicles with the same traveling time arranged in descending order with respect to their capacity.
- Let $Vehicles\_List$ be a set of vehicles $k \in K$ which have not returned to the shelter while the entire demand has been met
- Let T_{evac} be the time that the last evacuee is dropped off at shelter $\{t\}$
- Let $Demand\_List$ be a list with the demand of nodes arranged in descending order of demand.
- Let $Furthest\_List$ be a list of nodes of equal demand, arranged in descending order with respect to their distance from the starting location of current vehicle (or from the shelter).
- Let $Nearest\_List$ be a list with nodes of equal demand, arranged in ascending order with respect to their distance from the starting location of current vehicle (or from the shelter).
- Let $Total\_Distance$ be the list with the traveling distances of each vehicle. Note that initially all the elements in $Total\_Distance$ are equal to zero
- Let $st = 2$ minutes be the loading/unloading time of each vehicle
- Let $Distance$ be the total traveling distance of all vehicles:
  $Distance = \sum_{k \in K} Total\_Distance(k)$
- Let $node$ be the last node that vehicle $k \in K$ visits during its last route

A.2 H1 and the corresponding pseudocode

Heuristic algorithm 1 comprises the following steps:

Step 1. Sort the vehicles in descending order with respect to their initial capacity ($List$).

Step 2. Sort the demand of nodes in descending order ($Demand\_List$).

Step 3. Set the first vehicle k in $List$ as current vehicle (CV), delete it from $List$ and route it considering the following cases:

- **Case 1:** In case there are more than one nodes with the same highest demand in $Demand\_List$
  - **Subcase 1:** If CV's capacity is higher than the first element in $Demand\_List$
    - Route CV to the furthest of the nodes with the same highest demand. Set this node as CN
  - **Subcase 2:** If CV's capacity is less or equal to the first element in $Demand\_List$
    - Route CV to the nearest of the nodes with the same highest demand. Set this node as CN
- **Case 2:** In case the first element in $Demand\_List$ is unique, route vehicle CV to the first node in $Demand\_List$ and set this node as current node (CN).

Step 4. Record the travel time of CV ($Time\_List$) and its travel distance ($Total\_Distance$)

Step 5. Update the capacity of current vehicle (CV) and the demand of current node (CN) as follows:

- **Case 1:** In case that CN's demand is greater than CV's capacity
  - CV picks up Q_k evacuees and returns to the shelter
  - update the travel time of CV :
    $Time\_List(CV) = Time\_List(CV) + L_{CN\{t\}} + 2 \cdot st$
  - update the demand of CN: $D_{CN} = D_{CN} - Q_{CV}$,
  - update the capacity of CV: $Q_{CV}=0$
  - update the traveling distance of CV:
    $Total\_Distance(CV) = Total\_Distance(CV) + P_{CN\{t\}}$
- **Case 2:** In case that CN's demand is lower than CV's capacity
  - CV picks up D_{CN} evacuees
  - update the travel time of CV: $Time\_List(CV) = Time\_List(CV) + st$
  - update the capacity of CV: $Q_{CV} = Q_{CV} - D_{CN}$,
  - update the demand of CN: $D_{CN} = 0$.

Step 6.

- **Case 1:** If $List$ is not exhausted and demand is not met go to **Step 2**

- **Case 2** If demand is met
    - route all vehicles (those which are not at the shelter) to the shelter
    - update their traveling time and their travel distance
    - set T_{evac}= maximum element in $Time\_List$
    - set $Distance = sum(Total\_Distance)$
    - **End**
- **Case 3:** If $List$ is exhausted and demand is not met, sort $Time\_List$ in ascending order
    - **Subcase 1:** In case there are more than one vehicles with the same minimum traveling time in $Time\_List$
        - Select among the vehicles with the same minimum travel time, the one that has the highest capacity and set it as current vehicle(CV)
    - **Subcase 2:** In case the first element in $Time\_List$ is unique, set the corresponding vehicle as CV.
        - **Subcase 2.1:** In case that CV is at the shelter
            - restore its capacity
            - repeat **Step 2**
            - repeat **Case 1** or **Case 2** of **Step 3** (depends on CV's capacity)
            - repeat **Steps 4-6**.
        - **Subcase 2.2:** In case that CV is not at the shelter
            - find its current node and route it to its nearest node (nn) with nonzero demand
            - update its travel time and its travel distance:

              $Time\_List(CV) = Time\_List(CV) + L_{CN\{nn\}}$

              $Total\_Distance(CV) = Total\_Distance(CV) + P_{CN\{nn\}}$
            - Set the nearest node as CN and go to **Step 5**.

The proposed heuristic algorithm 1 for PEHFP is implemented using Matlab R2010b on a PC equipped with a 1.8 GHz Intel Core i5 and 4 GB of RAM. The pseudo code of the algorithm is given in the following:

Step 1. Set $T_{evac} = 0, Time\_List = 0, Total\_Distance = 0$

Step 2. **While** $\sum_{c\in C} D_c > 0$

Step 3. **While** $List$ is not empty

- Set the first vehicle $k \in K$ in $List$ with capacity Q_k as current vehicle (CV=k)
- Delete vehicle k from the $List$
- Sort nodes in descending order with respect to their demand ($Demand\_List$)

Step 3.1 **If** $Demand\_List(1) \neq Demand\_List(2) \neq \cdots .. \neq Demand\_List(z), z > 1$

- Set node i with demand $D_i = Demand\_List(1)$ as current node (CN)

Elseif $Demand\_List(1) = Demand\_List(2) = \cdots = Demand\_List(z), z > 1$ **AND** $Q_{CV} > Demand\_List(1)$

- Find nodes $i = 1, 2, \ldots, z$ with demand $D_i = Demand\_List(1)$ and sort them in descending order with respect to their distance from the starting location of CV $(Furthest\_List)$
- Set $CN = Furthest\_List\ (1)$

Elseif $Demand\_List(1) = Demand\_List(2) = \cdots = Demand\_List(z), z > 1$ **AND** $Q_{CV} \leq Demand\_List(1)$

- Find nodes $i = 1, 2, \ldots, z$ with demand $D_i = Demand\_List(1)$ and sort them in ascending order with respect to their distance from the starting location of CV $(Nearest\_List)$.
- Set $CN = Nearest\_List\ (1)$

End

- Route CV to CN
- Set $Time\_List(CV) = Time\_List(CV) + L_{S^{CV}CN}$
- Set $Total\_Distance(CV) = Total\_Distance$ (CV) $+ P_{S^{CV}CN}$

Step 3.2 **If** $\boldsymbol{D_{CN} < Q_{CV}}$

- load vehicle CV with $\boldsymbol{D_{CN}}$ evacuees
- update the capacity of vehicle CV**:** $Q_{\boldsymbol{CV}} \leftarrow Q_{\boldsymbol{CV}} - D_{\boldsymbol{CN}}$
- update the demand of node CN: $D_{\boldsymbol{CN}} = 0$
- Set $Time\_List(CV) = Time\_List(CV) + st$

Elseif $\boldsymbol{D_{CN} \geq Q_{CV}}$

- load vehicle CV with Q_{CV} evacuees
- update the capacity of vehicle CV**:** $Q_{\boldsymbol{CV}} = 0$
- update the demand of node CN: $D_{CN} \leftarrow D_{CN} - Q_{CV}$
- route vehicle CV to shelter $\{t\}$ to drop off the evacuees
- Set $Time\_List(CV) = Time\_List(CV) + L_{CN\,\{t\}} + 2 \cdot st$
- Set $Total\_Distance(CV) = Total\_Distance$ (CV) $+ P_{CN\,\{t\}}$
- Vehicle CV becomes available and its capacity is restored from ICV

End

End

Step 4. Sort $Time\_List$ in ascending order

Step 5. **If** $Time\_List(1) = Time\_List(2) = \cdots = Time\_List(z), z > 1$

- Sort vehicles in descending order with respect to their capacity $(Capacity\_List)$.

