POLITECNICO DI MILANO

Scuola di Ingegneria dei Sistemi
Corso di Laurea Magistrale in Ingegneria Matematica
Dipartimento di Matematica

An ILS algorithm to solve a real Home Health Care Routing Problem

Relatore: Prof. Federico Malucelli
Correlatore: Prof.ssa Helena Ramalhinho Dias Lourenço

Tesi di Laurea di:
Marta Galvani
Matr. 818744

Anno Accademico 2014-2015

Abstract

Home Health Care (HHC) is defined as a set of ”medical and paramedical services delivered to patients at home”.
A patient in a hospital has a high cost for the community, so the main benefit of the HHC service is the significant decrease in the hospitalization rate that leads to costs reduction in the whole health system; this is important especially if the patient needs service only a few minutes a day. In addition the quality of life perceived by patients that stay at home is higher than if they stay at the hospital.
For these reasons the current trend in Europe and North America is to send nurses to visit patients in their home in order to reduce costs for the community and increase patients quality of life.
During the last decade the Health Care service industries experienced significant growth in many European countries due to the governmental pressure to reduce healthcare costs, the demographic changes and the development of new services and technologies. To take into account these challenges, Health Care decision problems have to be solved for Home Care services by rigorous methodologies involving Operations Research and Computer Science techniques.
In this work we try to solve a real case of Home Health Care Problem observed in the city of Ferrara, Italy. The idea is to create a weekly schedule for the operators which specifies where and when a nurse has to go on each day of the week and which service has to be performed. Main objectives of the optimization problem are to minimize the costs, maximize the patient-nurse loyalty ratio and the workload balance between nurses while satisfying the weekly requests.
As the problem is NP-hard, due to the great number of constraints and objectives that must be considered, in literature it is often reduced to known problems. The goal of this thesis is to solve the problem considering most of its realistic aspects. An algorithm based on the Iterated Local Search (ILS) method is applied in 60 different instances. All the subparts of the algorithm are explained.
As the problem can be seen as a multiobjective optimization problem, the ILS algorithm is then extended introducing the concept of Pareto optimal solutions. At the end of the algorithm, a pool of approximate Pareto optimal solutions is presented.

Finally, stochastic service times are introduced to make the solution more realistic. Simulated Iterated Local Search (SimILS) is merged with the ILS for multiobjective problems to obtain a new algorithm, developed in this thesis, that can be called MultiObjective Simulated Iterated Local Search, MOSimILS.
Each algorithm consists of about 900 Python lines. Obtained results encourage to keep on studying in this direction.

Sommario

L'assistenza sanitaria a domicilio viene definita come un insieme di ”servizi medici e paramedici erogati al paziente a casa”.
Poiché un paziente in ospedale ha un alto costo per la comunità, l'assistenza sanitaria domiciliare ha come principale beneficio la significativa decrescita della percentuale di ospitalizzazione che porta a una riduzione dei costi nell'intero sistema sanitario; questo è importante soprattutto se un paziente necessita assistenza solo pochi minuti al giorno. Inoltre la percezione della qualità di vita è più elevata nei paziente a cui è consentito restare nella propria abitazione, rispetto a chi si trova in ospedale.
Per queste ragioni il trend corrente in Europa e Nord America è quello di mandare le infermiere ad assistere i pazienti nelle proprie abitazioni con lo scopo di ridurre i costi per la comunità e aumentare la qualità di vita del paziente.
Nell'ultimo decennio le compagnie di assistenza domiciliare hanno sperimentato una significativa crescita in molti paesi Europei a causa delle pressioni governative per la riduzione dei costi di assistenza sanitaria, dei cambiamenti demografici e dello sviluppo di nuovi servizi e tecnologie. Per prendere in considerazione questi cambiamenti i problemi di assitenza domiciliare devono essere affrontati tramite l'utilizzo di metodologie rigorose che coinvolgono tecniche che vanno dall'Informatica alla Ricerca Operativa.
In questo lavoro cercheremo di risolvere un caso reale di assistenza sanitaria domiciliare osservato nella città di Ferrara, Italia. L'idea è quella di creare una pianificazione settimanale degli operatori che specifichi, per ogni infermiera disponibile, dove e quando si debba recare ogni giorno della settimana e quali servizi debba svolgere. Gli obiettivi principale del problema sono la minimizzazione dei costi, la massimizzazione della fedeltà paziente-infermiera e il bilanciamento del carico di lavoro tra infermiere, rispettando la richiesta settimanale di servizi.
Essendo il porblema considerato NP-difficile, a causa dell'elevato numero di vincoli e obiettivi che devono essere considerati, in letteratura è stato spesso ridotto a problemi noti. L'obiettivo di questa tesi è quello di risolvere il problema senza trascurare i suoi aspetti reali. Con questo scopo un algoritmo basato sul metodo di Iterated Local Search è stato applicato su 60 istanze differenti. Tutte le sottoparti

dell'algoritmo vengono chiaramente spiegate nel corso di questa tesi.
Poiché il problema considerato può essere visto come un problema multiobiettivo, l'algoritmo ILS è stato quindi successivamente esteso introducendo il concetto di soluzioni Pareto Ottime. Alla fine dell'algoritmo viene presentato un insieme di soluzioni di approssimazione della frontiera Pareto Ottima.
In conclusione per avere soluzioni più realistiche sono stati introdotti tempi di servizio stocastici. L'algoritmo di Simulated Iterated Local Search (SimILS) è stato integrato con l'ILS per problemi multiobiettivo così da ottenere un nuovo algoritmo, sviluppato in questa tesi, che chiameremo Multiobjective Simulated Iterated Local Search, MoSimILS.
Ogni algoritmo presentato consiste in circa 900 righe di codice Python. Le soluzioni ottenute incoraggiano a continuare gli studi in questa direzione.

Ringraziamenti

Desidero ricordare tutti coloro che mi hanno aiutata nella stesura delle tesi con suggerimenti, critiche osservazioni e anche tutti coloro che ho avuto vicino in questi anni universitari.

Ringrazio anzitutto il professor Federico Malucelli, Relatore, e la professoressa Helena Ramalhinho Dias Lourenço, Correlatore, per la grande disponibilità e cortesia prestata, per il tempo che mi hanno dedicato sia di persona che da lontano.

Un ringraziamento speciale va ai miei genitori, Elena e Mauro, che mi hanno sempre sostenuta sia moralmente che economicamente in questi anni, stimolandomi e appoggiandomi in ogni mia iniziativa. Grazie alle mie sorelle, Flora e Carla, per aver sempre condiviso con me successi e insuccessi.
Ringrazio i miei zii, Laura e Franco, e i miei cugini, Eugenio e Alfredo, che mi hanno accolta nella loro casa in questi anni di studio, circondadomi del loro affetto e rendendo ancora più forte e solido il nostro legame.

Ringrazio il mio ragazzo Luca che in questi due anni di specialistica non solo non mi ha mai fatto mancare nulla, ma è stato per me anche un valido confronto dal punto di vista universitario.
Ringrazio in special modo la mia amica Chiara, per la sua costante comprensione e appoggio, e insieme a lei le mie amiche Sofia, Sarah e Stefania, che mi hanno sempre fornito il loro sostegno e che condividono con me un'amicizia iniziata ben prima dei miei studi universitari.

Un ultimo ringraziamento ai compagni di studi, in particolar modo a Chiara, Cristina, Ilaria e Jacopo, per essermi stati vicini sia nei momenti difficili, sia nei momenti felici: sono stati per me più veri amici che semplici compagni. E a Vittorio che ha sempre saputo consigliarmi la strada in questi ultimi anni e che ha dedicato del tempo alla correzione di questa tesi.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 Motivation and challenges

Home Health Care service consists of assistance provided by medical staff, such as nurses, physical therapists and home care aides, to people with special needs, for example old adults, chronically ills or disabled people.
The main reasons for the development of HHC service are that quality of life perceived by patients who stay home, that is higher than if they stay at the hospital, and the high cost of the hospitalization. On one side, people prefer to stay in their own house even if, for some reasons, they are not able to take care of themselves. On the other side, a patient in a hospital has a high cost for the community, so the main benefit of the Home Health Care service is the significant decrease in the hospitalization rate that leads to costs reduction in the whole health system; this is important especially if the patient needs service only a few minutes a day.
During the last decade the Health Care service industries experienced significant growth in many European countries due to the governmental pressure to reduce healthcare costs, the demographic changes and the development of new services and technologies.
The complexity of the problem is mainly due to the large number of assisted patients, to the limited resources and to the service delivery in an often vast territory. To not waste time and to avoid process inefficiencies, Health Care decision problems have to be solved by rigorous methodologies involving Operations Research and Computer Science techniques. The main goal of optimizing the HHC service is to provide a high level of service while maintaining low cost.
We consider a particular application of this service in the city of Ferrara. Planning is nowadays mostly done by hand, usually by experienced nurses, but the fast growing of requests makes this procedure more complex. Service is activated by a doctor to a patient, and therapies are decided with nurses specifying the type

and frequency in a week. Week planning decides the sequence of services a nurse needs to perform in each day of the week.
It can be observed that Home Health Care is an optimization problem with many constraints often depending on specific cases and with the objective of finding a feasible solution while minimizing costs and maximizing the service quality.
A good solution to HHCP has to satisfy all the requests with the constraints of available human resources and working time limit, and at the same time has to obtain some goals such as the minimization of costs, the maximization of loyalty between nurse and patient and the balance of workload between nurses. Deterministic assumptions are one of the most common simplifications of the problem.
In this work we first try to solve the deterministic version of this problem using Iterated Local Search algorithm maintaining all the constraints to fit as better as possible the real problem. The main challenge is to solve the problem as a multiobjective one considering the three main objective functions of HHCP and then introducing stochastic service time studying a more realistic case.
We then extend the ILS for the deterministic version to the multiobjective and stochastic problem. This constitutes the main contribution of this thesis.

1.2 Outlines

The remainder of the thesis is structured as follows.

Chapter 2 gives a general idea about what Home Care and Home Health Care services are: what they consist of and which part of the problem can be considered as optimization problems. A list of works about HHCP is given. Then the home health care service in Ferrara hospital is introduced explaining which kind of data and constraints we have to consider, and which are the main goals of the problem. A mathematical model is presented for our problem.

Chapter 3 introduces the Iterated Local Search method, explaining its construction and mentioning some applications. The algorithm is then applied to Home Health Care optimization problem considering as objective function the sum of costs, expressed in time, and the loyalty of the problem, as a summation of different nurses visiting the same patient. The ILS algorithm has three main phases: construct an initial solution, local search method and a perturbation phase. In this chapter we explain each phase in details.

Chapter 4 defines HHCP as a multiobjective problem with three main goals: minimization of costs, maximization of loyalty and maximization of the balance of workload between nurses. In this way we avoid to sum quantities that are incomparable such as time and number of nurses. The problem is still solved with ILS algorithm but after introducing the concept of Pareto optimal solution. A pool of Pareto optimal solutions is used to choose pseudo randomly the solution to perturb in the ILS procedure and then the pool is updated considering all the solutions found during the Local Search.

Chapter 5 introduces the stochastic service times to be more closed to the real problem. The SimILS algorithm is merged with the multiobjective ILS in order to solve the stochastic multiobjective HHCP. A simulation with a triangular distribution is applied to the service times in order to have a simulated value of the objective functions. The idea is to create and update the pool of approximate Pareto optimal solutions using the stochastic values and not the deterministic ones.

Chapter 2

Home Care and Health Care services

In this chapter Home Care and Home Health Care services are introduced. Main aspects of these services are explained and differences between them are outlined. Some literature about HHCP is mentioned. Finally the real Home Heath Care problem in the city of Ferarra, taken into account in thesis, is explained, focusing on the objectives and constraints we have to consider solving it as an optimization problem. At the end of the chapter a mathematical model for the HHCP is formulated.

2.1 Home Care service

Home care service means a care that it is performed at patient's house. This service allows a person that needs a special treatment to stay in his/her home. Most people prefer to remain in their own home, however situations and conditions may come up, preventing people to take care of themselves. Sometimes, it can be necessary to take care of these people in an institution, but when it is possible, it is better to let them stay at home. This service is usually needed by people who are getting older, chronically ill, recovering from a surgery or disabled. So home care is especially useful for people who need nursing, therapies or aide services. People may need care service if:

- they have trouble getting around;
- they have wounds that need to be cleaned, or they need injections or other treatments;
- they need to learn more about their medical conditions and how to monitor them;

- they need help with bathing, dressing and meal preparation;
- they need care emotional support when in the final stages of an incurable disease.

So Home care services can include many different types of services, for example:

- health care, such as having a home health aide come to your home for usually medical services;
- personal care, such as help with bathing, washing your hair or getting dressed;
- homemaking, such as cleaning, yard work and laundry;
- cooking or delivering meals.

In other words one can get almost any type of help he/she needs in his/her home.
Some types of care and community services are free or donated, many other types must be paid for. People that need help can get it in many different ways. First of all from people they know, as family, friends and neighbours: these are the biggest source of help for many older or disabled people. Second, they can get help from the community of local government resources, as for instance from religious group or from social services. The last possibility is to ask to a private agency or to a hospital.
So besides family members and friends many different types of care providers can come to help people with movement and exercises, wound care and daily living. Home health care nurses can help manage problems with their wounds, other medical problems and any medications that they may be taking. Physical and occupational therapists can make sure their home is set up so that it will be easy and safe to move around and take care of themselves.
As home care is especially needed by elderly people and as the life expectation has increased in recent decades, contributing to the ageing of the population, we can assume that the request of home care services will grow very fast.
Of course home care has some limitations. Sometimes it can happen that some situations or conditions may make care in an institution a better choice than home care. For example, caregivers may not be available to adequately address the needs of the older person or caregiver burnout and stress may prevent continued safe care for people in the home. Also, serious medical situations that require frequent testing, breathing treatments, or intravenous medications can make an institution more appropriate than home care.
Finally home care is not always the least expensive choice and also this aspect has to be considered choosing between home care or an institution.

2.2 Home Health Care service

As we said in the previous section, Home Health Care is one of the service that can be done to people with special needs treating them at home. It is important to understand the difference between home health care and home care: the former is in fact more medically oriented. This is why people who provide home health care are often licensed practical nurses, therapists or home health aides. Most of them work for home health agencies or hospitals.
Like general Home Care also Home Health Care service is generally provided to elderly and terminal patients, however it can be applied to other patient categories such as children, post-surgery patients, people hit by cerebral ictus, etc. The service consists in sending nurses from an hospital or a private company to patients' home in order to maintain or improve their life conditions, performing the needed services and avoiding the hospitalization.
The reason of this service from a human point of view is obvious: no one likes to stay in a hospital, although it is sometimes necessary; patients who need to stay in bed but do not require constant medical surveillance prefer their own bed at home. Home health care helps people with special needs live independently for as long as possible, given the limits of their medical conditions.
At the same time, another aspect we have not taken into account yet is that a patient in hospital has a high cost for the community that can be avoided if the patient needs services only a few minutes a day. Home care service is recognized to be one of the major solutions to contain costs in health care.
For these reasons the current trend in Europe and North America is to send nurses to visit patients in their home with the aim of giving better quality of life to patients and reducing costs for the community. Health care service is growing very fast due to the increasing average age of population, the desire to improve the quality of service and the limited economic resources.
For all these reasons it is very important to develop the home health care service in an efficient manner maintaining a high quality of the service. However, delivering this service is not an easy task because of the large number of personnel that participate in the process, of the variety of clinical and organizational decisions, and of the difficulty of synchronizing human and material resources at patients' home. After these observations, it is obvious that planning home health care is a problem that involves Operational Research.