Step 5.1 **While** $Capacity\_List$ is not empty

- Set the first vehicle $k \in K$ in $Capacity\_List$ with capacity Q_k as CV
- Delete vehicle k from the $Capacity\_List$

Step 5.2 **If** CV is at the shelter ($node = \{t\}$)

- Sort the nodes in descending order with respect to their demand ($Demand\_List$)

Step 5.3 **If** $Demand\_List(1) \neq Demand\_List(2) \neq \cdots .. \neq Demand\_List(z), z > 1$

- Set the node i with demand $D_i = Demand\_List(1)$ as current node CN

Elseif $Demand\_List(1) = Demand\_List(2) = \cdots = Demand\_List(z), z > 1$ **AND** $Q_{CV} > Demand\_List(1)$

- Find the nodes $i = 1, 2, \dots, z$with demand $D_i = Demand\_List(1)$ and sort them in descending order with respect to their distance from the shelter ($Furthest\_List$)
- Set $CN = Furthest\_List\ (1)$

Elseif $Demand\_List(1) = Demand\_List(2) = \cdots = Demand\_List(z), z > 1$ **AND** $Q_{CV} \leq Demand\_List(1)$

- Find the nodes $i = 1, 2, \dots, z$ with demand $D_i = Demand\_List(1)$ and sort them in ascending order with respect to their distance from the shelter ($Nearest\_List$).
- Set $CN = Nearest\_List\ (1)$

End

- Route CV to CN
- Set $Time\_List(CV) = Time\_List(CV) + L_{\{t\}CN}$
- Set $Total\_Distance(CV) = Total\_Distance$ (CV) $+ P_{\{t\}CN}$

Elseif CV is not at the shelter (node $\neq$ {t})

- Route vehicle CV from its CN to node l with $\min\{L_{CNl}, l \in C \backslash \{CN\}\}$ and demand $D_l \neq 0$.
- Set $Time\_List(CV) = Time\_List(CV) + L_{CNl}$
- Set $Total\_Distance(CV) = Total\_Distance$ (CV) $+ P_{CNl}$
- Set $CN = l$

End

Step 5.4 **If** $D_{CN} < Q_{CV}$

- load vehicle CV with D_{CN} evacuees
- update the capacity of vehicle CV: $Q_{CV} \leftarrow Q_{CV} - D_{CN}$
- update the demand of node CN: $D_{CN} = 0$
- Set $Time\_List(CV) = Time\_List(CV) + st$

Elseif $D_{CN} \geq Q_{CV}$

- load vehicle CV with Q_{CV} evacuees
- update the capacity of vehicle CV: $Q_{CV} = 0$
- update the demand of node CN: $D_{CN} \leftarrow D_{CN} - Q_{CV}$
- route vehicle CV to the shelter $\{t\}$ to drop off the evacuees onboard
- Set $Time\_List(CV) = Time\_List(CV) + L_{CN\{t\}} + 2 \cdot st$
- Set $Total\_Distance(CV) = Total\_Distance$ (CV) + $P_{CN\{t\}}$
- Vehicle CV becomes available and its capacity is restored from ICV

End

End

Else

- Set the first vehicle (k, with capacity Q_k) in the $Time\_List$ as CV

Repeat ***Step 5.2-Step 5.4***

End

End

Step 6. Find the vehicles which are not at the shelter

Step 6.1 While $Vehicles\_List$ is not empty

- Set the first vehicle $k \in K$ in $Vehicles\_List$ as CV
- Delete vehicle k from the $Vehicles\_List$
- Route vehicle CV from its current node to the shelter
- Set $Time\_List(CV) = Time\_List(CV) + L_{CN\{t\}} + st$
- Set $Total\_Distance(CV) = Total\_Distance$ (CV) + $P_{CN\{t\}}$

End

Step 7. Find $maxelement\{Time\_List\}$

- $T_{evac} = maxelement\{Time\_List\}$
- $Distance = \sum_{k \in K} Total\_Distance(k)$

Step 8. Stop

Appendix.B PEHFP: Algorithm and Pseudocode for Heuristic Algorithm 2

B.1 Notation

The notation for the second heuristic algorithm is exactly the same as of the previous algorithm apart from the $Arrival\_List$, which is a list with vehicle arrival times at the shelter. Note that initially all the elements of $Arrival\_List$ are zero. Moreover, note that the last node that vehicle k ∈ K visits during its last route, previously denoted as $node$, is not used in the second algorithm.

B.2 Heuristic algorithm 2 and the corresponding pseudocode

Heuristic algorithm 2 comprises the following steps:

Step 1. Sort the vehicles in descending order with respect to their initial capacity ($List$).

Step 2.

- **Case 1:** If $List$ is not empty and demand is not met
  - Set the first vehicle k in $List$ as current vehicle (CV) and delete it from $List$
- **Case 2:** If $List$ is empty and demand is not satisfied
  - Sort $Time\_List$ in ascending order
  - **Subcase 2.1:** In case there are more than one vehicles with the same minimum traveling time in $Time\_List$
    - Select among the vehicles with the same minimum travel time the one that has the highest capacity and set it as current vehicle(CV)
  - **Subcase 2.2:** In case that the first element in $Time\_List$ is unique, set the corresponding vehicle as CV.

Step 3. Sort the demand of nodes in descending order ($Demand\_List$).

- **Case 1:** In case there are more than one nodes with the same highest demand in $Demand\_List$:
  - **Subcase 1.1:** If CV's capacity is higher than the first element in $Demand\_List$
    - Route CV to the furthest of the nodes with the same highest demand and set this node as CN
  - **Subcase 1.2:** If CV's capacity is less or equal to the first element in $Demand\_List$

- Route CV to the nearest of the nodes with the same highest demand and set this node as CN

- **Case 2:** In case the first element in $Demand\_List$ is unique, route vehicle CV to the corresponding node (with the highest demand) and set this node as current node (CN)

Step 4. Record CV's traveling time ($Time\_List$) and its travel distance ($Total\_Distance$)

Step 5. Update the capacity of CV and the demand of CN as follows:

- **Case 1:** In case CN's demand is higher than CV's capacity
  - CV picks up Q_{CV} evacuees and returns to the shelter
  - update the demand of CN: $D_{CN} = D_{CN} - Q_{CV}$
    - update the capacity of CV: $Q_{CV} = 0$
    - CV returns to the shelter
      - update the travel distance of CV:

        $Total\_Distance(CV) = Total\_Distance(CV) + P_{CN\{t\}}$
      - update the travel time of CV :

        $Time\_List(CV) = Time\_List(CV) + L_{CN\{t\}} + 2 \cdot st$
- **Case 2:** In case CN's demand is lower than CV's capacity
  - CV picks up D_{CN} evacuees
  - update the capacity of CV: $Q_{CV} = Q_{CV} - D_{CN}$
  - update the demand of CN: $D_{CN} = 0$
  - update the traveling time of CV:

    $Time\_List(CV) = Time\_List(CV) + st.$
  - $t$he last visited node is CN

Step 6.

- **Case 1:** If demand is not met
  - **Subcase 1.1:** If the remaining capacity of CV is higher than zero:
    - route CV from its CN to the nearest node (nn) with nonzero demand
    - update travel distance of CV:

      $Total\_Distance(CV) = Total\_Distance(CV) + P_{CN\{nn\}}$
    - update traveling time of CV:

      $Time\_List(CV) = Time\_List(CV) + L_{CN\{nn\}}$
    - Repeat **Steps 5-6**
  - **Subcase 1.2:** If the remaining capacity of CV is equal to zero
    - Repeat **Steps 2-6**
- **Case 2:** If demand is met
  - route all the vehicles to the shelter
  - update their traveling time and their travel distance

- set T_{evac}= maximum element in $Time\_List$
- set $Distance = sum(Total\_Distance)$
- **End**

The proposed heuristic algorithm 1 for PEHFP is implemented using Matlab R2010b on a PC equipped with a 1.8 GHz Intel Core i5 and 4 GB of RAM. The pseudo code of the algorithm is given in the following:

Step 1. Set $T_{evac} = 0$, $Time\_List = 0$, $Arrival\_List = 0$,$Total\_Distance = 0$

Step 2. **While** $\sum_{c \in C} D_c > 0$

Step 3.