Literature

We now present some studies about home health care from the Operational Research point of view. Optimization problems in health care have received considerable attention during the last few decades. Operations Research is now utilized more frequently to address day to day hospital management, resource constrained operations or treatment planning aspects in health care. Key health care optimization issues include service planning, resource scheduling, logistics, medical therapeutics, disease diagnosis and preventive care.

As seen, different aspects have to be considered in home health care problem. Rais and Viana (2010) made a survey of OR applied to Health Care.

Many studies are focused on optimization and cost control measures, others include estimation of future demand for services in order to build enough capacity or selection of hospital locations for covering a target population.

Demand forecasting is essential and provides the input to many optimization problems, about this argument we can mention for example Finarelli and Johnson (2004), Cote and Tucker (2001), Jones et al.(2008), Beech (2001), Lanzarone et al. (2009).

Other publications take into account the capacity management and the location selection, both for health care services and medical material, within these publications Daskin and Dean (2004), Smith-Daniels et al. (1988) or Rahman and Smith (2000).

Location problems, i.e. where to set healthcare centres for maximising accessibility, has also been extensively studied by many authors as Smith et al. (2009), Murawski and Church (2009), Rahman and Smith (1999) and Hodgson et al. (1998). Of course we can also observe many other aspects and the literature about home health care is huge.

The aspect we take into account in this work is the health care management and logistics, so all the optimization involving patients scheduling and nurses scheduling. Until recently, most personal scheduling problems in hospital were solved manually, of course scheduling by hand used to be a very time consuming task and planners had no automatic tools to test the quality of a constructed schedule. The importance of a systematic approach to create a good timetable is very high especially in health care, where it is unacceptable not to fully support patient care needs and staff requirements. Mathematical or heuristic approaches can easily produce a number of solutions and they can report upon the quality of schedules.

In the following, we present a list of some important works about HHCP. Usually this problem is seen as a combination between the *Period Vehicle Routing Problem with Time Windows* and the *Nurse Rostering Problem*.

In Burke (2002) it has been demonstrated that Variable Neighbourhood Search is a successful method in tackling real world instances of the nurse rostering problem. In Burke et al. (2008) they describe a heuristic ordering approach to the nurse rostering problem which is hybridised with a VNS.

Eveborn et al. (2006) focus on a staff planning for home care in Sweden. They describe and develop a decision support system called Laps Care that takes into account a lot of aspects of HHCP. The problem is formulated using a set par-

titioning model, and it makes use of a repeated matching algorithm, combining optimization methods and heuristics. The proposed visit planning is evaluated according to two performance criteria: the efficiency of the plan and its quality.
Cheng and Rich (1998) consider two types of nurses: part time and full time. The global problem is formulated as a Vehicle Routing Problem with Time Windows. The objective is minimizing the extra working time costs. they propose a mathematical model and a simple heuristic to find a solution firstly building several routes simultaneously and then attempting to make improvements on these tours.
Bäumelt et al. implement a C# algorithm to solve HHCP minimizing the cost of the travelling time. They integrate algorithms for solving *Vehicle Routing Problem* and *Nurse Rostering Porblem* considering that each visit of the patient must occur at a defined time window.
Berteles and Fahle (2006) present another optimization and planning tool. They consider many constraints using thin time windows and with nurses characterized by different skills to limit the number of feasible solutions. The used heuristic builds a set of patients to assign to each nurse and then find an optimal sequencing for each set of patients.
Borsani et al. (2006) propose a health care problem close to the one we will study in the next chapters. They are interested in the problem of deciding which human resource should be used and when to execute the service during the planning horizon in order to satisfy the care plan for each patient. The objective is also the balance of nurses' workload. they propose a simple model of assignment ignoring the routing part of the problem setting the travel and the working time as a deterministic constant.
Begur et al. (1997) study the scheduling and routing of nurses in Alabama and develop a support system decision for HHC providers. It develops a list of patients to visit for each nurse ranked in order to maximize the productivity taking into account route construction, nurse availability and patients' needs and availability.

2.3 The home health care service in Ferrara

The goal of this work is to model and solve the Home Health Care Problem focusing in nurse scheduling and in particular observing the case in the city of Ferrara, that we explain in the following.
Home Care service is generally activated by the doctor assisting a patient and therapies are decided with the nurses, here also denoted as operators. As said before patients can be elderly or terminal ill people, but also children, post-surgery patients etc., i.e. people that do not need to be monitored all the time but just need service for a few minutes a day.
One patient may need more than one service and each service has a given fre-

quency during a week. So every patient has to be treated respecting his care plan, which includes the number, the type and the sequence of visits that he or she should receive.
Services are provided by a set of nurses, the days a nurse is on duty, he/she has a maximum available working time. Every day each nurse starts from the hospital, travels by car from one patient's home to the next and finally returns to the hospital. A treatment lasts from 5 to 60 minutes, depending on its specific characteristics.
As Ferrara is a medium-size town (about 150.000 inhabitants), the considered area is rather large and its population ageing. Although most of the population is concentrated in town, a number of elderly people live in the countryside. For these reasons the service is characterized at the same time by high variance of duration and significant geographical dispersion of the requests.
HC structure provides a weekly visit plan for all its operators. Each nurse has to know the patients to be visited, the type of service to perform and the order and the day of the visit. The assignment patient-operator can be revised periodically, usually weekly, according to the specific needs.
We use a short term planning determining the weekly visit plan for each operator in order to respect the patient's care plan needed on that respective week. Considering a planning horizon of one week, we can divide it into time slots (a day from Monday to Sunday) and for each time slot the nurses on duty for each day is known.
It is crucial to deliver the service in a cost effective manner while not deteriorating service quality, so we have different objectives. The main objectives are: minimize service total costs and maximize service quality. The costs are related with normal and extra hours of the nurses, where extra hours are paid more and should be avoided. We consider a team of 13 nurses. Nurses work 5 days in a week and they have the same fixed working time a day that we consider of 8 hours, each minutes more is considered as extra work.
On the other side, the service quality is measured in terms of loyalty and route balance. The loyalty is related to the relation between patient and nurse, a patient feels more comfortable if the same operator visits him or her. Another aspect that should be considered is the balance of working time between nurses to avoid the burn out phenomena, i.e. nurses who get tired and stressed and act in an unfriendly manner to patients deteriorating the quality of service.
Summarizing, a good solution should achieve the minimization of the travel times over the service times, i.e. the minimization of the time during which a nurse is on duty but is not delivering any service. So the routing problem consists in deciding the best routes for operators considering the real travel distances. In addition, in order to avoid the burn out phenomenon, it is important to have an equi-distribution of the workload. Finally, from the patient point of view, it is

important to have a good degree of loyalty; continuity of care is important, and assigned operator should become a reference figure for the patient during the duration of care plan. For this reason we have to minimize the number of different nurses who visit one patient.

From a mathematical point of view what we have is a set S of services and a planning horizon H of one week divided in days h. For each service the frequency f_s and the duration a_s, respectively the number of times the service must be performed during week and the estimated time to perform it, are given. We have to observe that for each service a minimum number of days e_s must pass between two repetitions of it.
Each service must be performed on a specific patient p_s who is linked with a specific address. What we know is the distance $d_{ss'}$ between to different patients expressed in time. Of course we also know the time needed to go from hospital to a specific patient, and to return from this patient to hospital.
The problem of HHC is focused on planning the services, i.e. distributing the services during the week and for each day deciding the optimal tours between services while not exceeding the service time of a nurse. On the other side, the assignment between services and nurses has to be decided. In other words the problem consists in two big different problems: a routing problem and an assignment problem.
This two problems have to be solved while considering the goals mentioned before:

- minimize the costs, i.e. minimize the working times of nurses avoiding extra work;
- maximize the loyalty between patient and nurse, i.e. minimize the number of different nurses visiting the same patient;
- maximize the balance, i.e. maximize the equidistribution of workload between nurses in order to minimize the phenomenon of burn out.

The case in Ferrara has already been used as real case study for the HHCP. In Cattafi et al. (2012) they apply a Constraint Logic Programming to the HHCP in Ferrara. They model the problem with the objective of reducing disparities in workload of the nurses and improving the loyalty between nurse and patient. A new constraint that address the routing component of the problem by embedding into a constraint an efficient solver for the TSP is also implemented. They try five different search strategies and compare the results with hand-made solutions used by nurses.

In Boccafoli (2012) the problem is modelled considering all the constraints trying to maintain it as close as possible to the reality. He proves that exact method are not an efficient choice to solve the HHCP without simplifying it into other known problems. The problem is then solved using two different methaheuristcs: *Variable Neighbourhood Search* and *Adaptive Large Neighbourhood Search* after using a method to find an initial solution.
Both works consider deterministic travel and service times, according also with medical staff this assumption simplifies too much the problem. In addition they use a single objective function. In this work we try to explain better the problem firstly considering it as a multiobjective problem and then introducing stochastic service times.

2.3.1 Mathematical model

In this section we propose the mathematical model to the HHCP.

Parameters

In the HHC problem the following parameters must be considered:

- H= set of days of the week $h \in H$, from Monday (1) to Sunday (7).
- P= set of patients $p \in P$ that must be visited during the planning week.
- S= set of services $s \in S$.
  Each service is characterized by a vector: $(p_s, f_s, e_s, a_s, g_s)$ in order we have: the patient p_s who needs this service, the frequency f_s (i.e. the number of times the service must be performed during the week), the minimum time e_s that must elapse between two repetition of the service, the time a_s needed to perform the service and finally a vector g_s of 0 and 1 representing the availability of the service during the week. $g_s^h = 1$ if service *s* can be performed on day *h*, 0 otherwise.
- N= set of nurses $n \in N$. Each nurse is characterized by a vector d_n of 0 and 1 that represents the availability of the nurse during the week. $d_n^h = 1$ if nurse *n* is available on day *h*, 0 otherwise.
- T= Set of all possible tours $t \in T$, i.e. sequences of services each one starting and finishing at the hospital. We have T_p set of tours involving at least one service of patient *p*. In order to solve exactly the problem the tours must be generated before, of course, as the number of patients and services increases, the complexity of generating this tours is too high.

- $c_{ss'}$= cost of performing sequentially service *s* and service *s'*. This cost is given by the distance between the two patients who need the two services respectively (in time), $d_{ss'}$, plus the duration of service *s'*, $a_{s'}$. In this thesis when we talk about costs we are talking about working time.
- We also have L= maximum time of workload for each nurse on one day and W_{max}=maximum extra work for each nurse on each day.

We can observe that the cost of a tour *t* can be evaluated by:

$$c_t = \sum_{s,s' \in t} c_{ss'}$$

In addition the number of days a nurse *n* is on duty is:

$$d_n = \sum_{h \in H} d_n^h$$

Variables

Now we introduced the variables of the problem. To solve the problem we have to consider two different decisions:
-SCHEDULING: establish the days on which the service is performed and decide which nurse should perform it.
-ROUTING: we have to make sequences of services in an efficient way per nurse and per day.

The variables we will use are the following:

$$x_{nt}^h = \begin{cases} 1, & \text{if tour } t \text{ is assigned to nurse } n \text{ on day } h \\ 0, & \text{otherwise} \end{cases}$$

$$z_p^n = \begin{cases} 1, & \text{if patient } p \text{ is served by nurse } n \\ 0, & \text{otherwise} \end{cases}$$

$$u \in \mathbb{R}^+$$

$$w \in \mathbb{R}^+$$

The first variable is used to specify where the nurse *n* works during day *h*. As a tour is formed by services, each of them related to a patient, we know where a nurse has to go, which service she has to perform and the needed working time.
The second variable is used to observe more directly the relation between a patient and a nurse.

u and *w* are related with the working time. The first one is chosen, after evaluating the average working time between nurses, as the maximum of them. The second one is the maximum between all the extra working time needed in all tours, if there is extra time.

Formulation of HHCP

The complete formulation can be written as follow:

$$min \quad \alpha_0 u + \alpha_1 \sum_{p \in P} (\sum_{j \in N} z_p^n - 1) + \alpha_2 w \tag{2.1}$$

s.t.

$$u \geq \frac{1}{d_n} \sum_{t \in T} \sum_{h \in H} c_t x_{nt}^h \qquad \forall n \in N \tag{2.2}$$

$$w \geq \sum_{t \in T} c_t x_{nt}^h - L \qquad \forall n \in N, \forall h \in H \tag{2.3}$$

$$w \leq W_{max} \tag{2.4}$$

$$\sum_{h \in H} \sum_{n \in N} x_{nt}^h \leq 1 \qquad \forall t \in T \tag{2.5}$$

$$\sum_{t \in T} x_{nt}^h \leq d_n^h \qquad \forall n \in N, \forall h \in H \tag{2.6}$$

$$\sum_{t \in T} \sum_{n \in N} x_{nt}^h \leq \sum_{n \in N} d_n^h \qquad \forall h \in H \tag{2.7}$$

$$\sum_{h \in H} \sum_{t \in T: s \in t} \sum_{n \in N} x_{nt}^h = f_s \qquad \forall s \in S \tag{2.8}$$

$$\sum_{t \in T: s \in t} x_{nt}^h \leq g_s^h \qquad \forall s \in S, \forall h \in H, \forall n \in N \tag{2.9}$$

$$\sum_{h=h'}^{h+e_s-1} \sum_{t \in T: s \in t} \sum_{n \in N} x_{nt}^h \leq 1 \qquad \forall s \in S, \forall h' = [1..|H| - e_s] \tag{2.10}$$

$$z_p^n \geq x_{nt}^h \qquad \forall p \in P, \forall n \in N, \forall t \in T_p, \forall h \in H \tag{2.11}$$

$$z_p^n \in \{0, 1\} \qquad \forall n \in N, \forall p \in P \tag{2.12}$$

$$x_{nt}^h \in \{0, 1\} \qquad \forall n \in N, \forall t \in T, \forall h \in H \tag{2.13}$$

(2.14)

$$u \in \mathbb{R}^+ \tag{2.15}$$

$$w \in \mathbb{R}^+ \tag{2.16}$$

Let us explain the meaning of each constraint in the formulation.
(1) is the objective function and can be changed assigning different values to α_i. The first and the third parts are for routing, they minimize respectively the workload and the overtime. The second part is for the loyalty and minimizes the different number of nurses visiting the same patient.
Observe that in the objective function we are not considering the working balance: the third important main goal. This is because in the first part of this work the balance will not be considered to allow us to compare the first results with others similar works. The balance will be taken into account after the third chapter.

Constraint (2) defines *u* as the maximum average of workload between all nurses and constraint (3) and (4) define *w* as the maximum extrawork found over all nurses and all days that must be less than W_{max}.
Constraints (5) to (7) are for the scheduling part. The first constraint tells us that each tour can be assigned to one nurse maximum. The second one says that on each day *h* and for each nurse *n* we can assign only one tour to the nurse *n* and only if this nurse is available on the considered day. And the last one is about assigning tours to days: we can assign a number of tours to day *h* that must be less or equal to the number of nurse in charge on the considered day.
Constraints (8) to (10) are for the routing part of the problem. With (8) we guarantee that the frequency f_s of a service s is respected. To make this constraint working with all type of services we have to consider services that have frequency greater than one but can be done in the same day as a unique service with needed time: $a_s * f_s$. Constraint (9) is to guarantee that a service *s* in performed on day *h* only if it is available.
(10) imposes an interval at least of e_s between two repetition of one service.
(11) express the relation between variables, if patient *p* is not assigned to nurse *n*, then we do not assign a tour *t* which involves patient *p* to the nurse *n*.

After presenting the mathematical model of the HHCP we can say that the problem is NP-hard since it is an extension of routing problems.
As the planning process in HHC requires to satisfy a large number of constraints and objectives, regarding both the efficiency of the system and the quality of the care, the required computational time to solve the problem with an exact method can be considerably high. In addition we have to consider that the master plan is usually heavily modified from day to day due to changes in the service and in the patients requirements, thus the effort of finding optimal solution can be useless.
As the real problem is NP-hard in literature is usually reduced to known problems or variants of them. In Boccafoli (2012) different approaches to find optimal solutions through exact methods are considered, as the work shows, all of these approaches appear unsatisfactory to solve HHCP.
We can conclude that the best way to approach to this problem is using meta-

heuristics approaches. In the next chapter we propose a metaheuristic based on ILS method to solve the HHCP trying to maintain it closed to the real problem.

Chapter 3

Solving HHCP using Iterated Local Search

In this chapter we introduce the Iterated Local Search method, explaining its construction and mentioning some applications. The algorithm is then applied to Home Health Care optimization problem considering as objective function the sum of costs, expressed in time, and the loyalty of the problem, as a summation of different nurses visiting the same patient. All sub parts composing the ILS algorithm for HHCP are explained in details.

3.1 Iterated Local Search

We propose a metaheuristic based on Iterated Local Search to solve the HHCP. The Iterated Local Search is a simply but efficient method to solve difficult and complex optimization problems, like the one treated in this thesis.
The basic idea behind ILS method is that a metaheuristic method is preferable to be simple, both conceptually and in practice, effective and general purpose. Indeed, to have greater performance, metaheuritics have become more and more sophisticated and request problem-specific knowledge, often loosing generality and simplicity. To counter this, the idea is to split the metaheuristic algorithm into few parts, trying to have a general purpose part separated from the problem-specific knowledge in the algorithm.
The essence of Iterated Local Search metaheuristic is to build iteratively a sequence of local optimal solutions. At each iteration a perturbation phase is then applied to restart the process.

3.1.1 Iterating a local search

Given a heuristic optimization algorithm, in our case *LocalSearch*, it can be shown that using iteration this algorithm can be significantly improved.
The idea on which the Iterated Local Search is developed is explain in the following lines.
First how *LocalSearch* works is explained: a cost function *C*, that must be minimized, and a set of candidate solutions *S* are given. *LocalSearch* defines a way of mapping the elements in the set *S* to the elements in the smaller set S^*, set of locally optimum solutions.
Usually comparing the distributions of costs density of elements in the two sets it can be observed that the mean and the variance for solutions in S^* are smaller than the ones for solutions in *S*. This means that it is better to use local search than sample randomly in *S*.
Usually Local Search methods can get stuck in a local minimum, where no improving neighbours are available. A simple modification consists in iterating calls to the local search routine, each time starting from a different initial configuration. The difference with restart approaches is that ILS, instead of simply repeating Local Search starting from an initial solution, optimizes solution *s* with *LocalSearch*, perturbates the local optimal solution and applies *LocalSearch* again. This procedure is repeated iteratively, until a termination condition is met.
Initial solutions should employ as much information as possible to be a fairly good starting point for local search. Most local search operators are deterministic while the perturbation mechanism should introduce non deterministic component to explore the solutions space. The perturbation must be strong enough to allow the escape from basins of attraction, but low enough to exploit knowledge from previous iterations.

The ILS algorithm works as follow:

Algorithm 1: *Iterated Local Search*

Result: Iterated Local Search
s_0 = GenerateInitialSolution;
$s^* = LocalSearch(s_0)$;
repeat
  $s' = Perturbation(s*, \textit{history})$;
  $s^{*'} = LocalSearch(s')$;
  $s^* = AcceptanceCriterion(s^*, s^{*'}, \textit{history})$;
until *termination condition met*;

After finding an initial solution s_0, a local search is applied on it, obtaining a new solution s^*. Then starting from a current solution s^*, a perturbation is applied reaching an intermediate state $s' \in S$ and then *LocalSearch* is applied again on it obtaining $s^{*'}$; if element $s^{*'}$ passes an acceptance test it becomes the new element of the walk, otherwise we come back to s^*. Then we iterate the procedure obtaining finally a walk on S^* that is the resulting of a stochastic walking on S^*. The procedure is shown in the figure below taken from Lourenço, Martin and Stüzle (2010).

Most of the ILS complexity is due to the historical dependence, when perturbations or acceptance criterion depend on any of the previous s^*.
Historical independent perturbations and acceptance criterion can be chosen, obtaining a random walk dynamics on S^*, that is Markovian. In this case the acceptance criterion is the difference of costs between s^* and $s^{*'}$, or it can be also choose to accept only improving solutions.
Also a stopping criterion, based on maximum time, can be added, so only two nested local search are obtained.
The efficiency of the method depends both on the kinds of perturbations and on the acceptance criteria.

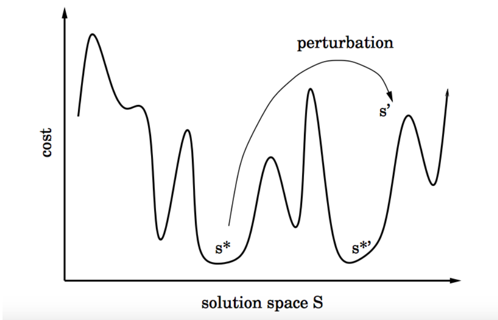

Figure 3.1: Rapresentation of Iterated Local Search procedure

3.1.2 Procedure in details

To design an ILS to solve a specific problem four components must be defined:

- *GenerateInitialSolution*
- *LocalSearch*
- *Perturbation*
- *AcceptanceCriterion*

Starting with a basic version of ILS usually still higher performance are obtained than the random restart approach. Then each single part of the ILS method can be optimized and finally the interactions between the four components must be considered to globally optimize the method.

Initial Solution

To have a good initial solution s_0 is important in order to reach faster an high-quality solution, because of the historical dependence of the walk on S^*.
In the basic version of ILS the starting point can be a random $s \in S$, or, better than this, a solution given by a greedy algorithm.
For short to medium sizes runs the initial solution is important to reach the highest quality solutions possible. For much longer runs the initial solution seems to be less relevant.

Local Search

ILS algorithm is quite sensitive to the choice of the embedded heuristic, so, when it is possible, it is better to optimize it. Often it is true that better the local search, better the corresponding ILS, but this is not always true. Sometimes it is better to choose worst but faster local search, especially if the total computational time is fixed.
Another important thing is that the local search should not undo the perturbation. We have also to consider that, sometimes, it can be useful to replace Local Search with Tabu Search or Simulated Annealing limiting their run time.
As said before a Local Search defines a many to one mapping from the set S, i.e. the set of all candidate solutions, to the smaller set S^* of locally minimum solutions. Let now C be the cost function of the considered combinatorial optimization problem that is to be minimized, comparing the distributions of costs density of elements in the two sets, S and S^*, it can be observed that the mean and the variance for solutions in S^* are smaller than the ones for solutions in S.

Perturbation

Perturbation is used by ILS to escape from local optima. In the basic version of ILS, a perturbation can be a random move in a higher order neighbourhood than the one used by local search algorithm, often this leads to a satisfactory algorithm. We have to notice that perturbations must not be too small or too big, if they are too small $s^{*'}$ is often rejected obtaining a short walk, if they are too big a random walk is obtained, which has worst results. In addition, to avoid short cycles, perturbations can being randomized instead of be deterministic.
We have to choose how to perturb a solution, so the *strength* of the perturbation, i.e. we have to choose how many solution components are modified. The perturbation strength effects the speed of the local search.
For some problems we can hope to have satisfactory ILS when using perturbations of fixed size, but for more difficult problems this may lead to poor performance.

In addition for some problems appropriate perturbation strength is very small and independent of the instance size, for other problems large perturbation sizes are better.
Perturbation strength can also be modified during the run, to do that the search history can be exploited. The perturbation can be adapted changing its strength deterministically or a Tabu Search can be used to implement a perturbation scheme. Finally more complex perturbation schemes can also be developed: this can be done modifying the definition of the instance and then run *LocalSearch* on s^*, obtaining s' as output. Another way is to generate perturbations optimizing a subpart of the problem.
It is important to observe that another reason why this metaheuristic can perform better than random restart is that *LocalSearch* works faster on solution obtained by perturbing the local optimum than another random solution. So in a given amount of time, ILS can explore many more local optima then the random restart.

Acceptance Criterion

When we find a new $s^{*'}$, using *perturbation* and *LocalSearch*, then we have to decide if accepting it or not.
The acceptance criterion can be used to control the balance between intensification and diversification of the search. To have a strong intensification we can consider *Better* acceptance criterion, that consists in accepting only better solutions. Contrarily if we want to have a strong diversification a random walk acceptance criterion can be used, which accepts all the new solutions found. Of course an intermediation between these two acceptance criterion can be used. A Simulated Annealing type acceptance criterion can be used, accepting $s^{*'}$ if it is better then $s*$, otherwise if it is worse we accept it with a probability $exp\{(C(s*) - C(s^{*'}))/T\}$ where T is the temperature. With high temperature we favour diversification over intensification, the contrary if the temperature is low. To have better results we can prefer intensification until is useful and then prefer diversification for a while, before resume intensification.

Global optimization

Optimize the single components of ILS it is not enough to have a good search, we have also to globally improve it, considering the relations between components. If for example we consider that the initial solution is irrelevant for a problem, as sometimes happens, we can notice that *Perturbation* depends on the choice of *LocalSearch* and *AcceptanceCriterion* depends on both the choices of *Perturbation* and *LocalSearch*.
Improving the single components we have to remember that they are related each

other, especially the perturbation should not be easily undo by the local search and that the balance between intensification and diversification is determined both by *Perturbation* and *AcceptanceCriterion*.

3.1.3 Applications of ILS method

ILS algorithm has been successfully applied to different combinatorial optimization problems, sometimes getting high performances while, in other cases, it is merely competitive with other metaheuristics. A review of ILS applications can be found in Lourenço, Martin and Stüzle (2003) and (2010).
One of the first applications we have to quote is ILS for *Travelling Salesman Problem*. A basic ILS algorithm on TSP consists in: generating an initial solution with a greedy heuristic, using a *LocalSearch* such as: $2_{opt}, 3_{opt}$ or Lin-Kernighan to find s^*, perturbing it with a double-bridge move and finally accepting $s^{*'}$ only if $f(s^{*'}) \leq f(s*)$.
Quoting to some authors who used ILS algorithm to solve a TSP follow. Baum (1986) can be considered the first ILS algorithm for this problem. He tests several variants of first-improvement type algorithm for the local search and uses a random 2-exchange step to perturb the solution. Only better quality tours are accepted. He gets poor performance probably because of poor choice for perturbation.
Martin, Otto and Felten (1991-1996) develops the first high performing algorithm for the TSP called: *Large-Step Markov Chains*. For the local search they consider both the 3-opt and the Lin-Kernighan heuristic. The perturbation is done using a double-bridge move and finally new solution is accepted using a simulated annealing criteria.
Iterated Lin-Kernighan is developed in Johnson (1990-1997) and differs from the LSMC principally for the using of random double-bridge moving for the perturbation.
An high performing ILS algorithm can be found as part of Concorde software package, developed by Applegate et al. Applegate makes tests on very large TSP instances up to 25 millions cities.
Another application of ILS is for *Vehicle Routing Problem*. Lau et al. (2001) uses ILS for *Vehicle Routing Problem with Time Windows*, Osman (1993) for capacitated VRP, Ribeiro and Lourenço (2005) for multi-period inventory routing.
Big applications can be found for several scheduling problems. For single machine total weighted tardiness problem, iterated dynasearch or ILS with VND local search are used, Congram, Potts, Van del Velde (1998), den Besten et al. (2000). ILS is used also for single and parallel machine scheduling, Brucker, Hurink, Werner (1996-1997), and flow shop scheduling, Stüzle (1998), Yank et al. (2000). For job shop scheduling application we can quote Lourenço (1995)

for ILS with optimization of subproblems and Kreiple (2000) for total wighted tardiness job shop problem.

3.2 ILS for HHCP

In this thesis an ILS is proposed to solve the HHCP problem. In the following sections we describe in detail all the components of the developed ILS method.

3.2.1 Initial Solution

Before starting with the initial solution algorithm we have to introduce some elements that must be generated.

- CostMatrix : matrix of costs between two different services s and s'.

$$c_{s,s'} = d_{p_s,p_{s'}} + a_{s'} \tag{3.1}$$

  i.e. the cost to go from service s to service s' is equal to the distance $d_{ss'}$ between the patients that need respectively service s and s' plus the time needed to perform service s', $a_{s'}$. Note than both distances and costs are expressed in minutes. In this matrix is included the time to go to a patient from the hospital and to come back to it considering $a_0 = 0$. We observe that in this thesis when we talk about costs we are expressing them in time.

- PriorityMatrix : matrix of N_s rows and 7 columns. For each service a vector of seven elements is created. $priority_s^h$ is equal to:

  - 0 if the service s is not available on day h;
  - 1 if service s is available on day h but it is not urgent, i.e. h is not the last day during which service s can be performed;
  - 2 if service s has $f_s = 1$ and h is the last day during which it can be performed or if service s has $f_s > 1$ and h is the last day during which it can be performed considering that it must be repeated f_s times with e_s days between each repetition.

  Lets take as example two different services:

  s1: $f_s = 1 \quad e_s = -1 \quad g_s = [0, 0, 1, 1, 0, 1, 0]$
  s2: $f_s = 2 \quad e_s = 1 \quad g_s = [0, 1, 0, 1, 1, 1, 0]$

  The priority is respectively:

0	1	0	1	1	2	0

0	1	0	2	1	2	0

Figure 3.2: Priority vectors

Clarke and Wright

The initial solution method is based on the Clarke and Wright algorithm, therefore we briefly introduce this well known algorithm.
This procedure was designed for the Vehicle Routing Problem. In this problem, a depot and a number of customers are given, and we have to deliver goods from the first to the seconds.
We have only a given number of vehicles available, each with a certain capacity. Each vehicle must cover a route and each route must start and end at the depot. The problem consists in deciding how to allocate the customers among routes and to which vehicle assign a route, minimizing the total cost and respecting the constraints (number of available vehicles and capacity).
To solve the VRP, in Clarke and Wright (1964) a savings algorithm was proposed. This algorithm is an heuristic algorithm, for this reason it does not provide an optimal solution with certainty, but usually a very good one.

The algorithm is based on the concept of savings. Given two routes we can evaluate how much we can save using *merge* on these two, i.e. joining them.
The algorithm starts with a number of routes equal to the number of costumers, each route consists in starting in the depot, visiting one customer and coming back.
Given the cost c_{ij} between two customers the savings obtained combining two routes, one visiting as last costumer *i* and the other visiting first costumer *j*, are evaluated as follows:

$$S_{ij} = c_{i0} + c_{0j} - cij$$

It is obvious that larger is the value of s_{ij} better is to *merge* the two routes.
So, at each iteration, savings are evaluated, sorted and then *merge* operator is used on the two tours with larger saving, the algorithm is repeated again until all the savings found are negative, i.e. it is not convenient to *merge* tours.
The algorithm is presented below.

The savings algorithm is useful in a section of the procedure used to find a feasible Initial Solution for the HHCP.
The algorithm we used is a modified version of the original one. In fact we add the condition that two paths can be *merged* only if the total cost of the new paths

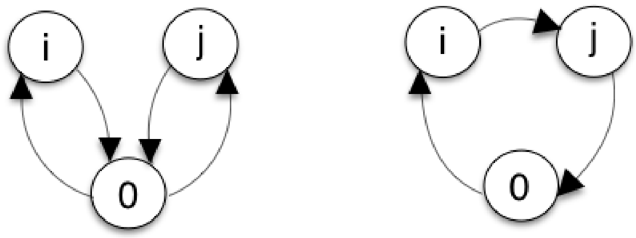

Figure 3.3: Illustration of the savings concept

Algorithm 2: *Clarke and Wright*

---

input : Set of Services, Cost Matrix
output: Set of paths with optimized costs

```
1  Initialize the set of paths with [0, s, 0] for each service in the given set;
2  repeat
3      Compute saving cost between all customers;
4      Sort the savings;
5      Pick savings in order and merge the paths of the corresponding
       customers;
6  until There is no possible improvement;
```

---

is less than the maximum working time L.
The modified procedure is shown below.