Step 3.1 **If** $List$ is not empty

- Set the first vehicle $k \in K$ in $List$ with capacity Q_k as current vehicle $(CV = k)$
- Delete vehicle k from the $List$
- Sort the demand of nodes in descending order $(Demand\_List)$

Step 3.1.1 **If** $Demand\_List(1) \neq Demand\_List(2) \neq \cdots \neq Demand\_List(z), z > 1$

Set node i with demand $D_i = Demand\_List(1)$ as current node (CN)

Elseif $Demand\_List(1) = Demand\_List(2) = \cdots = Demand\_List(z), z > 1$

AND $Q_{CV} > Demand\_List(1)$

- Find nodes $i = 1, 2, \dots, z$ with demand $D_i = Demand\_List(1)$ and sort them in descending order with respect to their distance from the starting location of CV $(Futhest\_List)$
- Set $CN = Furthest\_List\ (1)$

Elseif $Demand\_List(1) = Demand\_List(2) = \cdots = Demand\_List(z), z > 1$

AND $Q_{CV} \leq Demand\_List(1)$

- Find nodes $i = 1, 2, \dots, z$ with demand $D_i = Demand\_List(1)$ and sort them in ascending order with respect to their distance from the startinglocation of $CV(Nearest\_List)$.
- Set $CN = Nearest\_List\ (1)$

End

- Route CV to CN
- Set $\mathrm{Time\_List(CV)} = \mathrm{Time\_List(CV)} + \mathrm{L}_{\mathrm{S_{CV}}\mathrm{CN}}$
- Set $\mathrm{Total\_Distance(CV)} = \mathrm{Total\_Distance(CV)} + \mathrm{P}_{\mathrm{S_{CV}}\mathrm{CN}}$

Step 3.2 **Elseif** $List$ is empty

Sort the $Arrival\_List$ in ascending order

Step 3.2.1 **If** $Arrival\_List(1) = Arrival\_List(2) = \cdots = Arrival\_List(z), z > 1$

- Set vehicle k with capacity $Q_k = max\{Q_{Arrival\_List(j)}, j = 1,2, \dots, z\}$ as current vehicle $(CV = k)$
- CV becomes available and its capacity is restored from ICV
- Delete vehicle k from the $Arrival\_List$

Else

- Set the first vehicle k with capacity Q_k) in $Arrival\_List$ as current vehicle $(CV = k)$
- Vehicle CV becomes available and its capacity is restored from ICV
- Delete vehicle k from the $Arrival\_List$

End

- Sort nodes in descending order with respect to their demand $(Demand\_List)$

Step 3.3 **If** $Demand\_List(1) \neq 1Demand\_List(2) \neq \cdots \neq Demand\_List(z), z > 1$

- Set node i with demand $D_i = Demand\_List(1)$ as current node (CN)

Elseif $Demand\_List(1) = Demand\_List(2) = \cdots = Demand\_List(z), z > 1$

AND $Q_{CV} > d_{(Demand\_List(1))}$

- Find nodes $i = 1, 2, \dots, z$ with demand $D_i = Demand\_List(1)$ and sort
- them in descending order with respect to their distance from the shelter
- $(Furthest\_List)$
- Set $CN = Furthest\_List\ (1)$

Elseif $Demand\_List(1) = Demand\_List(2) = \cdots = Demand\_List(z), z > 1$

AND $Q_{CV} \leq d_{(Demand\_List(1))}$

- Find nodes $i = 1, 2, \dots, z$ with demand $D_i = Demand_List(1)$and sort
- them in ascending order with respect to their distance from the shelter
- $(Nearest\_List)$.
- Set $CN = Nearest\_List\ (1)$

End

- Route vehicle CV to CN
- Set $Time\_List(CV) = Time\_List(CV) + L_{\{t\}CN}$
- Set $Total\_Distance(CV) = Total\_Distance(CV) + P_{\{t\}CN}$

End

Step 4.

Step 4.1 **If** $D_{CN} < Q_{CV}$

- load vehicle CV with D_{CN} evacuees
- update the capacity of vehicle CV: $Q_{CV} \leftarrow Q_{CV} - D_{CN}$
- update the demand of node CN: $D_{CN} = 0$

- Set $Time\_List(CV) = Time\_List(CV) + st$

Elseif $D_{CN} \geq Q_{CV}$

- load vehicle CV with Q_{CV} evacuees
- update the demand of node CN: $D_{CN} \leftarrow D_{CN} - Q_{CV}$
- route vehicle CV to the shelter t to drop off the evacuees
- Set $Arrival\_List(CV) = Time\_List(CV) + L_{CN\{t\}} + 2 \cdot st$
- Set $Time\_List(CV) = Arrival\_List(CV)$

End

Step 4.2 **While** $Q_{CV} > 0$

- Route vehicle CV from CN to node l with $min\{L_{CN\,l}, l \in C \setminus \{CN\}\}$ and demand $D_l \neq 0$
- Set $Time\_List(CV) = Time\_List(CV) + L_{CN\,l}$
- Set $Total\_Distance(CV) = Total\_Distance(CV) + P_{CNl}$
- Set $CN = l$

Repeat *Steps 4.1-4.2*

End

End

Step 5. Find $max\ element\ \{Time\_List\}$

- $T_{evac} = \max element\{Time\_List\}$
- $Distance = \sum_{k \in K} Total\_Distance(k)$

Step 6. Stop

Appendix C. PEHFP: Pseudo code of Heuristic for enabled and disabled population evacuation

Notation

Since most of the notation has been already defined in Section 2.3.2, we present only the additional notation.

Nodes and vehicles

- We consider three categories of evacuees (enabled, partially disabled and totally disabled evacuees). Let $P = \{1,2,3\}$ be the set of evacuees category and let $p \in P$. Let $p = 1$ denote enabled evacuees, $p = 2$ denote partially disabled evacuees and $p = 3$ denote totally disabled evacuees.
- Let D_i^p be the demand of evacuee type $p \in P$ at pick up point $i \in C$
- Let Q_k^p be the capacity of vehicle $k \in K$ for evacuee type $p \in P$
- Let $ICV = \{Q_{0,k}^p, k \in K, p \in P\}$ be an array of initial vehicle capacity ($Q_{0,k}^p$ is the initial capacity of vehicle $k \in K$ for evacuees type of $p \in P$) . ICV is an $3 \times |K|$ array, the rows of which correspond to evacuee type and the columns to vehicles. Note that a vehicle that can transport partially disabled evacuees ($p = 2$) can also transport enabled evacuees ($p = 1$), while a vehicle that can transfer enabled evacuees ($p = 1$) cannot necessarily transfer partially disabled evacuees ($p = 2$) or totally disabled evacuees ($p = 3$). In addition, a vehicle $k \in K$ that can transfer totally disabled evacuees ($p = 3$) may transfer partially disabled evacuees ($p = 2$). Finally, we assume that an ambulance can transfer only one totally disabled evacuee ($p = 3$) per ride.
- Consider the following indicators:

$$r_k = \begin{cases} 1, if\ Q_{0,k}^2 > 0\ AND\ Q_{0,k}^3 > 0 \\ 0, if\ Q_{0,k}^2 = 0\ AND\ Q_{0,k}^3 > 0 \end{cases}, k \in K$$

$$x_k = \begin{cases} 1, if\ Q_{0,k}^2 > 0 \\ 0, if\ Q_{0,k}^2 = 0 \end{cases}, k \in K$$

- Let $Dis\_veh = \sum_{k \in K} x_k + \sum_{k \in K} str_k$, be the number of vehicles that can transport partially disabled evacuees.
- Let $Initial\_Dis\_Dem = \sum_{c \in C} D_c^2$, be the total initial demand for partially disabled evacuees at all the pick-up locations.
- Let r_k be an indicator with

$$r_k = \begin{cases} 1, & if\ vehicle\ k \in K\ , is\ at\ s^k \\ 0, & otherwise \end{cases}$$

- Let $a = 3$ be a coefficient of capacity conversion.
- Let w_k be an indicator with $$w_k = \begin{cases} 1, & if Q_{0,k}^2\ , k \in K, has\ been\ converted\ to\ Q_{0,k}^1 \\ 0, & otherwise \end{cases}$$
- Let $Demand\_List^p$ be the list with demand of nodes D_i^p arranged in descending order, $p \in P$
- Let st^p be the service time of evacuees. In particular:

$$st^p = \begin{cases} 2, & p = 1 \\ 6, & p \epsilon \{2,3\} \end{cases}$$

- Let $\sum_{i \in C} Dis\_evac_i^2$ be the number of partially disabled evacuees that vehicle $k \in K$ collected at its last route
- Let $\sum_{i \in C} Dis\_evac_i^3$ be the number of totally disabled evacuees that vehicle $k \in K$ collected at its last route
- Let n_k be an indicator with

$$n_k = \begin{cases} 1, & if\ vehicle\ k \in K\ executes\ its\ first\ route \\ 0, & otherwise \end{cases}$$

- Let Dis_evac_i be the number of partially disabled evacuees that vehicle $k \in K$ collects from node $i \in C$
- Let $Nodes\_Array$ be an array with all nodes $i \in C$ for which $D_i^1 > 0$and at the time when $\sum_{c \in C} D_c^2 = 0$, a vehicle $k \in K$ with $Q_k^2 > 0$is about to collect evacuees from node *i*
- Let $Vehicles\_Array$ be an array with all vehicles $k \in K$ of type $x_k = 1$ with remaining capacity $Q_k^2 > 0$ which is converted to Q_k^1 and they serve node $i \in C$ with $D_{i \in C}^1 > 0$
- Let $Convertion\_Array$ be an array of vehicles $k \in K$ with $w_k = 0$

The proposed heuristic algorithm for this version of PEHFP is implemented using Matlab R2010b on a PC equipped with a 1.8 GHz Intel Core i5 and 4 GB of RAM. The pseudo code of the algorithm is given in the following:

Pseudocode

Step 1. Set $T_{evac} = 0, Time\_List = 0, Total\_Distance = 0, List = \emptyset$

Step 2.

Step 2.1 **If** $\sum_{k \in K} str_k > 0$ **AND** $\sum_{c \in C} D_c^3 > 0$

Step 2.1.1 - *Sort, first, all vehicles $k \in K$ with $str_k = 1$ in ascending order with respect to their initial capacity for partially disabled*

Step 2.1.2 - Sort vehicles with $str_k = 0$ in ascending order with respect to their initial capacity for enabled

Elseif $\sum_{k \in K} str_k = 0$ **AND** $\sum_{c \in C} D_c^3 > 0$

- Sort vehicles with $str_k = 0$ in ascending order with respect to their initial capacity for enabled

End

- Insert the sorted vehicles into $List$

Step 3.

Step 3.1 **While** $\sum_{c \in C} D_c^3 > 0$

Step 3.2 **If** $List$ is not empty

- Set the first vehicle $k \in K$ in $List$ with capacity Q_k^p as current vehicle ($CV = k$)
- Remove vehicle k from the $List$

Else

- Execute **Step 5**

End

Step 3.2.1 **If** $r_{CV}^p = 1$ **OR** node $= \{t\}$

Step 3.2.2 - Sort the demand of nodes for totally disabled in ascending order(Demand_List^3)

Step 3.2.3 **If** $Demand\_List^3(1) \neq Demand\_List^3(2) \neq \cdots \neq Demand\_List^3(z), z > 1$

- Set as current node (CN) the node i with demand $D_i = Demand\_List^3(1)$

Elseif $Demand\_List^3(1) = Demand\_List^3(2) = \cdots = Demand\_List^3(z), z > 1$

If $\sum_{i=1}^{z} D_i^2 > 0$ **AND** $Q_{CV}^2 > 0$

- Sort the demand of nodes for enabled in ascending order (Demand_List^2)

If $Demand\_List^2(1) \neq Demand\_List^2(2) \neq \cdots \neq Demand\_List^2(z), z > 1$

- Set as current node (CN) the node i with demand $D_i = Demand\_List^2(1)$

Elseif $Demand\_List^2(1) = Demand\_List^2(2) = \cdots = Demand\_List^2(z), z > 1$

- Find the nodes $i = 1,2,.,z$ with demand $D_i^2 = Demand\_List^2(1)$ and sort them in ascending order with respect to their distance from the starting location of CV (or from the shelter if CV is at the shelter) ($Nearest\_List$)
- Set $CN = Nearest\_List(1)$

End

Else

- Find the nodes $i = 1,2,.,z$ with demand $D_i^3 = Demand\_List^3(1)$ and sort them in ascending order with respect to their distance from the starting location of CV (or from the shelter if CV is at the shelter) ($Nearest\_List$)
- Set $CN = Nearest\_List(1)$

End

End

If $n_{CV} = 1$

Set $Time\_List(CV) = Time\_List(CV) + L_{S^{CV}CN}$

Set $Total\_Distance(CV) = Total\_Distance(CV) + P_{S^{CV}CN}$

Else

Set Time_List(CV) = Time_List$(CV) + L_{\{t\}CN}$

Set $Total\_Distance(CV) = Total\_Distance(CV) + P_{\{t\}CN}$

End

Elseif $node \neq \{t\}$

- Route vehicle CV to node l with $\min\{L_{nodel}, l \in C \backslash \{CN\}\}$ and demand $D_l^2 \neq 0$.

- Set $Time\_List(CV) = Time\_List(CV) + L_{node\ l}$
- Set $Total\_Distance(CV) = Total\_Distance(CV) + P_{nodel}$
- Set $CN = l$

End

Step 4.

Step 4.1 **If** $Q_{CV}^2 > 0$ **AND** $Q_{CV}^1 > 0$ **AND** $Q_{CV}^3 > 0$ **AND** $\sum_{c\in C} D_c^2 > 0$

- Load CV with $Q_{CV}^1 + Q_{CV}^3$ evacuees
- Update the demand of CN for enabled: $D_{CN}^1 \leftarrow D_{CN}^1 - Q_{CV}^1$
- Update the demand of CN for totally disabled: $D_{CN}^3 \leftarrow D_{CN}^3 - Q_{CV}^3$
- Set $Time\_List(CV) = Time\_List(CV) + Q_{CV}^3 * st^3$

Step 4.1.1 **If** $D_{CN}^2 \geq Q_{CV}^2$

- Load CV with Q_{CV}^2 evacuees
- Update the demand of CN for partially disabled: $D_{CN}^2 \leftarrow D_{CN}^2 - Q_{CV}^2$
- route vehicle CV to the shelter t to drop off the evacuees
- Set $Time\_List(CV) = Time\_List(CV) + Q_{CV}^3 * st^3 + (Q_{CV}^2 * 2 * st^2) + L_{CN\{t\}}$
- Set $Total\_Distance(CV) = Total\_Distance(CV) + P_{CN\{t\}}$

Elseif $D_{CN}^2 < Q_{CV}^2$

- Load CV with D_{CN}^2 evacuees
- Update the capacity of CV for partially disabled: $Q_{CN}^2 \leftarrow Q_{CV}^2 - D_{CN}^2$
- Set $Time\_List(CV) = Time\_List(CV) + D_{CN}^2 * st^2$
- Update the demand of CN for partially disabled: $D_{CN}^2 = 0$

End

Elseif $Q_{CV}^2 > 0$ **AND** $Q_{CV}^1 = 0$ **AND** $Q_{CV}^3 = 0$ **AND** $\sum_{c\in C} D_c^2 > 0$

- Go to *Step 4.1.1*.