Algorithm 3: *Modified Clarke and Wright*

---

input : Set of Services, Cost Matrix, Maximum working time L
output: Set of paths with duration lower than L

```
1  Initialize the set of paths with [0, s, 0] for each service in the given set;
2  repeat
3      Compute saving cost between all services;
4      Sort the savings;
5      Pick savings in order and observe the corresponding paths;
6      if Resulting path has duration ≤ L then
7          merge the paths;
8          go to 3;
9      else
10         go to 5;
11 until There is no possible improvement;
```

Algorithm: Initial Solution

We can now start searching a feasible initial solution. The algorithm for the initial solution is presented below.

Algorithm 4: *Generate Initial Solution*

input : Set of Patients, Set of Nurses, Nurses' availability, Set of services, Matrix of distances between patients

output: Schedule of daily visits for each nurse

```
Initialize Cost Matrix and Priority Matrix;
for h ← 1 in 7:
    N_h = number of available nurses on day h;
    Let U be the subset of S of services with priority 2 on day h;
    if U is empty then
        go on;
    else
        if h=1 then
            go to 12;
        else
            PutServicesOnPrevioseDays(U,S,h,T,C);
    if U is not empty then
        tours[h]= ClarkeAndWright(U,L);
        M_h = number of tours in h;
        if M_h is more than N_h then
            Split the shortest paths on the others even if the cost is more
            than L;
    update S;
AssignToursToNurse(N,S,T)
```

Algorithm 5: *Put services in previous days*

input : set U, set S, day h, set of Tours, Matrix of Costs
output: Updated set U, Updated set S, Updated tours

foreach *service* $s \in U$ **do**
  for $h' \leftarrow 1$ **in** $h - 1$**:**
    if $g_s^d \neq 0$ **then**
      foreach *tour* $t \in T$ **do**
        $cost = \min_j(TotalCost(t, h') + c_{j,s} + c_{s,j+1} - c_{j,j+1})$;
        if $cost \leq L$ ***and*** *∃s' ∈ tour[h',t] : p(s') = p(s)* **then**
          add *s* in position *j'+1* of tour *t'* of day *h'*;
          $U \setminus \{s\}$;
          go on with services;
      $\{t', j'\} = \text{argmin}\, TotalCost(t, h') + c_{j,s} + c_{s,j+1} - c_{j,j+1}$;
      $cost = \min_{t,j}(TotalCost(t, h') + c_{j,s} + c_{s,j+1} - c_{j,j+1})$;
      if $cost \leq L$ **then**
        add *s* in position *j'+1* of tour *t'* of day *h'*;
        $U \setminus \{s\}$;
        go on with services;
      else if $M_{h'} < N_{h'}$ **then**
        $tours[d, t] \cup [0, s, 0]$;
        $U \setminus \{s\}$;
        go on with services;
update S;

Algorithm 6: *Assign Tours to Nurses*

input : Set of Nurses, Nurses' availability, Set of services, Set of tours for each day
output: Schedule of daily visits for each nurse

```
for h ← 1 in 7:
    N_h = Set of nurses available on day h;
    T_h = Set of tours on day h;
    if h == 1then
        Assign randomly tours to nurses in N;
    else
        foreach Nurse n in N_h do
            foreach Tour t in T_h do
                Evaluate how many patients the nurse n has in common
                with tour t;
        Assign the tours to the available nurse maximizing the number of
        patients in common;
```

The algorithm works in a simple and fast way to find a feasible initial solution. It starts evaluating the cost matrix and the priority matrix as explained before.
For each day *h* a subset of services *U* with priority 2 on that day is found. If the set is empty we go on to the next day, otherwise, if the day is different from 1 (i.e. we are not on Monday) we try to put the services in *U* in the previous days.
This part evaluates, for each service in *U*, if we have a previous available day h' in which we can perform this service; if we find a day we try to put the service in one of the paths of this day, or in a new one if the number of paths in h' is less than the number of available nurse $N_{h'}$. If in the first available day, in which the service s can be performed, there is a tour in which another service of the same patient is performed and the cost of the tour, after putting the service s in the position where it costs less, is less than the maximum cost L, the service is put in this tour. Otherwise the service s is putted in the tour and in the position where it costs less to us. We have to observe that to make the workload above the maximum cost L we put the service only if the cost of the new path is lower than the maximum cost to avoid extra work. If no tours are found where to put service s but the number of tours $M_{h'}$ in day h' is less than the number of available nurse $N_{h'}$ then a new tour $[0, s, 0]$ is added. If it is not possible to put the service s in day h' the following available day is checked.
After this operation (or if we are on Monday), if the set *U* is not empty it means that the algorithm has not been able to put all the services in the set on previous days because the nurses are all already full. To optimize the distribution of these

services in tours, we can use a modified Clarke and Wright algorithm. The algorithm is shown separately from the previous one. The decision of applying the modified Clarke and Wright algorithm only on a small subset of services is due to minimize the needed computational times.
At the end of the Clarke and Wright algorithm we obtain a set of optimized paths with costs lower than L. A problem that we can have is to find a number of paths that is greater than the number of available nurses on this days, this can happen especially on Saturday or Sunday when the number of nurses is low. In this case we have to split some paths on the others, to make it faster and to have lower costs we split the shortest paths. In this case paths with costs bigger than L can be found.
Then we go on with the procedure.

To have a feasible solution it is important to remember that each time an element s that has $f_s > 1$ is introduced some procedures are performed:

- the available day vector of service s is updated observing also e_s
- the frequency f_s is reduced of one unit
- service s is put again in set S of services that have still to be performed

After running this algorithm the problem is still not completely solved. We have also to specify on which day a nurse is working and which tour has to do.
To decide how to assign tours to nurses we try to maximize the loyalty. Of course results will be quite bad because we do not care so much about the loyalty in the algorithm used to find the initial solution. The loyalty is improved after with the metaheuristics methods.
Times needed to find the initial solutions are very low: less than 0:10 seconds for each instance.

3.2.2 Local Search

Several methods can be used to improve the initial solution, we start using a simple local search. We present the neighbourhood and the algorithm used to improve the solution.

Operators

All the used operators can be summarized in one: the operator *move*. This operator consists in taking a service from a given tour on a given day and put it in another tour in the position where it costs less.

The difference between operators is the criterion used to select from which nurse, from which day or from which tour we are going to take the service to be moved, or which service we choose, and to which nurse and which tour we move it.
All operators can move a service only if constraints about available days and days to wait between two repetitions of the same service are satisfied.

M1: from an overloaded nurse to an under loaded one

This operator is chosen in order to minimize the *u* (i.e. the maximum average of working time between all nurses).
We select the nurse with higher average of working time and we move one service from this nurse to another one with a lower average of working time.
Calling u_n this quantity we can evaluate it as:

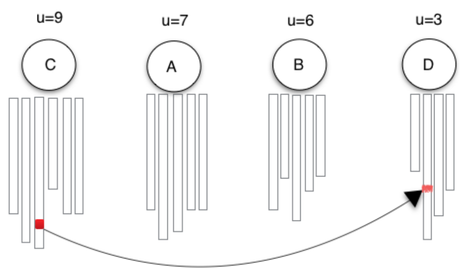

Figure 3.4: Illustration of operator M1

$$u_n = \frac{1}{d_n} \sum_{h:n \in N_h} u_n^h$$

Where d_n is the number of days on which the nurse *n* is in charge, N_h is the set of nurses available during day *h* and u_n^h is the total working time of nurse *n* during day *h*.

The service to move and the nurse and tour where to move it are chosen in order to minimize the costs.
In addition a service is moved only if the total cost of the new tour does not exceed the maximum working time L. This operator is used as function of the local search to improve the *u* as follows.

Algorithm 7: *ImproveU*

input : Schedule of daily visits for each nurse
output: Improved schedule of daily visits for each nurse

Evaluate u_n *and order the nurses with increasing* u_n *in the sequence* N;
$k = 0$;
n_{max}=$N[len(N)]$ /* nurse with maximum u_n */;
while $k < len(N)$ **do**
    n_{min}=$N[k]$ /* nurse with a lower u_n */;
    Try to use M1 between n_{max} and n_{min};
    if $u_{new} \leq u_{old}$ *or* $f_{new} \leq f_{old}$ **then**
        return 1;
    else
        $k++$
return 0;

M2: from a nurse with less services of patients p to another one with a lot of services of him

This *move* is chosen in order to maximize the loyalty, i.e. to have for each patient the lowest number of different nurses visiting him.
In this case the service to be moved is chosen in a different way. We evaluate for a patient *p* how many nurses visit him during the week and how many services each nurse does to this patient. We order the nurses considering the number of services each nurse does to patient *p* and we try to move all the services of the nurse with lower number of services, of the considered patient, to the ones with higher.

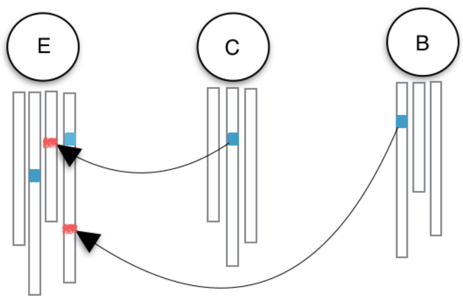

Figure 3.5: Illustration of operator M2

In this case the services to be moved are fixed but we still have to choose to which nurse and to which tour move them.
In the same way as before we start with the nurse with highest number of services and we evaluate for each tour and position which is the best place where to put the observed service and we move it only if the cost of the resulting tour is less than the maximum working time L.

If no place is found we try to move the service to the following nurse in the order and so on.
This operator is used as function in the Local Search in order to improve the loyalty.

Algorithm 8: *ImproveL*

input : Schedule of daily visits for each nurse
output: Improved schedule of daily visits for each nurse

```
P=set of patients with loyalty greater than 1;
foreach p ∈ P do
    N =set of nurses performing services on patient p;
    if | N |> 1 then
        Order nurses in N considering the increasing number of services
        performed to p;
        foreach n ∈ N do
            Try to use M2 between nurses with lower number of services of
            p and nurses with higher;
```

M3: from an overloaded nurse to an under loaded one serving the same patient

This *move* is chosen in order to maximize the loyalty, i.e. to have for each patient the lowest number of different nurses visiting him, without penalize the *u*.
We evaluate for a patient *p* how many nurses visit him during the week and which is the average of working time of each of these nurses u_n.
We order the nurses considering their u_n and we try to move all the services from the nurse with higher u_n to one with lower value of it.

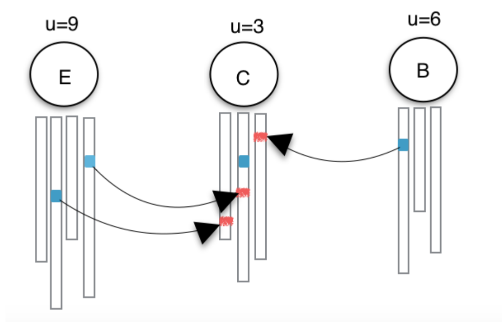

Figure 3.6: Illustration of operator M3

In this case the services to be moved are fixed but we still have to choose to which nurse and to which tour move them.
In the same way as before we start with the nurse with lowest value of average working time and we evaluate for each tour and position where is the best place to put the observed service and we move it only if the cost of the resulting tour is less than L.

If no place is found we try to move the service to the following nurse in the order and so on.
This operator is used as function in the Local Search in order to improve the loyalty without penalize too much the *u*.

Algorithm 9: *ImproveLU*

input : Schedule of daily visits for each nurse
output: Improved schedule of daily visits for each nurse

```
P=set of patients with loyalty greater than 1;
foreach p ∈ P do
    N =set of nurses performing services on patient p;
    if | N |> 1 then
        Order nurses in N considering the decreasing u_n;
        foreach n ∈ N do
            Try to use M3 between nurses with lower value of u_n and nurses
            with higher;
```

M4: from the longest tour to another one

We have to remember that sometimes the initial solution generates an extra work *w* that must be avoided to not have high costs. This extra work time is generated when the Clarke and Wright algorithm generates a number of tours that is bigger than the number of available nurses. In this case the shortest tours are split among the others tours without considering the exceeding of the maximum length for a tour.

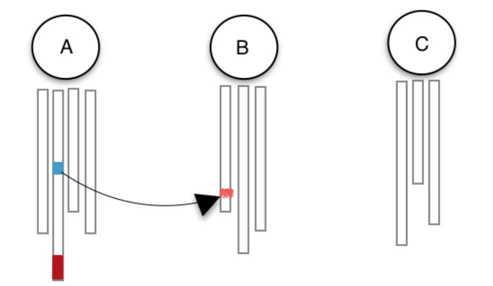

Figure 3.7: Illustration of operator M4

With the operator M4 we try to reduce this extra work time, because sometimes the operator M1 is not sufficient. This happens when the nurse with the maximum extra work time is not the same as the nurse with the maximum u_n, i.e. the maximum average of working time during the week.
This operator works in a very easy way taking a service from the tour that exceed more the maximum working time and putting it in another tour. Obviously both the service and the position where to put it are chosen with the goal of minimizing the cost and not exceeding the maximum working time L.

Algorithm 10: *ImproveW*

input : Schedule of daily visits for each nurse
output: Improved schedule of daily visits for each nurse

$\bar{t}$ =*tour with highest extra working time*;
use M4 on $\bar{t}$;

Algorithm: Local Search

Now we can explain the algorithm used to improve the initial solution.
The algorithm is chosen in order to be fast and to find a solution that is a compromise between the different objective functions we have to consider.
Of course we have to notice that each time we move a service we consider the feasibility of this moving, i.e. the service must be available in the day where we are moving it and if the service has frequency greater than one we have to update its availability as in the Initial Solution Algorithm. So each time an element *s* that has $f_s > 1$ is moved some procedures are performed:

- the available day vector of service s is updated considering also e_s
- the frequency f_s is reduced of one unit
- service s is put again in set S of services that have still to be performed

Algorithm 11: *Local Search*

input : Schedule of daily visits for each nurse
output: Improved schedule of daily visits for each nurse

$iter = 0$;
repeat
    if $w > 0$ **then**
        ImproveW()
    $iter_1 = 0$;
    repeat
        ImproveU();
        $iter_1 = iter_1 + 1$;
    until *There is no improvement or* $iter_1 = 50$;
    ImproveL();
    ImproveLU();
    $iter = iter + 1$;
until *Number of consecutive equal solutions* $= 2$ *or* $iter = 15$;

The algorithm works starting from the initial solution and trying to improve it. At the beginning it checks if the extra work time *w* is bigger than 0, if yes function *ImproveW* is used. Then it tries to make the *u* lower with a maximum of 50 *move*, i.e. it uses function *ImproveU* until there are not more improvement or the maximum number of iteration is achieved.
Then it uses functions *ImproveL* and *ImproveLU*.
The algorithm stops when for two consecutive iterations we obtain the same results or if the number of iterations is bigger than 15.

After optimizing the *u* and after optimizing the *l* the solution is saved in a vector. This vector contains all the couple of solutions $(u + 10w, l)$. Between all this solutions we choose the one with the minimum *f*, i.e. the one with the minimum: $u + 10w + l$. On this solution the perturbation will be applied. The chosen objective function considers the three variable *u,w* and *l* giving different values to the weights α_0, α_1 and α_2. α_0 and α_2 are taken equal to one while the extra hours are multiply by 10 because they cost more.