Elseif $(Q_{CV}^2 = 0$ **AND** $Q_{CV}^1 > 0$ **AND** $Q_{CV}^3 > 0)$ **OR** $(Q_{CV}^2 > 0$ **AND** $Q_{CV}^1 > 0$ **AND** $Q_{CV}^3 > 0$ **AND** $\sum_{c\in C} D_c^2 = 0)$

- Load CV with $Q_{CV}^1 + Q_{CV}^3$ evacuees
- Update the demand of CN for enabled: $D_{CN}^1 \leftarrow D_{CN}^1 - Q_{CV}^1$
- Update the demand of CN for totally disabled: $D_{CN}^3 \leftarrow D_{CN}^3 - Q_{CV}^3$
- route vehicle CV to the shelter t to drop off the evacuees
- Set $Time\_List(CV) = Time\_List(CV) + (Q_{CV}^3 \cdot st^3 \cdot 2) + L_{CN\{t\}}$
- Set $Total\_Distance(CV) = Total\_Distance(CV) + P_{CN\{t\}}$

End

Step 4.2 **If** $\sum_{c\in C} D_c^2 = 0$ **AND** $\sum_{k\in K} str_k > 0$

- Route all vehicles $k \in K$ with $str_k = 1$ AND $Time\_List(k) \neq 0$ AND $node \neq \{t\}$ to the shelter
- Set $Time\_List(k) = Time\_List(k) + (\sum_{i\in C} Dis\_evac_i^3 \cdot st^3) + (\sum_{i\in C} Dis\_evac_i^2 \cdot st^2) + L_{node\{t\}}$
- Set $Total\_Distance(k) = Total\_Distance(k) + P_{node\{t\}}$

End

Step 5.

If $List$ is empty

- Insert to a new array the travel times of vehicles that can transfer totally disabled evacuees ($Time\_List\_Array$)
- Sort $Time\_List\_Array$ in ascending order

If $Time\_List\_Array(1) = Time\_List\_Array(2) = \ldots = Time\_List\_Array(z), z > 1$

If $\sum_{j=1}^{z} Q_{Time\_List\_Array(j)}^2 > 0$ AND $\sum_{c\in C} D_c^2 > 0$

- Set as current vehicle CV, the vehicle k with capacity $Q_k^2 = \max\{Q_{Time\_List(j)}^2, j = 1, 2, \ldots, z\}$.

Else

- Set as current vehicle CV, the vehicle k with capacity $Q_k^1 = \max\{Q_{Time\_List(j)}^1, j = 1, 2, \ldots, z\}$.

End

Elseif $Time\_List\_Array(1) \neq Time\_List\_Array(2) \neq \ldots \neq Time\_List\_Array(z), z > 1$

- Set the first vehicle (k, with capacity Q_k^1) as current vehicle CV

End

- Go to *Step 3.2.1*

Else

- Go to Step 3.2

End

End

Step 6.

Step 6.1 **If** $Dis\_veh \neq 0$ **AND** $\sum_{c \in C} D_c^2 > 0$

Step 6.1.1 - Sort, first, vehicles $k \in K$ with $Time\_List(k) = 0$ **AND** $x_k = 1$ in ascending order with respect to their capacity for partially disabled. In case that any of the vehicles for partially disabled have the same capacity for disabled, sort them in ascending order with respect to their initial capacity for enabled.

Step 6.1.2 - Sort the vehicles for enabled in descending order with respect to their initial capacity.

Step 6.1.3 - Insert the sorted vehicles to $List$

Step 6.2 **Elseif** $Dis\_veh \neq 0$ **AND** $\sum_{c \in C} D_c^2 = 0$

Step 6.2.1 - Convert the initial capacity of vehicles with $x_k = 1$ into capacity for enabled: $ICV_k^1 \leftarrow ICV_k^1 + (ICV_k^2 * a)$

Step 6.2.2 - Set $ICV_k^2 = 0$

Step 6.2.3 - Sort all vehicles in descending order with respect to their initial capacity.

Step 6.2.4 - Insert the sorted vehicles to $List$

Step 6.3 **Elseif** $Dis\_veh = 0$

- Sort all the vehicles in descending order with respect to their initial capacity.
- Insert the sorted vehicles to $List$

End

Step 7. **While** $\sum_{p=1}^{2} \sum_{c \in C} D_c^p > 0$

Step 8. **If** $List$ is not empty

Step 9. **If** $Dis\_veh \neq 0$ **AND** $\sum_{c \in C} D_c^2 > 0$

If $\sum_{c \in C} D_c^2 = 0$ **AND** $\sum_{c \in C} D_c^1 > 0$ **AND** $\sum_{k \in K} r_k^2 = 0$

- Set the first vehicle $k \in K$ in $List$ with capacity Q_k^1 as current vehicle ($CV = k$)
- Delete vehicle k from the $List$

Elseif $\sum_{c \in C} D_c^2 = 0$ **AND** $\sum_{c \in C} D_c^1 > 0$ **AND** $\sum_{k \in K} r_k^2 > 0$

- Find the vehicles $k \in K$ with $r_k^2 = 1$
- **Repeat *Steps 6.2.1-6.2.4***
- Set the first vehicle $k \in K$ in $List$ with capacity Q_k^p as current vehicle ($CV = k$)
- Delete vehicle k from the List

End

Elseif $Dis\_veh = 0$ **OR** ($Dis\_veh \neq 0$ **AND** $\sum_{c \in C} D_c^2 = 0$)

- Set the first vehicle $k \in K$ in $List$ with capacity Q_k^1 as current vehicle ($CV = k$)
- Delete vehicle k from the $List$

Elseif ($\sum_{c \in C} D_c^2 = 0$ **AND** $\sum_{c \in C} D_c^1 = 0$) **OR** ($\sum_{c \in C} D_c^2 > 0$ **AND** $\sum_{c \in C} D_c^1 = 0$ **AND** $\sum_{k \in K} r_k^2 = 0$)

- Set $List = \emptyset$

End

Elseif $List$ is empty

- Go to Step 13

End

Step 10.

If $r_{CV}^p = 1$ **OR** $node = \{t\}$

If $Dis\_veh \neq 0$ **AND** $x_{cv} = 1$ **AND** $\sum_{c \in C} D_c^2 > 0$

- Sort nodes' demand for disabled in descending order($Demand\_List^2$)

If $Demand\_List^2(1) \neq Demand\_List^2(2) \neq \cdots \neq Demand\_List^2(z), z > 1$

- Set node i with demand $D_i = Demand\_List^2(1)$ as current node (CN)

Elseif $Demand\_List^2(1) = Demand\_List^2(2) = \cdots = Demand\_List^2(z), z > 1$

If $\sum_{c \in C} D_c^1 > 0$

- Sort the demand of nodes for enabled in descending order $(Demand\_List^1)$
- Set node i with demand $D_i = Demand\_List^1(1)$ as current node (CN)

Else

- Find the nodes $i = 1,2,.,z$ with demand $D_i^2 = Demand\_List^2(1)$ and sort them in ascending order with respect to their distance from the starting location of CV (or from the shelter if CV is at the shelter) $(Nearest\_List)$
- Set $CN = Nearest\_List(1)$

End

End

Elseif $(Dis\_veh \neq 0$ **AND** $x_{cv} = 0)$ **OR** $Dis\_veh = 0$ **OR** $(Dis\_veh \neq 0$ **AND** $\sum_{c \in C} D_c^2 = 0)$

- Sort the demand of nodes for enabled in descending order$(Demand_List^1)$)

If $Demand\_List^1(1) \neq Demand\_List^1(2) \neq \cdots \neq Demand\_List^1(z), z > 1$

- Set node i with demand $D_i = Demand\_List^1(1)$ as current node (CN)