3.2.3 Perturbation

In this section we explain how the perturbation is performed on the solution found by the Local Search. This is based on a random destroy-construct approach using a Geometric distribution function.
After several tests we decide to make the perturbation removing all the services of a set of patients and inserting them again in the order they have been removed. Of course we have to choose a criterion for selecting patients and for deciding where to put the services again in the tours.
After running the *LocalSearch* and selecting the best solution found, i.e. the one with minimum *f*, we have for each nurse on which day and in which tour a patient has to be visit. Observing that the loyalty is the most critical part of the objective function the idea is to sort all patients on increasing loyalty and then took the 20% of them, the index of the removed patients are chosen according to a Geometric distribution. In this way patients with lowest loyalty have an higher probability to be taken. Then all the services of these patients are removed from the tours and reinserted in the places where they cost less.
Of course, removing and reinserting services, we have to consider the availability of the services, the frequency and the waiting time between two repetitions.
The algorithm for the perturbation is presented next.

Algorithm 12: *Perturbation*

input : Best solution of Local Search
output: Perturbed Solution

P = list of patients ordered by increasing loyalty;
r_p = *Geometric*$(p, 0.2 * n_p)$, index of patients to be removed;
S_{r_p} = set of patients' services to be removed;
foreach *service* $s \in S_{r_p}$ **do**
    Put *s* where it cost less

3.2.4 Algorithm: ILS for HHCP

Connecting all the sub parts explained in the previous sections we can now formulate the final algorithm to apply ILS method to HHCP.
The selected stopping criterion is a maximum running time that is set at *TimeLimit* $= 1500s$.
For the Acceptance Criterion all the solutions found after the perturbations are always accepted so we do not put the part for the acceptance criterion in our algorithm. The reason under this choice is to have a strong diversification that can be useful in next chapters of this thesis, when the pool of approximate Pareto optimal solutions is introduced.

Algorithm 13: *Iterated Local Search*

s_0 = GenerateInitialSolution;
$s* = LocalSearch(s_0)$;
repeat
    $s' = Perturbation(s*, \textit{history})$;
    $s*' = LocalSearch(s')$;
until *time*$\geq$*TimeLimit*;

3.2.5 Application on real case: ASL of Ferrara

In this section we present the instances on which the previous algorithm is applied and the obtained results for each instance. First results obtained after using the *InitialSolution* algorithm are presented, then the ones obtained after using the ILS method. We can observe the improvement of results before and after using the metaheuristic.
We also compare results obtained to previous results found in other works on

HHCP with a different metaheuristic but same instances and time limit.

3.2.5.1 Description of Data

Real data are given by ASL of Ferrara and they represent the historical of requests R during February 2010, i.e. a period of four weeks.
One request $r \in R$ is represented by $(h, type_s, p_s)$ representing respectively the day on which the request is performed, the type of the service and the patient who needs it.
A service s is a couple of elements $(type_s, p_s)$ respectively type of service and patient who needs it. The set of service is $S = \{s = (type_s, p_s) \mid \exists h \in H : (h, type_s, p_s) \in R\}$.

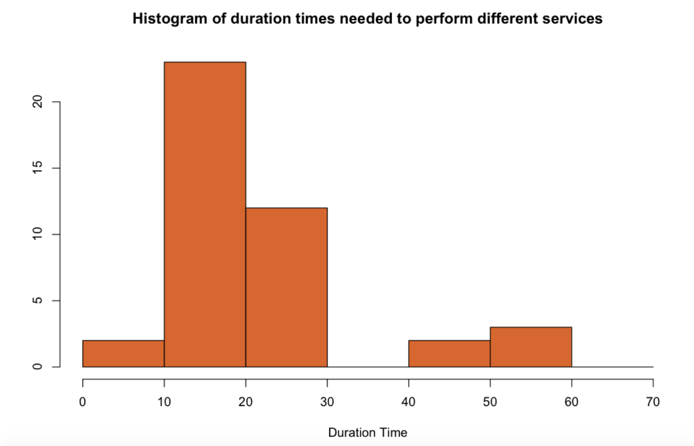

Figure 3.8: Number of services that need a certain ammount of time

We have a list of type of services with their duration, 42 different types of service can be performed. We can observe in figure 3.8 for each duration time how many types of service take that time.
In addition we have on which day a service is performed. We can observe in figure 3.9 for each day of the week how many services are performed and make an histogram on the number of services for each day on each week.
The number of nurses is 13 and we have the availability of them over all the week. Observing it, the number of available nurses from Monday to Friday is 12, it becomes 4 on Saturday and 1 on Sunday.

From these real data 60 instances were generated in Boccafoli (2012). These instances are generated starting from the historical data. On this instances the fre-

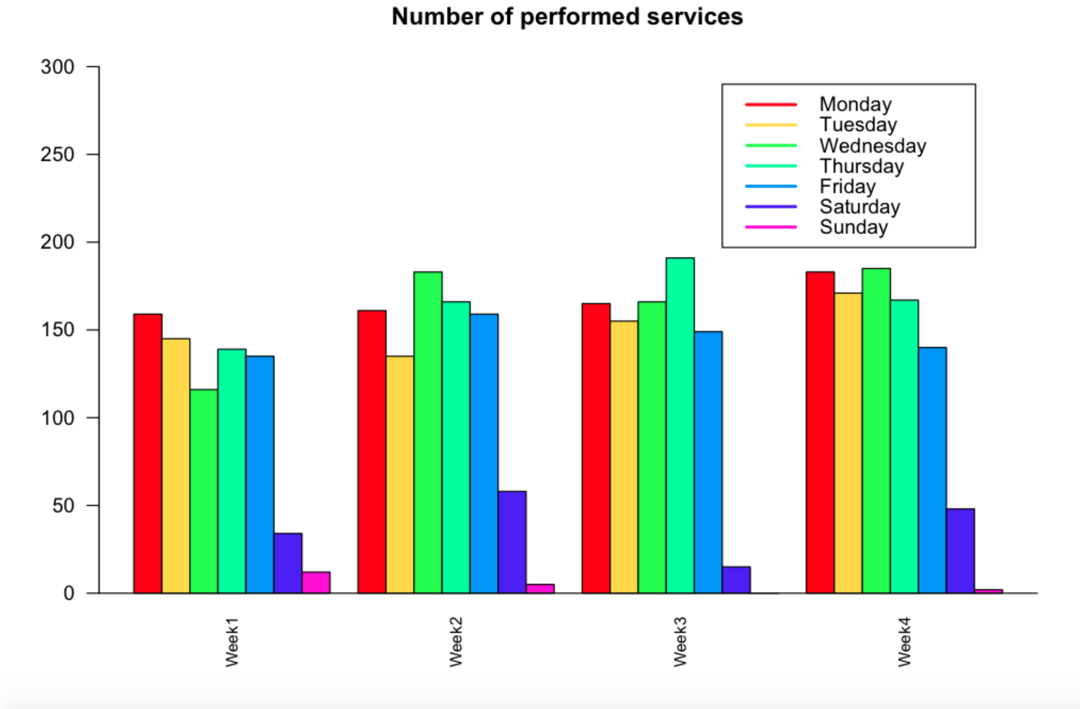

Figure 3.9: Histograms of the number of requests for each day of the week from real data

quency f_s and a needed waiting time between two repetition of a service e_s are added to each request. e_s is putted at -1 when the frequency is 1.

The final instances we have to work with are summarized as follows:

- NurseDay : d_n^h is equal to 1 if nurse *n* can work during day h, it is 0 otherwise.
- Patient : list of patients (each patient correspond to a number) that need to be served during the considered week;
- Distances : matrix of distances between all the patients and between patients and hospital. The distances are expressed in time;
- TypeOfService : list of all types of services. For each service we have: code, full name, average of duration time of the service;
- Test : list of all the services needed during the considered week.
  For each service we have:
  - type of the service, expressed with the code;

- p_s, patient that need service s;
- f_s, frequency of the service s, i.e. number of times the service s must be performed during the week;
- e_s, number of days that need to pass before repeating the service, the value is equal to -1 if the service has frequency 1;
- g_s, vector of available days of service s. The value in position h in the vector is equal to 1 if the service s can be done on day h, 0 otherwise; the number of services is N_s.

3.2.5.2 Results

In this section we present the results found for each instance, first after using the *InitialSolution* algorithm, then after using the ILS method. We can observe the improvement of results before and after using the methaeuristic.
We also compare the results obtained by ILS to previous results found in others works on HHCP with different a metaheuristic but same instances and time limit.

Each algorithms consists in about 900 Python lines and results are obtained running the algorithm on 60 different instances.
In Table 3.2 we can observe the results founded for each instance using the *InitialSolution* algorithm, in Table 3.3 the new values after using ILS method and in Table 3.4 the percent variation of the values for each instance. In these tables R is the total number of requests and P is the number of patients that must be visited during the week. *u* is the maximum of the average working time between all nurses, *l* is the total number of different nurses visiting the same patients and *w* are the maximum needed extra hours between all days and all nurses. Finally *f* is the value of the objective function obtained as: $u + 10w + l$.

The percent variation for each value of *u, w, loyalty* and *f* is evaluated as follow:

$$var_X = \frac{X_{IS} - X_{ILS}}{X_{IS}}$$

and then multiplied by 100, where X_{IS} and X_{ILS} are the values found respectively after the initial solution and the ILS algorithm.
In Table 3.3 the value of the workload balance between al nurses is also shown. This value is expressed as the variance of the values of u_n, u_n is the average working time during the week for each nurse n. This is useful to compare solutions obtained after applying ILS method with the ones obtained in next chapters where the balance is introduced.

Initial Solution						
name	R	P	u	l	w	f
m540_s13	719	228	474.6	206	0	680.6
m540_s19	731	219	471.4	220	0	691.4
m540_s11	732	220	478.0	199	0	677.0
m550_s13	733	229	487.2	222	0	709.2
m540_s18	737	219	474.6	211	0	685.6
m540_s14	737	223	470.8	215	0	685.8
m540_s15	740	218	479.0	208	0	687.0
m540_s16	742	215	463.0	229	0	692.2
m550_s19	742	219	465.8	229	0	694.8
m540_s10	743	238	474.8	216	0	690.8
m560_s13	747	228	474.4	231	0	705.4
m540_s17	747	230	491.0	230	0	721.0
m550_s14	750	225	475.6	229	0	704.6
m550_s11	751	222	447.2	209	0	656.2
m550_s15	752	218	475.6	213	0	688.6
m560_s19	753	222	485.0	223	0	708.0
m550_s16	754	215	479.2	225	0	704.2
m550_s17	757	241	486.4	223	0	709.4
m560_s15	762	218	457.6	228	0	685.6
m560_s14	763	228	488.6	227	0	715.6
m570_s13	763	231	479.6	234	0	713.6
m560_s16	765	216	492.0	227	0	719.0
m560_s11	765	223	475.4	212	0	687.4
m570_s15	771	221	446.8	246	0	692.8
m540_s12	771	224	492.6	243	0	735.6
m560_s17	773	217	476.2	233	0	709.2
m570_s19	775	222	486.2	234	0	720.2
m560_s18	776	221	477.2	244	0	721.2
m570_s16	777	216	482.2	233	0	715.2
m570_s11	778	224	483.4	224	0	707.4
m570_s14	781	229	485.8	254	0	739.8
m550_s18	783	218	486.8	254	0	740.8
m550_s12	785	226	493.0	247	0	740.0
m580_s15	787	225	465.4	243	0	708.4
m570_s17	788	220	477.6	240	0	717.6
m580_s16	788	220	492.6	239	0	731.6
m580_s19	789	222	485.2	251	0	736.2

m580_s11	789	227	495.0	273	0	768.0
m560_s10	789	240	492.8	257	0	749.8
m570_s18	790	225	492.8	250	17	912.8
m580_s13	790	236	491.4	240	0	731.4
m580_s14	794	232	482.2	253	0	735.2
m550_s10	795	220	484.2	272	0	756.2
m590_s15	797	229	489.2	253	0	742.2
m560_s12	800	228	494.4	272	0	766.4
m590_s11	800	228	481.8	230	0	711.8
m570_s10	801	241	485.8	250	0	735.8
m590_s19	803	224	490.6	252	0	742.6
m590_s16	804	223	479.8	257	0	736.8
m580_s17	806	223	475.8	254	0	729.8
m590_s14	807	234	486.2	257	0	743.2
m590_s13	808	236	488.8	259	0	747.8
m580_s18	811	227	498.8	254	73	1482.8
m570_s12	812	231	492.4	278	0	770.4
m580_s10	814	225	475.8	247	0	722.8
m590_s18	822	229	512.8	259	107	1841.8
m580_s12	825	232	495.0	273	0	768.0
m590_s10	834	223	494.0	268	0	762.0
m590_s12	841	233	502.2	287	105.2	1839.2
m590_s17	848	228	493.8	287	0	780.8

Table 3.2: Results obtained using Initial Solution algorithm

Improved solution							
name	R	P	u	l	w	f	b
m540_s13	719	228	380.2	124	0	504.2	141.6
m540_s19	731	219	390.6	133	0	523.6	1074.0
m540_s11	732	220	376.0	118	0	494.0	327.6
m550_s13	733	229	407.0	128	0	535.0	407.3
m540_s18	737	219	395.4	148	0	543.4	141.5
m540_s14	737	223	385.8	129	0	514.8	548.9
m540_s15	740	218	384.6	125	0	509.6	644.7
m540_s16	754	215	385.6	129	0	517.8	421.1
m550_s19	742	219	398.0	123	0	521.0	365.3
m540_s10	743	238	382.0	141	0	523.0	1109.5

m560_s13	747	228	400.0	129	0	529.0	312.0
m540_s17	747	237	383.0	129	0	512.0	398.9
m550_s14	750	225	392.8	128	0	520.8	341.8
m550_s11	751	222	377.8	115	0	492.8	250.5
m550_s15	752	218	389.2	124	0	513.2	393.8
m560_s19	753	222	396.8	133	0	529.8	313.4
m550_s16	754	215	393.4	129	0	522.4	410.0
m550_s17	757	241	390.0	135	0	525.0	665.1
m560_s15	762	218	395.8	135	0	530.8	527.0
m560_s14	763	228	393.4	124	0	517.4	313.6
m570_s13	763	231	411.8	133	0	543.0	583.7
m560_s16	765	216	403.7	137	0	540.7	655.0
m560_s11	765	223	390.2	125	0	515.2	230.7
m570_s15	771	221	400.7	136	0	536.7	454.2
m540_s12	771	224	408.0	148	0	556.0	452.3
m560_s17	773	217	396.6	164	0	560.6	946.4
m570_s19	775	222	402.7	142	0	544.7	494.8
m560_s18	776	221	422.0	147	0	569.0	566.3
m570_s16	777	216	404.7	130	0	534.7	282.5
m570_s11	778	224	393.9	118	0	511.8	245.3
m570_s14	781	229	405.8	134	0	539.7	240.5
m550_s18	783	218	398.6	154	0	552.6	430.4
m550_s12	785	226	412.4	147	0	559.4	217.0
m580_s15	787	225	411.7	151	0	562.7	256.2
m570_s17	788	220	403.4	163	0	566.4	563.3
m580_s16	788	220	407.8	144	0	551.7	1319.0
m580_s19	789	222	415.4	148	0	563.4	420.0
m580_s11	789	227	436.0	178	0	614.0	339.1
m560_s10	789	240	417.4	149	0	566.4	696.5
m570_s18	790	225	422.8	159	0	581.7	258.0
m580_s13	790	236	413.0	141	0	554.0	549.2
m580_s14	794	232	409.4	146	0	555.4	505.7
m550_s10	795	220	419.4	164	0	583.4	331.4
m590_s15	797	229	418.7	145	0	563.7	973.0
m560_s12	800	228	425.0	174	0	599.0	544.9
m590_s11	800	228	407.2	129	0	536.2	467.7
m570_s10	801	241	420.4	146	0	569.4	310.1
m590_s19	803	224	419.6	149	0	568.6	914.3
m590_s16	804	223	418.6	147	0	565.6	501.1
m580_s17	806	223	425.2	167	0	592.2	309.6

m590_s14	807	234	423.0	139	0	562.0	319.4
m590_s13	808	236	429.7	145	0	574.7	442.6
m580_s18	811	227	430.2	167	0	597.2	485.2
m570_s12	812	231	435.6	172	0	607.6	494.5
m580_s10	814	225	421.6	139	0	560.6	447.3
m590_s18	822	229	436.6	171	0	607.6	409.8
m580_s12	825	232	437.6	184	0	621.6	384.3
m590_s10	834	223	435.8	160	0	595.0	1156.9
m590_s12	841	233	457.6	168	0	625.6	347.5
m590_s17	848	228	436.4	183	0	619.4	577.7