Elseif $Demand\_List^1(1) = Demand\_List^1(2) = \cdots = Demand\_List^1(z), z > 1$

If $Q_{cv}^1 > Demand\_List^1(1)$

- Find the nodes $i = 1,2,.,z$ with demand $D_i^1 = Demand\_List^1(1)$ and sort them in ascending order with respect to their distance from the starting location of CV (or from the shelter if CV is at the shelter)$(Furthest\_List)$
- Set $CN = Furthest\_List\ (1)$

Elseif $Q_{cv}^1 \leq Demand\_List^1(1)$

- Find the nodes $i = 1,2,.,z$ with demand $D_i^1 = Demand\_List^1(1)$ and sort them in ascending order with respect to their distance from the starting location of CV (or from the shelter if CV is at the shelter) $(Nearest\_List)$
- Set $CN = Nearest\_List(1)$

End

End

End

If $n_{CV} = 1$

- Set $Time\_List(CV) = Time\_List(CV) + L_{S^{cv}CN}$
- Set $Total\_Distance(CV) = Total\_Distance(CV) + P_{S^{cv}CN}$

Else

- Set $Time\_List(CV) = Time\_List(CV) + L_{\{t\}CN}$
- Set $Total\_Distance(CV) = Total\_Distance(CV) + P_{\{t\}CN}$

End

Elseif $node \neq \{t\}$

Set $CN = node$

If $x_{cv} = 1$ **AND** $\sum_{c \in C} D_c^2 > 0$ **AND** $Q_{CV}^2 > 0$

- Route vehicle CV to node l with $\min\{L_{CNl}, l \in C \backslash \{CN\}\}$ and demand $D_l^2 \neq 0$.
- Set $\text{Time\_List}(CV) = \text{Time\_List}(CV) + L_{CN\,l}$
- Set $Total\_Distance(CV) = Total\_Distance(CV) + P_{CNl}$
- Set $CN = l$

Elseif $x_{cv} = 0$ **OR** $(x_{cv} = 1$ **AND**$(\sum_{c \in C} D_c^2 = 0$ **OR** $Q_{CV}^2 = 0))$

- Route vehicle CV to node l with $\min\{L_{CNl}, l \in C \backslash \{CN\}\}$ and demand $D_l^1 \neq 0$.
- Set $\text{Time\_List}(CV) = \text{Time\_List}(CV) + L_{CN\,l}$
- Set $Total\_Distance(CV) = Total\_Distance(CV) + P_{CNl}$
- Set $CN = l$

End

End

Step 11.

Step 11.1 **If** $Dis\_veh \neq 0$ AND $Q_{CV}^2 > 0$ AND $\sum_{p=1}^{2} \sum_{c \in C} D_c^p > 0$

Step 11.1.1 **If** $D_{CN}^2 \geq Q_{CV}^2$ **AND** $D_{CN}^1 \geq Q_{CV}^1$

- load vehicle CV with $Q_{CV}^1 + Q_{CV}^2$ evacuees
- update the demand of CN for enabled : $D_{CN}^1 \leftarrow D_{CN}^1 - Q_{CV}^1$
- update the demand of CN for disabled :$D_{CN}^2 \leftarrow D_{CN}^2 - Q_{CV}^2$
- route vehicle CV to the shelter t to drop off the evacuees
- Set $Total\_Distance(CV) = Total\_Distance(CV) + P_{CN\{t\}}$
- Set

$$Time\_List(CV) = Time\_List(CV) + L_{CN\,\{t\}} + Q_{CV}^2 \cdot 2st^2 + \sum_{i \in C} Dis_{evac_i}{}^3 \cdot st^3 + \left(\sum_{i \in C} Dis_{evac_i}{}^2 - Q_{CV}^2\right) \cdot st^2$$

Elseif $D_{CN}^2 < Q_{CV}^2$ **AND** $D_{CN}^1 < Q_{CV}^1$

- load vehicle CV with $\sum_{p \in \{1,2\}} D_{CN}^p$
- update the capacity of vehicle CV:$Q_{CV}^p \leftarrow Q_{CV}^p - \sum_{p \in \{1,2\}} D_{CN}^p$
- Set $\mathrm{Time\_List}(CV) = \mathrm{Time\_List}(CV) + D_{CN}^2 * st^2$
- update the demand of node CN: $\sum_{p \in \{1,2\}} D_{CN}^p = 0$

Elseif $D_{CN}^2 \geq Q_{CV}^2$ **AND** $D_{CN}^1 < Q_{CV}^1$

- load vehicle CV with Q_{CV}^2 evacuees and with D_{CN}^1 evacuees
- update the capacity of vehicle $CV$for enabled:$Q_{CV}^1 \leftarrow Q_{CV}^1 - D_{CN}^1$
- update the demand of CN for enabled: $D_{CN}^1 = 0$
- update the demand of CN for disabled:$D_{CN}^2 \leftarrow D_{CN}^2 - Q_{CV}^2$
- Set $\mathrm{Time\_List}(CV) = \mathrm{Time\_List}(CV) + Q_{CV}^2 * st^2$
- update the capacity of vehicle $CV$for disabled:$Q_{CV}^2 = 0$

Elseif $D_{CN}^2 < Q_{CV}^2$ **AND** $D_{CN}^1 \geq Q_{CV}^1$

- load vehicle CV with D_{CN}^2 evacuees and with Q_{CV}^1 evacuees
- update the demand of CN for enabled:$D_{CN}^1 \leftarrow D_{CN}^1 - Q_{CV}^1$
- update the capacity of CV for enabled: $Q_{CV}^1 = 0$
- update the capacity of CV for disabled: $Q_{CV}^2 \leftarrow Q_{CV}^2 - D_{CN}^2$
- Set $\mathrm{Time\_List}(CV) = \mathrm{Time\_List}(CV) + D_{CN}^2 * st^2$
- update the demand of CN for partially disabled: $D_{CN}^2 = 0$

End

If $\sum_{c \in C} D_c^2 = 0$

Step 11.1.2 - Find all the vehicles $k = 1,2,\ldots,u$ with $x_k = 1$ and $w_k = 0$ and convert their capacity for partially disabled to capacity for enabled ($Convertion\_Array$).

While $Convertion\_Array$ is not empty

- Set CV= $Convertion\_Array(1)$
- Set $Convertion\_Array(1) = \emptyset$
- Find the last node($node$) that CV visited during its last route

If $node \neq \{t\}$

- Update $CV's$ initial capacity for disabled: $ICV_{CV}^2 = ICV_{CV}^2 - Q_{CV}^2$
- Update $CV's$ initial capacity for enabled: $ICV_{CV}^1 = ICV_{CV}^1 + (Q_{CV}^2 * a)$
- Convert $CV's$ capacity for disabled to capacity for enabled and update $CV's$ capacity for enabled: $Q_{CV}^1 = Q_{CV}^1 + (Q_{CV}^2 * a)$
- Set $Q_{CV}^2 = 0$

If $D_{node}^1 > 0$

- Insert $node$ to $Nodes\_Array$
- Insert CV to $Vehicles\_Array$

End

Elseif $node = \{t\}$

- Update $CV's$ initial capacity for enabled: $ICV_{CV}^1 = ICV_{CV}^1 + (Q_{CV}^2 * a)$
- Set $ICV_{CV}^2 = 0$
- Set $Q_{CV}^p = ICV_{CV,}^p\ p \in \{1,2\}$

End

End

Step 11.1.3 **While** $Nodes\_Array$ is not empty

- Set $CN = Nodes\_Array\,(1)$
- Delete CN from $Nodes\_Array$
- Set $CV = Vehicles\_Array(1)$
- Delete CV from $Vehicles\_Array$

Step 11.1.4 **If** $D_{CN}^{1} \geq Q_{CV}^{1}$

- load CV with Q_{CV}^{1} evacuees
- update the demand of CN: $D_{CN}^{1} = D_{CN}^{1} - Q_{CV}^{1}$
- route vehicle CV to the shelter t to drop off the evacuees
- Set $Total\_Distance(CV) = Total\_Distance(CV) + P_{CN\{t\}}$

If $\sum_{i\in C} Dis\_evac_i = 0$

- Set Time_List(CV) = Time_List$(CV) + st^1 + L_{CN\{t\}}$

Else

- Set Time_List(CV) = Time_List$(CV) + (\sum_{i\in C} Dis_{evac_i}^{\ 2} * st^2) + L_{CN\{t\}}$

End

Elseif $D_{CN}^{1} < Q_{CV}^{1}$

- load CV with D_{CN}^{1} evacuees
- update the capacity of CV: $Q_{CV}^{1} \leftarrow Q_{CV}^{1} - D_{CN}^{1}$
- Set $D_{CN}^{1} = 0$

If $Dis\_evac_{CN} = 0$

- Set Time_List(CV) = Time_List$(CV) + st^1$

Else

- Set Time_List(CV) = Time_List(CV)

End

End

End

End

Step 11.2 **Elseif** $x_{cv} = 0$ **OR (** $x_{cv} = 1$ **AND** $Q_{CV}^{2} = 0)$

Go back to *Step* 10.1.4

Step 11.3 **Elseif** $(str_{CV} = 1$ **OR** $x_{cv} = 1)$ **AND** $\sum_{c\in C} D_c^2 > 0$ **AND** $\sum_{c\in C} D_c^1 = 0$

Step 11.3.1 **If** $D_{CN}^{2} \geq Q_{CV}^{2}$

- Load CV with Q_{CV}^{2} evacuees
- update the demand of CN: $D_{CN}^{2} = D_{CN}^{2} - Q_{CV}^{2}$
- route vehicle CV to the shelter t to drop off the evacuees
- Set Time_List(CV) = Time_List$(CV) + (Q_{CV}^{2} * st^2 * 2) + L_{CN\{t\}} + \sum_{i\in C} Dis_{evac_i}^{\ 3} \cdot st^3$

Elseif $D_{CN}^{2} < Q_{CV}^{2}$

- load CV with D_{CN}^{2} evacuees
- update the capacity of CV: $Q_{CV}^{2} \leftarrow Q_{CV}^{2} - D_{CN}^{2}$
- Set Time_List(CV) = Time_List$(CV) + (D_{CN}^{2} * st^2)$
- Set $D_{CN}^{2} = 0$

End

End

Step 12.

If $\sum_{c\in C} D_c^2 > 0$ **AND** $\sum_{c\in C} D_c^1 = 0$

- Route all the vehicles $k \in K$ for enabled to the shelter
- Set Time_List(k) = Time_List$(k) + st^1 + L_{node\,\{t\}}$
- Route all the vehicles $k \in K$ for disabled with capacity $Q_{CV}^{2} = 0$ to the shelter

If $\sum_{i\in C} Dis\_evac_i^3 > 0$

- Set Time_List(k) = Time_List$(k) + (\sum_{i\in C} Dis_{evac_i}^{\ 2} * st^2) + (\sum_{i\in C} Dis_{evac_i}^{\ 3} * st^3) + L_{node\,\{t\}}$

Else

- Set Time_List(k) = Time_List$(k) + (\sum_{i\in C} Dis_{evac_i}^{\ 2} * st^2) + L_{node\,\{t\}}$

End

End

- Set $Total\_Distance(\boldsymbol{k}) = Total\_Distance(\boldsymbol{k}) + P_{node\,\{t\}}$

Step 13.

If $List = \emptyset$

If $\sum_{c\in C} D_c^2 = 0$

Repeat *Steps 11.1.2-11.1.4*

- Sort the $Time\_List$ in ascending order
- Insert to a new array the travel times of vehicles with $x_k = 1$ ($Time\_List\_Array$)
- Sort $Time\_List\_Array$ in ascending order

If $Time\_List\_Array(1) = Time\_List\_Array(2) = \ldots = Time\_List\_Array(z), z > 1$

- Set as current vehicle CV, the vehicle k with capacity $\mathrm{Q}_\mathrm{k}^1 = \max\{Q^1_{Time\_List\_Array(j)}, j = 1, 2, \ldots, z\}$.

Else

- Set the first vehicle (k, with capacity Q_k^1) as current vehicle CV

End

Elseif $\sum_{c\in C} D_c^1 = 0$

- Insert to a new array the travel times of vehicles with $wc_k = 1$ ($Time\_List\_Array$)
- Sort $Time\_List\_Array$ in ascending order

If $Time\_List\_Array\,(1) = Time\_List\_Array\,(2) = \ldots = Time\_List\_Array\,(z), z > 1$

- Set as current vehicle CV, the vehicle k with capacity:
- $\mathrm{Q}_\mathrm{k}^2 = \max\{Q^2_{Time\_List\_Array\,(j)}, j = 1, 2, \ldots, z\}$.

Else

- Set the first vehicle (k, with capacity Q_k^2) as current vehicle CV

End

Elseif $\sum_{c\in C} D_c^2 > 0$ **AND** $\sum_{c\in C} D_c^1 > 0$

- Sort the $Time\_List$ in ascending order

If $Time\_List(1) = Time\_List(2) = \ldots = Time\_List(z), z > 1$

If $\sum_{j=1}^{z} wc_j > 0$

- Set as current vehicle $\boldsymbol{CV}$, the vehicle k with capacity:
- $\mathrm{Q}_\mathrm{k}^2 = \max\{Q^2_{Time\_List(j)}, j = 1, 2, \ldots, z\}$.

Else

- Set as current vehicle CV, the vehicle k with capacity:
- $\mathrm{Q}_\mathrm{k}^1 = \max\{Q^1_{Time\_List(j)}, j = 1, 2, \ldots, z\}$.

End

Else

- Set the first vehicle (k, with capacity $\mathrm{Q}_\mathrm{k}^\mathrm{p}$) as current vehicle CV

End

End

- **Go to step 10**

Elseif $List \neq \emptyset$

- **Go to step 8**

End

End

Step 14.

- Find all the vehicles $k \in K$ of which $node \neq \{t\}$ and route them to the shelter

If $\sum_{i\in C} Dis_{evac_i}^{\;2} = 0$ **AND** $\sum_{i\in C} Dis_{evac_i}^{\;3} = 0$

- Set $\mathrm{Time\_List}(k) = \mathrm{Time\_List}(k) + L_{node\,\{t\}} + st^1$

Else

- Set $\mathrm{Time\_List}(k) = \mathrm{Time\_List}(k) + L_{node\,\{t\}} + (\sum_{i\in C} Dis_{evac_i}^{\;2} * st^2) + (\sum_{i\in C} Dis_{evac_i}^{\;3} * st^3)$

End

- Set $Total\_Distance(k) = Total\_Distance(k) + P_{node\,\{t\}}$

Step 15.

- Find $maxelement\{Evac\_time\}$
- Set $T_{\boldsymbol{evac}} = maxelement\{Evac\_time\}$
- Set $Distance = \sum_{k\in K} Total\_Distance(k)$

Appendix D: Input data for the case study

The following tables present the necessary data in terms of a) the number of evacuees per village, b) the transportation network that links the villages with the shelter, the transportation network between villages, the transportation network that links each vehicle's starting point with the villages, c) the public and private fleet of vehicles available for the evacuation, for the PEHFP.

D.1 Evacuees

Table D1, presents the population of each village that need to be evacuated. The evacuees are categorized as follows: a) enabled evacuees that will be transported via buses, 4x4 vehicles, and vans, b) disabled evacuees with total disability that will be transported via ambulances and emergency mobile units and, c) disabled evacuees with partial disability that will be transported via vans or ambulances (if needed).