Table 3.4: Results obtained after applying the ILS algorithm

Percent variation						
name	R	P	u	l	w	f
m540_s13	719	228	19.89	39.80	0	25.92
m540_s19	731	219	17.14	39.54	0	24.27
m540_s11	732	220	21.34	40.70	0	27.03
m550_s13	733	229	16.46	42.34	0	24.56
m540_s18	737	219	16.69	29.85	0	20.74
m540_s14	737	223	18.05	40.00	0	24.93
m540_s15	740	218	19.71	39.90	0	25.82
m540_s16	754	215	16.72	43.67	0	25.19
m550_s19	742	219	14.56	46.29	0	25.01
m540_s10	743	238	19.55	33.80	0	24.29
m560_s13	747	228	15.68	44.16	0	25.00
m540_s17	747	237	21.99	43.91	0	28.99
m550_s14	750	225	17.41	44.10	0	26.08
m550_s11	751	222	15.52	44.97	0	24.90
m550_s15	752	218	18.17	41.78	0	25.47
m560_s19	753	222	18.19	40.36	0	25.17
m550_s16	754	215	17.90	42.67	0	25.82
m550_s17	757	241	19.82	39.46	0	25.99
m560_s15	762	218	13.50	40.79	0	22.58
m560_s14	763	228	19.48	45.37	0	27.69
m570_s13	763	231	14.14	43.16	0	23.91
m560_s16	765	216	17.95	39.65	0	24.79
m560_s11	765	223	17.92	41.0	0	25.05

m570_s15	771	221	10.31	44.71	0	22.53
m540_s12	771	224	17.17	39.09	0	24.41
m560_s17	773	217	19.48	29.61	0	20.95
m570_s19	775	222	17.17	39.39	0	24.36
m560_s18	776	221	11.56	39.75	0	21.10
m570_s16	777	216	16.07	44.20	0	25.23
m570_s11	778	224	18.51	47.32	0	27.76
m570_s14	781	229	16.46	47.24	0	27.04
m550_s18	783	218	18.11	39.37	0	25.40
m550_s12	785	226	16.34	40.48	0	24.40
m580_s15	787	225	11.53	37.86	0	20.56
m570_s17	788	220	15.53	32.08	0	21.07
m580_s16	788	220	17.21	39.74	0	24.58
m580_s19	789	222	14.38	41.03	0	23.47
m580_s11	789	227	11.91	34.79	0	20.05
m560_s10	789	240	15.30	42.02	0	24.45
m570_s18	790	225	14.20	36.4	100	36.27
m580_s13	790	236	15.95	41.25	0	24.25
m580_s14	794	232	15.09	42.29	0	24.45
m550_s10	795	220	13.38	39.70	0	22.85
m590_s15	797	229	14.41	42.68	0	24.05
m560_s12	800	228	14.00	36.02	0	21.84
m590_s11	800	228	15.48	43.91	0	24.63
m570_s10	801	241	13.34	41.60	0	22.61
m590_s19	803	224	14.47	40.87	0	23.43
m590_s16	804	223	12.75	42.80	0	23.23
m580_s17	806	223	10.63	34.25	0	18.85
m590_s14	807	234	12.98	45.91	0	24.38
m590_s13	808	236	12.09	44.01	0	23.14
m580_s18	811	227	13.75	34.25	100	59.72
m570_s12	812	231	11.53	38.12	0	21.13
m580_s10	814	225	11.39	49.20	0	22.44
m590_s18	822	229	14.85	33.97	100	67.01
m580_s12	825	232	11.59	32.60	0	19.06
m590_s10	834	223	11.78	40.29	0	21.91
m590_s12	841	233	08.88	41.46	100	65.98
m590_s17	848	228	11.62	36.23	0	20.67

Table 3.6: Percentage decrease between Initial Solution and ILS results

We can compare our results with results obtained by previous studies using Adaptive Laerge Neighborhood Search method with time limits 1500 seconds.

Comparison of results		
	ILS	ALNS
Obj. Funct.	551.83	508.24
Cost	405.89	482.54
Loyalty	143.7	25.6

Table 3.7: Comparison obtained of results with previous work

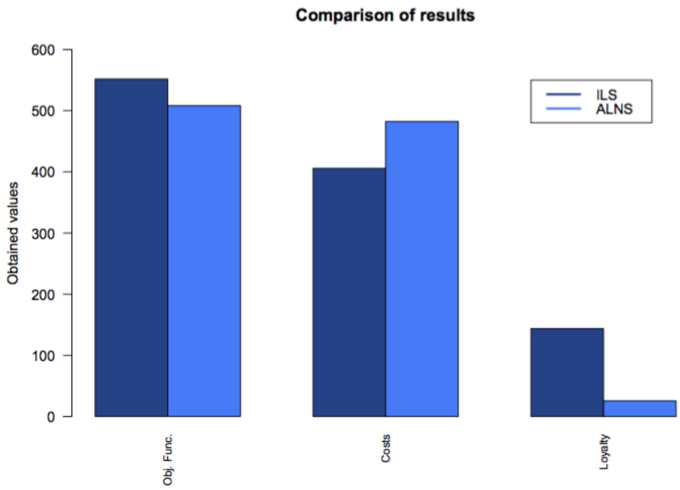

Figure 3.10: Comparison of obtained results with previous work

What we are comparing in this figure are the averages of the values of the objective function *f*, the costs $(u + 10w)$ and the *loyalty*.
We can observe that the mean of the objective functions overall the instances is more or less the same. The mean of costs $(u + 10w)$, obtained with ILS, is better, while the loyalty is worse.
In future works what we can do is to focus the local search more on the loyalty trying to improve it introducing others moves or rearrange the structure of the algorithm.

Chapter 4

HHCP as a Multiobjective Problem

The HHCP is a multiobjective problem, since the objectives described (cost, loyalty and balance of workload) can not in reality be expressed in a unique formula. In previous chapter we considered the objective function:

$$f = \quad \alpha_0 u + \alpha_1 \sum_{p \in P} (\sum_{j \in N} z_p^n - 1) + \alpha_2 w$$

with $\alpha_0 = \alpha_1 = 1$ and $\alpha_2 = 10$ just to help to focus the search for a good solution and to compare the solutions of ILS with other methods as the ALNS.
In this section we treat the HHCP as a multiobjective and a decision has to be taken according to the preferences of the decision maker between the solutions in the approximate Pareto frontier.
In this section we have considered the previous two objectives, costs and loyalty, and now we also consider another important objective: the balance of workload between nurses. This objective is related with the guarantee of equi distribution of working time.
In the following sections we explain in more details the multi objective problems and our new approach to HHCP.

4.1 Multiobjective problems

A multiobjective problem is a problem of the form:

$$min\{f_1(\overline{x}), f_2(\overline{x}), \dots, f_k(\overline{x})\}$$
$$\text{subject to } \overline{x} \in X$$

where we have $k \geq 2$ objective functions and $f(\overline{x}) = (f_1(\overline{x}), f_2(\overline{x}), \ldots, f_k(\overline{x}))$ is the vector of values of these objective functions. $\overline{x} = (x_1, \ldots, x_n)$ is the decision vector $\in \mathbb{R}^n$ and must belong to the feasible region X.
The objective is to minimize the value of all the objective functions simultaneously, of course a solution that minimizes all the objective functions can be found only if there is no conflict between them. Usually this is not true and the objective functions are in conflict and also they are usually incommensurable, i.e. they have different units.
The feasible objective region Z, i.e. the region of all possible values of vector $f(\overline{x})$ with $\overline{x} \in X$, can be obtained. This region is formed by vector $\overline{z} = (z_1, z_2, \ldots, z_k)$ where z_i is $f_i(\overline{x})$ $\forall i = 1, \ldots, k$.
To solve this kind of problem the idea is to find a compromise between the objective functions instead of a single solution (global optimization).

4.1.1 Pareto optimality

For a multiobjective problem the concept of optimal solution is difficult to define. There are several objective functions that measure the solution in different ways. Therefore, within multiobjective, the usual idea is to obtain the set of non dominated solutions and then the decision makers choose one among these ones.
Theoretically a solution $\overline{x}$ is dominated by $\overline{x'}$ iff :

$$\begin{cases} f_i(\overline{x'}) \leq f_i(\overline{x}) \quad \forall i \in 1, \ldots, k \\ \exists j \in 1, \ldots, k : f_j(\overline{x'}) < f_j(\overline{x}) \end{cases}$$

We said that $\overline{x}$ is strictly dominated by $\overline{x'}$ if the inequalities are strict. The corresponding vector $f(\overline{x'})$ is called non dominated.
With this definition we can define the region of *non dominated* solutions X' as the set of solutions $\overline{x}$ s.t. these solutions are non dominated by any other solution in the feasible region X.
From this definition we can introduce the Pareto optimal region as the non dominated set of the entire feasible search space X. Of course this space is formed by several non dominated solutions. So a solution $\overline{x}$ is said to be efficient iff:

$$\nexists \overline{x'} \in X | f_i(\overline{x'}) \leq f_i(\overline{x}) \quad \forall i \in 1, \ldots, k$$

The Pareto frontier is the set of corresponding vectors in Z of the solutions in the Pareto optimal region. The dimension of the Pareto Frontier depends on the number of objectives.

The decision makers' preferences must be taken into account to choose which is the best solution among those in the set. This preferences can be given *a priori,*

a posteriori or *progressively*.
In a priori techniques the preferences are expressed before the optimization procedure, in this way we can know which are the best subregions to investigate. This approach requires rich information from the decision makers in order to reduce the problem to a single-objective problem. Although in many situations uncertainty about the precise appearance of the decision makers utility function is often presented. A posteriori approaches do not require any information from the decision makers, the Pareto set is computed off-line allowing them to perform other tasks while waiting for the results. At the end of the optimization procedure the Pareto set is presented and the decision makers can choose among solutions contained in the set. In progressive techniques only partial information is required and the decision makers' preferences are given step by step during the search procedure. The positive aspect of this procedure is that only few alternatives have to be computed compared to the entire Pareto set, on the other hand the decision makers need to be present during the resolution procedure.

4.2 Pareto optimality on Iterated Local Search

The Pareto optimality concept has been used in Iterated Local Search algorithm to solve multiobjective problems. The motivation behind this algorithm can be seen in the increasing demand for simple but effective heuristics for the resolution of complex multiobjective optimization problems.
In Geiger (2006), a new algorithm called Pareto Iterated Local Search (PILS) is introduced combining two principle of local search: Variable Neighbourhood Search and Iterated Local Search. The concept is successfully tested on permutation flow shop scheduling problems under multiple objectives. General framework for the Pareto optimality in Iterated Local search is explained in the paper in section 3. PILS algorithm combines the two main driving forces of local search, intensification and diversification, into a single algorithm.
In Geiger (2008) a similar algorithm is applied to biobjective portfolio optimization problem.
In this work we introduced an algorithm inspired to PILS: the Multiobjective Iterated Local Search. The algorithm is expressed below.

Algorithm 14: *Multiobjective Iterated Local Search*

$s_0 =$ GenerateInitialSolution;
$\overline{S^*} = LocalSearch(s_0)$;
PO_{pool} = *create pool of Pareto optimal Solutions from set* $\overline{S^*}$;
repeat
    s^* = *solution picked up from* PO_{pool} *with some criteria*;
    $s' = Perturbation(s*, history)$;
    $\overline{S^{*\prime}} = LocalSearch(s')$;
    update PO_{pool} *with solutions in* $\overline{S^{*\prime}}$;
until *time≥TimeLimit*;
return PO_{pool};

The algorithm starts from an initial solution s_0, an improving search is performed until a set $\overline{S^*}$ of locally optimal alternatives is identified and store in a set $P0_{pool}$ representing the approximation of the true Pareto set. After the identification of a locally optimal set, a perturbation operator is performed on a solution s^* chosen with some criteria from $P0_{pool}$, the local search is then used on the perturbed solution s'. The archive of the currently best solutions is updated during the search. The final PO_{pool} set of approximate Pareto optimal solutions is then returned.

4.3 Resolution of HHCP as Multiobjective Problem

The HHCP can be formulated as a multiobjective problem by just modifying the objective function as follows:

$$min[f_1, f_2, f_3]$$

with:

$$f_1 = u + 10w$$
$$f_2 = \sum_{p \in P}(\sum_{j \in N} z_p^n - 1)$$
$$f_3 = Var(\overline{u_n})$$

We can observe that f_1 represents the costs, which consists in the maximum average of workload between all the nurses plus 10 times the maximum extra workload between all the tours.

f_2 represents the loyalty and it is the total number of different nurses visiting all the patients.
The last objective function f_3 represents the balance of workload between nurses, evaluated by the variance of $\overline{u_n}$. $\overline{u_n}$ is the vector of nurses' average working time during the week.

With this objective functions we represent in a more realistic way the objectives of the home health care planning. Now, the solution method should output the set of Pareto optimal solutions or at least an approximation of it.
A solution has to be chosen among the solutions in the pool of Pareto optimal solutions. In general decision makers should be interrogate to obtain informations about which is the best solution, i.e. which is the vector of weights on the objective functions. As having informations from decision makers can be difficult, we pseudo randomly choose a solution in the pool and at the end we present all the pool of approximate Pareto optimal solutions.

4.3.1 Algorithm: Multiobjective ILS for HHCP

Next we describe the modified algorithm based on ILS to solve the HHCP. This algorithm, since it is a metaheuristic, output the set of approximate Pareto optimal solutions.

Algorithm 15: *Multiobjective Iterated Local Search for HHCP*

$s_0 =$ GenerateInitialSolution;
$\overline{S^*} = LocalSearch(s_0)$;
$PO_{pool} =$ *create pool of Pareto optimal Solutions with* $\overline{S*}$;
repeat
    $crit = Uniform(1,3)$;
    sort PO_{pool} *by crit*;
    $sol = Geom(p)$;
    $s^* = PO_{pool}[sol]$;
    $s' = Perturbation(s^*, history)$;
    $\overline{S^{*\prime}} = LocalSearch(s')$;
    update PO_{pool} *with solutions in* $\overline{S^{*\prime}}$;
until *time*$\geq$*TimeLimit*;
return PO_{pool};

Next, we describe in detail this algorithm.
The algorithm is similar to the ILS algorithm but this time we have a pool of

Pareto optimal solutions PO_{pool} which is updated every time a set of solutions is found with the Local Search. After running the Local Search and update the pool we pseudo randomly choose the solution on which the ILS has to be applied.
What we do to choose the solution on which apply ILS is to choose randomly, using a *Uniform* distribution, one of the objective functions, sort the results in the pool of Pareto optimal solutions according to increasing value of the chosen criterion and then select a solution using a Geometric distribution. This method allows us to choose one solution with high value of the objective function selected, thank to the geometric distribution and the ordered values, and then improve it using ILS algorithm. The reason for choosing randomly the objective function to improve is due to the absence of decision makers preferences, in future the selection of it can be done following the subjection of decision makers.
At the end of the algorithm we can represent all the solutions found and the last updated approximation of the Pareto optimal frontier.
Comparison can be done between results obtained with ILS using the objective function f and the last results obtained with ILS using the Pareto optimal frontier. The best solution can be chosen in the pool of Pareto optimal solutions according to the decision maker's preferences.

4.3.2 Results

We present now the results obtained using the MoILS algorithm to solve the multi objective HHCP. For each instance the final result is a pool of Pareto optimal solutions with different values of *Costs*, *Loyalty* and *Balance*. As explained in previous chapters the first one is expressed in terms of times as $u + 10w$, the second in terms of different numbers of nurses visiting the same patient and the last one is the variance of the average working time for each nurses.
In real applications, the decision makers take into account the output solutions, i.e. the approximate Pareto Optimal solutions, and use other criteria to make the final decision.
Two graphics corresponding to two different instances are presented to show an example of representation of the final pool of Pareto optimal solutions.
In Table 4.2 the average values of the solutions in the pool in terms of *Costs*, *Loyalty* and *Balance* is presented for each instance. In addition also standard deviations for the values of the objective functions in the pool of approximate Pareto optimal solutions are shown. It can be noticed that standard deviation of *balance* values is higher, than for the other two objective functions. As in previous tables R is the total number of request during the week and P the number of different patients that have to be visited.

Mean values of objective functions								
name	R	P	u+10w	*sd*	l	*sd*	b	*sd*
m540_s13	719	228	394.0	6.8	130.8	3.5	86.9	68.8
m540_s19	731	219	401.3	10.1	145.4	5.1	329.6	270.4
m540_s11	732	220	376.8	4.3	137.8	5.5	20.6	43.6
m550_s13	733	229	395.4	7.8	151.0	5.0	31.7	37.6
m540_s18	737	219	392.0	1.2	157.4	3.0	10.1	2.2
m540_s14	737	223	391.7	8.0	130.9	3.2	69.2	56.1
m540_s15	740	218	389.5	2.4	139.7	8.9	102.2	18.8
m540_s16	754	215	398.4	8.9	149.4	5.6	115.7	75.9
m550_s19	742	219	401.3	16.4	147.8	5.7	86.2	326.5
m540_s10	743	238	387.2	1.5	154.6	15.1	34.5	6.7
m560_s13	747	228	407.4	9.8	158.7	6.2	132.5	168.6
m540_s17	747	237	390.6	4.3	144.6	4.5	56.7	13.5
m550_s14	750	225	407.9	4.9	128.0	2.6	163.1	82.4
m550_s11	751	222	387.9	2.3	134.7	21.5	41.2	9.5
m550_s15	752	218	405.9	4.3	142.4	8.2	173.5	36.8
m560_s19	753	222	406.4	7.5	139.3	3.8	110.5	87.7
m550_s16	754	215	400.4	10.1	141.7	4.0	102.0	50.8
m550_s17	757	241	399.3	2.4	144.0	4.5	71.0	14.9
m560_s15	762	218	409.3	3.5	155.0	3.5	81.3	102.7
m560_s14	763	228	389.4	2.1	132.0	3.9	25.6	24.9
m570_s13	763	231	409.2	2.2	149.4	7.1	68.5	14.8
m560_s16	765	216	410.6	7.2	158.5	3.7	172.5	119.8
m560_s11	765	223	394.3	8.4	138.1	5.9	62.2	113.2
m570_s15	771	221	404.3	3.5	160.6	8.2	27.9	34.0
m540_s12	771	224	413.2	2.5	152.9	9.0	57.1	14.2
m560_s17	773	217	412.1	3.8	170.1	5.4	195.0	29.5
m570_s19	775	222	414.4	9.2	152.2	4.9	153.4	113.5
m560_s18	776	221	426.1	14.8	161.8	7.3	74.3	12.3
m570_s16	777	216	406.5	2.7	143.3	2.8	113.1	26.6
m570_s11	778	224	398.5	5.6	127.5	3.1	32.2	36.1
m570_s14	781	229	416.7	3.2	150.6	9.8	100.8	23.4
m550_s18	783	218	413.6	4.8	168.7	24.8	119.8	28.9
m550_s12	785	226	423.3	11.4	160.5	4.5	112.0	22.3
m580_s15	787	225	418.8	4.9	149.2	4.0	103.7	71.6
m570_s17	788	220	426.0	5.8	171.3	11.7	274.0	66.1
m580_s16	788	220	428.4	7.3	145.9	4.0	219.8	148.9
m580_s19	789	222	431.4	4.9	142.8	4.2	189.3	38.5

m580_s11	789	227	407.4	2.9	138.5	22.2	92.3	20.6
m560_s10	789	240	421.9	4.9	166.6	3.4	89.2	89.6
m570_s18	790	225	467.2	0.9	173.2	4.1	85.9	68.9
m580_s13	790	236	425.8	3.4	146.9	6.3	117.1	22.9
m580_s14	794	232	421.7	5.4	140.5	1.2	131.8	9.2
m550_s10	795	220	433.0	5.6	168.6	5.4	183.2	42.7
m590_s15	797	229	453.4	5.6	164.0	4.8	309.2	202.3
m560_s12	800	228	423.9	2.7	199.0	29.2	81.4	16.3
m590_s11	800	228	421.8	4.7	134.2	7.5	174.0	41.9
m570_s10	801	241	424.9	6.9	157.1	5.6	84.8	68.1
m590_s19	803	224	434.3	6.2	158.3	4.0	288.2	167.2
m590_s16	804	223	424.1	8.6	168.8	6.3	106.9	81.7
m580_s17	806	223	425.0	10.7	182.0	8.7	101.0	65.4
m590_s14	807	234	432.7	3.1	141.0	3.2	140.3	44.4
m590_s13	808	236	429.9	8.6	165.7	5.4	127.9	210.9
m580_s18	811	227	521.3	99.2	174.3	16.2	152.4	23.4
m570_s12	812	231	444.2	4.2	198.8	5.9	171.6	80.7
m580_s10	814	225	429.8	3.7	145.6	5.9	116.1	25.9
m590_s18	822	229	606.9	76.7	174.8	6.2	243.3	37.9
m580_s12	825	232	439.6	3.0	192.1	24.5	122.8	11.8
m590_s10	834	223	452.6	2.8	153.5	2.9	300.3	71.7
m590_s12	841	233	475.5	45.3	202.6	8.1	140.9	7.8
m590_s17	848	228	440.4	5.8	183.4	5.1	146.5	42.2

Table 4.2: Average of the results in pool of approximate Pareto optimal solutions with MoILS

Comparing this results with the ones obtained after ILS, we can observe that both costs and *loyalty* maintain almost the same value, sometimes the *loyalty* is better, while the balance is always lower after MoILS algorithm. Of course we have to consider that results presented in Table 4.2 are a mean between all the solutions in the approximate pool of Pareto optimal ones.

Concluding we can say that considering the multiobjective model and the proposed MoILS the solutions make more sense for the decision makers. Solutions are in fact more realistic and decision makers can use others informations, difficult to consider on the model because non quantitative, to make the final decision.

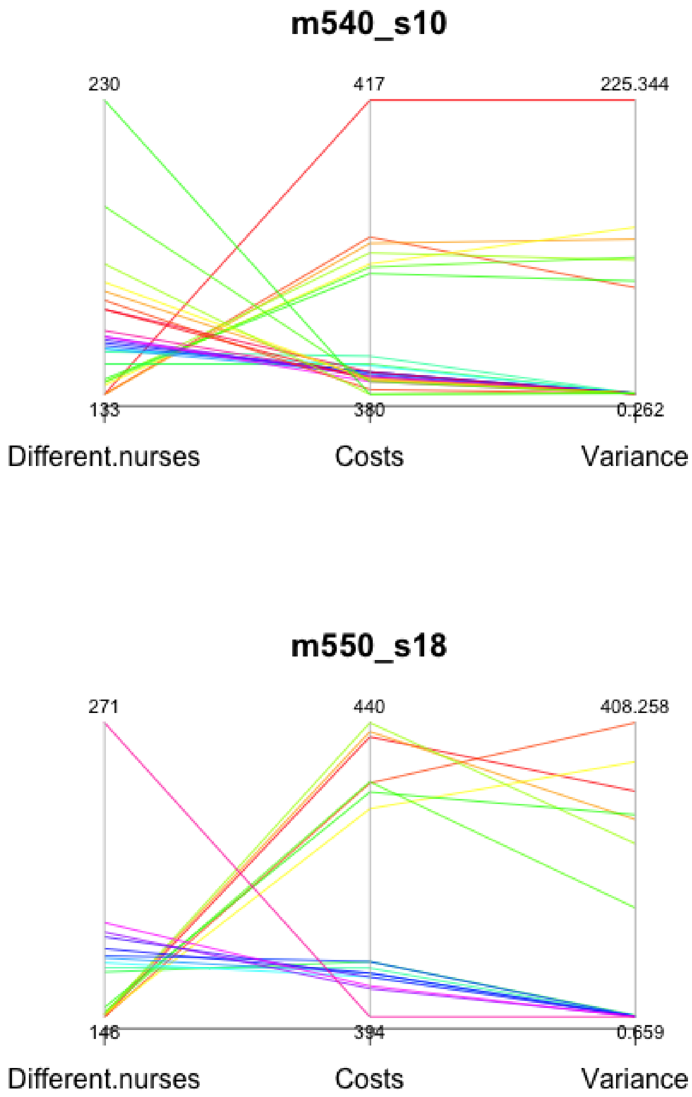

Figure 4.1: Pool of approximate Pareto optimal solutions of two different instances in parallel coordinates using MoILS

Chapter 5

HHCP as a Stochastic Problem

In this chapter stochastic service times are introduced to be more closed to the real problem. The SimILS algorithm is merged with the multiobjective ILS in order to solve the stochastic multiobjective HHCP. A simulation with a triangular distribution is applied to the service times in order to have a simulated value of the objective functions. The idea is to create and update the pool of approximate Pareto optimal solutions using the stochastic values and not the deterministic ones.

5.1 Stochastic HHCP

The complexity of the HHCP is mainly due to the large number of assisted patients, synchronization of resources and service delivery in an often vast territory. However we need also to consider random events that can effect the delivery of service and the feasibility of resource plans. Randomness can be found in different parts of the problem such as the changing in patients' conditions, resource availability, duration of operator transfers in the territory and service times.

The assumption of deterministic data simplifies the mathematical model to become tractable, but of course gives a less accurate model that does not reflect the uncertainty of the problem.

An interesting work was done by Lanzarone, Matta and Scaccabarozzi (2009) for estimating the number of services needed from a patient considering changing in his conditions, studying the evolution of the patient conditions as a Markov Chain. What we focus on is the randomness in service time, while maintaining deterministic the travel times.

After solving the HHCP under deterministic conditions, i.e. with deterministic service time, we would like to extend the problem introducing uncertainty. To solve this new problem we use the SimILS algorithm. Previous studies observed that exact methods used to solve stochastic Combinatorial Optimization Problems

are useful only with small and medium-size problems or with a reduced set of distributions. Another approach to solve stochastic Combinatorial Optimization Problems is to use metaheuristcs, that are more appropriate when the complexity of the problem is high. In the last decades there has been a growing interest for combining simulation and optimization problems using simheuristics to solve stochastic problems. So following the new trend SimILS combines the ILS method with simulation techniques, see Lourenço, Grasa and Juan (2014).
We now propose, to consider stochastic service times and to solve the HHCP, a new algorithm modifying the ILS algorithm for multiobjective problems using the SimILS. Introducing stochastic service times we have to consider that we are not using a simple ILS algorithm but a modified one for multiobjective.
The proposed algorithm is not only SimILS because it is applied to a multiobjective problem using the Pareto pool as in chapter 4. The new proposed algorithm is called Multiobjective Iterated Local Search, MoSimILS.
In the next sections we first explain general framework for SimILS and then how we joint it with MoILS proposing the MoSimILS.

5.2 SimILS

SimILS is an extension of the ILS method. The difference is that ILS works well on Combinatorial Optimization Problems under deterministic scenarios, but in real life uncertainty is often present.
SimILS integrates simulation methods inside the classical ILS framework in order to be able to solve stochastic Combinatorial Optimization Problems.
Sometimes we can use exact methods to solve stochastic Combinatorial Optimization Problems, but these methods are hard to use when we have large size instances or instances in which general distributions are used to model the randomness of the system. Approximate methods, as heuristics and metaheuristics, are more appropriate as the complexity of the problem grows. A way to solve stochastic Combinatorial Optimization Problems is to combine simulation and optimization techniques, using simheuristics. ILS is an excellent candidate to be combined with simulation because has all important attributes of a metaheuristic: it is accurate, speed, simple and flexible, see Cordeau et al. (2002). ILS requires few parameters and they can be easily adapted to different variants or extensions of a given problem.
SimILS integrates simulation at some specific step of the ILS algorithm, and was proposed by Lourenço, Grasas and Juan (2014).

Algorithm 16: *SimILS*

$s_0 =$ GenerateInitialSolution;
$s^* = LocalSearch(s_0)$;
$(s^*, sf(s^*), statistics) = Simulation(s^*, long)$;
repeat
  $s' = Perturbation(s^*, history)$;
  $s^{*'} = LocalSearch(s')$;
  $(s^{*'}, sf(s^*), statistics) = Simulation(s^*, short)$;
  $s^* = AcceptanceCriterion(s^*, s^{*'}, history)$;
until *termination condition met*;
$(s^*, sf(s^*), statistics) = Simulation(s^*, long)$;
return s^*, $sf(s^*)$

We can notice that particular simulations are added after applying the *LocalSearch* procedure together with a parameter that indicates if the simulation has to be run for short or long time, so we obtain a simulated objective function $sf(\cdot)$. A simulation component is also inserted at the end of the ILS precess. Simulation is used on the solution found after the Local Search and the idea is to simulate a fixed number from a given distribution and consider the mean of these value, as in a Monte Carlo simulation. The use of simulation is oriented to estimate the expected cost value of new a solution.
To avoid long computational time the simulation should be applied just over a selected and reduced subset of solutions. For that reason a short run simulation is performed in the main loop and only on a selected subset of solutions. In other hands since the accuracy of a simulation depends on the number of runs executed a long run simulations is performed at the end of the loop to the final subset of solutions.
SimILS algorithm works different if the stochastic part is in the constraints, in the objective function or in both of them. In the case with stochastic constraints what the algorithm does is to check if the new generated solution after the simulation verify the constraints with a certain probability, only in this case the simulation is accepted. In problems with stochastic objective functions the algorithm maintains two best solutions: the best local optimum solution for the deterministic objective function and the best local optimum solution for the stochastic objective function obtained after the simulation. The second one is updated only if the new simulated value of the objective function is better that the previous one, also simulated. Combining the two algorithms we obtain the one for problem with both stochastic constraints and objective function.
SimILS has been recently developed but already some applications can be found. This extension of ILS has been used to solve VRP with stochastic demands by

Juan et al. (2011-2013) or the Permutation Flow Shop Problem with Stochastic Processing Times by Juan et al. (2014).

5.3 SimILS with multiobjective problems: MOSimILS

In this chapter we are modifying the HHCP, introduced in previous chapters, from its deterministic version to a new stochastic version. The goal of this section is to introduce uncertainty maintaining the multiobjectivity, to obtain this result the MoILS has to be merged with the SimILS. One of the main innovation of this thesis is the introduction of a new algorithm, the Multi Objective Simulated Iterated Local Search. The idea above this algorithm is to apply ILS method considering stochastic parameters introducing a Monte Carlo simulation, not only on a single solution but on a set of solutions representing the approximate pool of Pareto optimal ones.
The MoSimILS algorithm is then presented.