Table D1. *List of evacuees per village*

Village	Village ID	Enabled Evacuees	Disabled Evacuees (with total disability)	Disabled Evacuees (with partial disability)
Tramacastiel	100	37	1	6
Rubiales	200	26	1	4
El Campillo	300	33	1	6
Teruel (Shelter)	1000			
Villel (Shelter)	2000			

D.2 Road Network

Table D2, presents the road network that connects the villages with the shelter, the villages themselves and each vehicle's starting point with each village. In case study of Teruel the starting point of each vehicle is the same with the shelter (each vehicle starts its trip from city of Teruel and returns to it to drop off the evacuees). The routes presented in Table A2 are the best, based on the available road network of the area. Figure D1, depicts the best routes using Google maps.

Table D2. *Transport Network (best routes) between villages and shelter*

	Distances (in Km) & travel times of best Routes (in min)									
To	**Shelter (Teruel)**		**Tramacastiel**		**Rubiales**		**El Campillo**		**Villel**	
From	***Min***	***km***	***min***	***km***	***min***	***km***	***min***	***km***	***min***	***km***
Shelter (Teruel)	-		38	32.8	27	21.4	16	15.1	-	
Tramacastiel	38	32.8	-		62	52.8	51	46.3	23	17.8
Rubiales	27	21.4	62	52.8	-		15	6.9	-	
El Campillo	16	15.1	51	46.3	15	6.9	-		-	
Villel	-		23	17.8	-		-		-	

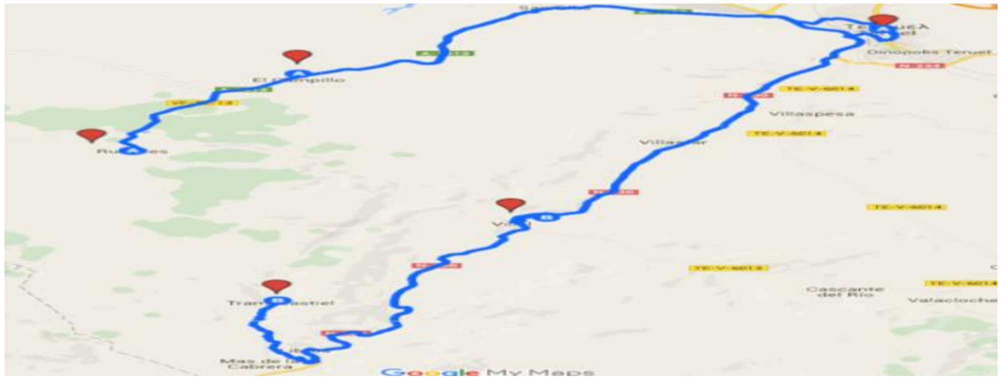

Figure D1. *Road network between evacuation problems nodes*

D.3 Public Vehicles

Table D3, presents the fleet of public vehicles that will be that will be available during the evacuation process. As it can be obtained, there are various types of vehicles available (e.g. cars, vans, etc.) that can be used only for enabled citizens. Furthermore, the table presents the number of each type of vehicle that is available and its capacity (seats). Last but not least, the starting point (depot) of each vehicle is given.

Table D3. *List of public vehicles available*

Type Of Vehicle	Number of Each Type Of Vehicle	Vehicle ID	Capacity Per Vehicle(seats)	Adapted for Disabled People	Starting Point			Company Name
					Address	Number	City	
Car 4x4	1	5	4	No	Temprado	4	Teruel	Agrupación de Voluntarios de Protección Civil Comarca Comunidad de Teruel
Van	1	6	7	No	Calle Temprado	3	Teruel	Comarca Comunidad de Teruel
Van	1	7	8	No	Polígono La Paz, Calle Berlín	s/n	Teruel	Diputación de Teruel - Parque Maquinaria
Van	1	8	8	No	Polígono La Paz, Calle Berlín	s/n	Teruel	Diputación de Teruel - Parque Maquinaria
Van	1	9	8	No	Polígono La Paz, Calle Berlín	s/n	Teruel	Diputación de Teruel - Parque Maquinaria
Patrol Car	1	10	4	No	Plaza la Catedral	1	Teruel	Ayuntamiento de Teruel
Patrol Car	1	11	4	No	Plaza la Catedral	1	Teruel	Ayuntamiento de Teruel
Patrol Car	1	12	4	No	Plaza la Catedral	1	Teruel	Ayuntamiento de Teruel
Patrol Car 4x4	1	13	4	No	Plaza la Catedral	1	Teruel	Ayuntamiento de Teruel
Patrol Car 4x4	1	14	4	No	Plaza la Catedral	1	Teruel	Ayuntamiento de Teruel

D.4 Private Vehicles

Table D4, presents the fleet of private vehicles that will be available during the evacuation process. Note that there are various type of vehicles available (e.g. common ambulance, emergency mobile unit, bus, etc.) that can be used for both enabled and partially disabled citizens, vehicles that can be used for enabled citizens and ambulances that can be used for both totally and partially disabled citizens. Furthermore, the table presents the number of each type of vehicle that is available, the capacity per vehicle as well as whether a vehicle is adapted for disabled people and its capacity for this category of people. Finally, the starting point (depot) of each vehicle is given.

Table D4. *List of private vehicles available*

Type Of Vehicle	Number of Each Type Of Vehicle	Vehicle's ID	Capacity of Each Vehicle			Starting Point			Company Name
			Enabled	Partially Disabled	Totally Disabled	Address	Number	City	
Common Ambulance	1	15	2	0	1	Polígono Los Hostales	Nave 1	Teruel	Transportes Sanitarios de Teruel S.L.
Common Ambulance	18	16-33	2	0	1	Polígono La Paz, Irún, Parcela 166	-	Teruel	Ambuiberica S.L.
Common Ambulance	3	34-36	2	0	1	Polígono Los Hostales	Nave 1	Teruel	Transportes Sanitarios de Teruel S.L.
Basic Life Support	1	37	1	0	1	Polígono Los Hostales	Nave 1	Teruel	Transportes Sanitarios de Teruel S.L.
Ambulance	2	38-39	1	0	1	San Miguel	3	Teruel	Cruz Roja Española
Emergency Mobile Unit	1	40	1	0	1	Polígono La Paz, Irún, Parcela 166	-	Teruel	Ambuiberica S.L.
Emergency Mobile Unit	2	41-42	1	0	1	Polígono La Paz, Irún, Parcela 166	-	Teruel	Ambuiberica S.L.
Colective Ambulance	8	43-50	1	2	1	Polígono La Paz, Estocolmo	13 B	Teruel	Nuevos Transportes Sanitarios de Aragón

Type Of Vehicle	Number of Each Type Of Vehicle	Vehicle's ID	Capacity of Each Vehicle			Starting Point			Company Name
			Enabled	Partially Disabled	Totally Disabled	Address	Number	City	
Colective Ambulance 4x4	2	51-52	1	1	1	Polígono La Paz, Estocolmo	13 B	Teruel	Nuevos Transportes Sanitarios de Aragón
Bus	2	53-54	55	0	0	Croacia	4	Teruel	Autocares Nolasco
Bus	1	55	22	0	0	Croacia	4	Teruel	Autocares Nolasco
Bus	8	56-63	50	0	0	Polígono Los Hostales, Nave 1-4	-	Teruel	Autobuses Teruel-Zaragoza, S.A.
Bus	2	64-65	55	0	0	Polígono Los Hostales, Nave 1-4	-	Teruel	Autobuses Teruel-Zaragoza, S.A.
Bus	2	65-67	22	0	0	Carretera Cubla	3	Teruel	Auto Transportes Teruel S.L.
Bus	4	68-71	55	0	0	Carretera Cubla	3	Teruel	Auto Transportes Teruel S.L.
Small Truck	1	72	9	1	0	San Miguel	3	Teruel	Cruz Roja Española
Small Truck 4X4	1	73	9	1	0	San Miguel	3	Teruel	Cruz Roja Española
Minibus	1	74	22	3	0	San Miguel	3	Teruel	Cruz Roja Española