Algorithm 17: *Multiobjective Simulated Iterated Local Search*

$s_0 =$ GenerateInitialSolution;
$\overline{S^*} = LocalSearch(s_0)$;
PO_{pool} = *create pool of Pareto optimal Solutions with* $\overline{S^*}$;
foreach $s \in PO_{pool}$ **do**
    $(s, sf(s), statistics)$ = *Simulation(s, short)*;
update PO_{pool};
repeat
    s^* = *solution picked up from* PO_{pool} *with some criteria*;
    $s' = Perturbation(s^*, history)$;
    $\overline{s^{*\prime}} = LocalSearch(s')$;
    generate PO^1_{pool};
    foreach $s \in PO^1_{pool}$ **do**
        $(s, sf(s), statistics)$ = *Simulation(s, short)*;
    update PO_{pool};
until *time*$\geq$*TimeLimit*;
foreach $s \in PO_{pool}$ **do**
    $(s, sf(s), statistics)$ = *Simulation(s, long)*;
update PO_{pool};
return PO_{pool} with simulated values;

What it can be seen is that MoSimILS maintains the main structure of MoILS

algorithm introducing a simulation part that is characteristic of SimILS method. The idea is to simulate the solution found during the *LocalSearch* and update the pool of approximate Pareto optimal solutions using the simulated values.
The first part of the algorithm works as in MoILS: an initial solution is generated and *LocalSearch* is applied on this solution. A first pool of approximate Pareto optimal solutions is estimated. To introduce uncertainty a short run simulation is applied on all the solutions on this set and stochastic values are obtained and the PO_{pool} is then updated.
After this first part the algorithm starts with the repetition of *Perturbation* and *LocalSearch* starting on a solution picked up from the PO_{pool} following some criteria depending on the different interest we have on optimize one or another objective function. Another difference between MOILS and MOSimILS in then introduced in line 12 of the algorithm. After generating the PO^1_{pool} with the solutions found during the *LocalSearch* a short simulation is applied before updating the PO_{pool}. It is important to specify that this simulation is applied not on all the solutions found during the *LocalSearch* but only on the set of approximate Pareto optimal solution in order to minimize the needed computational time. For the same reason the simulation at this step is chosen to be a short run simulation.
Before presenting the final pool of approximate Pareto optimal solutions a long run simulation is applied on all the solutions in the set and then the PO_{pool} is updated again.

With this algorithm Multiobjective problems with stochastic parameters can be solved using the main idea above ILS method, the iteration of a Local Search. As in ILS algorithm, how to find an initial solution, the *Local Search* and the *Perturbation* methods have to be chosen. Then the pool of Pareto optimal solutions has to be introduced in order to consider the multiobjectivity of the problem, as in MoILS. Finally a Monte Carlo simulation is used to introduce uncertainty, the distribution for the simulation has to be chosen, as in SimILS algorithm.
MoSimILS allows us to solve HHCP as a Multiobjective and stochastic problem with stochastic service times.

5.4 MOSimILS for HHCP

In this section we present the application of the MoSimILS to our HHCP considered as multiobjective problem with stochastic service times. We would like to consider a more realistic problem using stochastic service times instead of deterministic ones. We have to explain in details which choices are taken to apply the MoSimILS algorithm to HHC multiobjective problem.
The way the stochastic service times are generated is explained next. To simulate

the stochastic service times we use a triangular distribution with $min = a_s - 5$ and $max = a_s + 5$ to generate N different values of a_s. It is obvious that with different service times we obtain different values of objective functions.
The triangular distribution is a common method to generate service times and can be a good choice as we do not have big data about the service times, but we just have one value. So as we do not neither have the minimum or maximum times needed to perform a service we suppose that a nurse can stay in a patient's home for 5 minutes more or 5 minutes less. What we do is consider the data clustered around a central one. Of course a more accurate study can be done with more informations on the uncertainty of service times. The future addition of new informations about service times will not effect the design of the algorithm but just the Monte Carlo simulation should be adapted to the appropriate distribution.

Algorithm 18: *Multiobjective Simulated Iterated Local Search for HHCP*

$s_0 =$ GenerateInitialSolution;
$\overline{S^*} = LocalSearch(s_0)$;
PO_{pool} = *create pool of Pareto optimal Solutions with* $\overline{S^*}$;
foreach $s \in PO_{pool}$ **do**
  $(s, sf(s), statistics)$ = *Simulation(s, short)*;
update PO_{pool};
repeat
  $crit = Uniform(1, k)$;
  sort PO_{pool} *by crit*;
  $sol = Geom(p)$;
  $s^* = PO_{pool}[sol]$;
  $s' = Perturbation(s*, history)$;
  $\overline{S^{*\prime}} = LocalSearch(s')$;
  generate PO^1_{pool};
  foreach $s \in PO^1_{pool}$ **do**
    $(s, sf(s), statistics)$ = *Simulation(s, short)*;
  update PO_{pool};
until *time*$\geq$*TimeLimit*;
foreach $s \in PO_{pool}$ **do**
  $(s, sf(s), statistics)$ = *Simulation(s, long)*;
update PO_{pool};
return PO_{pool} with simulated values;

What is done in the algorithm is that given a solution to simulate we generate a given number N of simulated service times and with that we can obtain N different

simulated Cost Matrices. With this matrices we generate a vector of N new values for the solutions. The mean of these values is the new simulated solution. N is set with different number depending if we are performing a short or a long run simulation; in the first case we consider $N = 30$ while in the second one $N = 100$. We have to notice that the simulation only affects the value of the $cost = u + 10w$ and the $balance = var(u_n)$, while the *loyalty* remains the same. We have also to notice that even thought the simulations does not effect the *loyalty* the perturbation is centred only on it, as explained in section 3.2.3. In this way we work in different ways on the three different objective functions we are considering.
Time limit is always set at 25 minutes so we can compare the results with the ones in previous chapters.

5.4.1 Results

Results obtained using Multiobjective Simulated Iterated Local Search are shown. For each instance the average value of *Costs*, *Loyalty* and *Balance*, between all the solutions in the pool of approximated Pareto optimal solutions, are presented. In addition also the standard deviation for the values of the objective functions in the pool of approximate Pareto optimal solutions are shown.

In the last table we compare obtained solutions after MoILS and the more realistic MoSimILS. Solutions are different both because the random parts in the algorithm and the simulation. The important is to see if the solutions maintain more or less the same values of objective functions. It can be seen that, in almost all the instances, *cost* and *balance* are a little bit increased, while the *loyalty* is a little bit decreased. This means that with MoSimILS values are more realistic but the algorithm does not deteriorate the solutions.
Percentage variation is evaluated as in chapter 2 as:

$$var_X = \frac{X_{MoSimILS} - X_{MoILS}}{X_{MoSimILS}}$$

and then multiplied for 100, where X_{MoILS} and $X_{MoSimILS}$ are the values found respectively using MoILS and MoSimILS algorithm.
As in previous tables R is the total number of requests during the week and P the number of different patients that have to be visited.

Also two figures, graphically representing the pool of approximate Pareto optimal solutions, are shown for two different instances.
In the last figure pool of approximate Pareto optimal solutions are shown, both for MoILS and MoSimILS with different colours, for two different instances. It can be seen that *costs* and *balance* are higher in most solutions obtained using MoSimILS, while the *loyalty* does not seem significantly different.

Mean values of objective functions								
name	R	P	u+10w	*sd*	l	*sd*	b	*sd*
m540_s13	719	228	413.0	7.5	123.1	6.0	181.8	60.6
m540_s19	731	219	435.6	15.0	132.5	5.9	304.4	103.9
m540_s11	732	220	407.5	8.8	120.5	6.2	227.9	65.2
m550_s13	733	229	440.6	9.0	145.0	4.1	103.4	22.7
m540_s18	737	219	451.4	12.7	153.5	8.6	86.0	19.3
m540_s14	737	223	417.5	0.7	135.8	10.7	44.8	13.9
m540_s15	740	218	418.8	9.3	141.8	7.9	261.0	62.1
m540_s16	754	215	434.3	14.5	146.6	16.5	274.3	82.2
m550_s19	742	219	442.2	4.8	132.0	4.8	503.5	131.6
m540_s10	743	238	426.5	9.9	141.8	4.5	59.4	15.2
m560_s13	747	228	456.7	1.8	146.6	6.4	252.5	57.0
m540_s17	747	237	438.9	5.6	129.2	3.6	110.1	25.8
m550_s14	750	225	449.0	8.0	137.2	10.6	132.6	32.4
m550_s11	751	222	429.4	3.6	130.5	31.8	93.4	27.4
m550_s15	752	218	443.9	11.9	141.5	5.3	226.4	64.8
m560_s19	753	222	457.5	6.8	137.6	6.2	176.3	51.9
m550_s16	754	215	455.7	17.6	146.3	7.5	192.0	54.4
m550_s17	757	241	437.7	7.6	139.0	7.7	184.6	46.4
m560_s15	762	218	453.0	6.5	149.4	2.9	184.1	28.3
m560_s14	763	228	467.6	11.3	130.1	11.0	207.8	62.1
m570_s13	763	231	471.5	3.4	150.8	6.0	114.0	25.8
m560_s16	765	216	475.4	3.7	150.5	8.7	321.3	86.3
m560_s11	765	223	427.5	8.1	127.7	5.4	111.5	23.5
m570_s15	771	221	479.9	2.4	154.2	11.8	162.9	43.9
m540_s12	771	224	476.6	7.0	161.0	7.2	86.2	19.6
m560_s17	773	217	478.6	6.3	156.3	3.5	413.1	111.5
m570_s19	775	222	480.7	5.6	151.6	5.2	168.0	41.9
m560_s18	776	221	500.9	9.7	157.0	4.9	188.2	29.6
m570_s16	777	216	476.3	7.0	138.0	4.9	279.9	77.7
m570_s11	778	224	463.4	5.9	123.7	3.3	121.6	38.6
m570_s14	781	229	482.5	1.7	147.8	3.1	127.2	26.6
m550_s18	783	218	476.6	4.5	170.4	20.5	146.0	27.7
m550_s12	785	226	489.2	0.7	180.9	22.0	145.7	30.5
m580_s15	787	225	493.5	1.6	149.7	4.1	267.0	66.3
m570_s17	788	220	493.8	0.6	177.5	3.5	114.6	24.6
m580_s16	788	220	478.9	0.5	146.6	4.2	273.7	58.6
m580_s19	789	222	493.8	5.8	145.1	5.5	227.8	56.6

m580_s11	789	227	469.1	7.6	135.2	7.5	138.4	29.0
m560_s10	789	240	500.3	6.3	173.5	24.4	45.7	8.4
m570_s18	790	225	507.4	8.9	170.8	4.6	126.4	23.3
m580_s13	790	236	479.4	8.6	144.6	6.7	95.3	14.8
m580_s14	794	232	475.9	2.6	143.1	3.9	106.1	20.7
m550_s10	795	220	520.1	10.7	182.8	10.9	46.4	4.1
m590_s15	797	229	517.0	53.5	163.8	19.4	83.5	18.7
m560_s12	800	228	494.5	1.2	193.0	23.8	122.1	20.4
m590_s11	800	228	491.7	6.0	137.3	14.0	134.6	32.6
m570_s10	801	241	489.8	2.4	155.5	27.3	56.9	8.8
m590_s19	803	224	513.3	5.4	155.7	9.6	354.5	83.9
m590_s16	804	223	500.4	8.6	170.3	27.4	93.0	15.6
m580_s17	806	223	510.0	1.6	178.7	19.6	257.5	49.6
m590_s14	807	234	512.6	4.8	142.1	6.3	142.8	27.9
m590_s13	808	236	509.8	3.0	163.0	23.4	159.5	24.8
m580_s18	811	227	585.3	130.9	178.0	21.1	215.8	42.9
m570_s12	812	231	522.1	4.6	194.8	7.2	260.6	59.1
m580_s10	814	225	491.2	2.9	156.5	7.4	124.2	21.6
m590_s18	822	229	647.5	92.5	174.0	8.2	221.0	41.9
m580_s12	825	232	525.9	10.0	196.1	10.5	224.3	27.8
m590_s10	834	223	522.5	2.1	160.7	7.2	196.3	21.0
m590_s12	841	233	541.3	3.7	189.3	4.5	233.1	32.9
m590_s17	848	228	535.8	2.7	189.7	4.2	318.0	55.6

Table 5.2: Average of the results in pool of approximate Pareto optimal solutions with MoSimILS

Percent variation					
name	R	P	u+10w	l	b
m540_s13	719	228	4.6	-6.2	52.2
m540_s19	731	219	7.8	-9.7	-8.2
m540_s11	732	220	7.5	-14.35	90.9
m550_s13	733	229	10.25	-4.1	69.3
m540_s18	737	219	13.1	-2.5	88.3
m540_s14	737	223	6.1	3.6	-54.4
m540_s15	740	218	6.9	1.2	60.8
m540_s16	754	215	8.2	-1.9	57.6
m550_s19	742	219	9.2	-11.9	82.8

m540_s10	743	238	9.2	-9.0	41.9
m560_s13	747	228	10.7	-8.2	47.5
m540_s17	747	237	11.0	-11.9	48.5
m550_s14	750	225	9.2	6.7	-23.0
m550_s11	751	222	9.6	-3.2	55.8
m550_s15	752	218	8.5	-0.6	23.3
m560_s19	753	222	11.1	-1.2	36.3
m550_s16	754	215	12.1	3.1	46.8
m550_s17	757	241	8.7	-3.5	40.4
m560_s15	762	218	9.6	-3.7	55.8
m560_s14	763	228	16.7	-1.4	87.6
m570_s13	763	231	13.2	0.9	39.9
m560_s16	765	216	13.6	-5.3	46.3
m560_s11	765	223	7.7	-8.1	44.2
m570_s15	771	221	15.7	-4.1	82.8
m540_s12	771	224	13.3	5.0	33.7
m560_s17	773	217	13.8	-8.8	52.7
m570_s19	775	222	13.7	-0.4	8.6
m560_s18	776	221	14.9	-3.0	60.5
m570_s16	777	216	14.6	-3.8	59.5
m570_s11	778	224	14.0	-3.0	73.5
m570_s14	781	229	13.6	-1.9	20.7
m550_s18	783	218	13.2	0.9	17.9
m550_s12	785	226	13.4	11.2	23.1
m580_s15	787	225	15.3	0.03	61.6
m570_s17	788	220	13.7	3.4	-139
m580_s16	788	220	10.5	0.4	19.6
m580_s19	789	222	12.6	1.5	16.9
m580_s11	789	227	13.1	-2.4	33.3
m560_s10	789	240	15.6	3.9	-95
m570_s18	790	225	7.9	-1.4	32.0
m580_s13	790	236	11.1	-1.5	-22.8
m580_s14	794	232	11.3	1.8	-24.2
m550_s10	795	220	16.7	7.7	-294
m590_s15	797	229	12.3	-0.1	-270
m560_s12	800	228	14.2	-3.1	33.3
m590_s11	800	228	14.2	2.2	-29.2
m570_s10	801	241	13.2	-1.0	-49.2
m590_s19	803	224	15.3	-1.6	18.7
m590_s16	804	223	15.2	0.8	-14.9

m580_s17	806	223	16.6	-1.8	60.7
m590_s14	807	234	14.6	0.7	1.7
m590_s13	808	236	15.6	-1.6	19.8
m580_s18	811	227	10.9	2.0	29.3
m570_s12	812	231	14.9	-2.0	34.1
m580_s10	814	225	12.5	6.9	6.5
m590_s18	822	229	6.2	-0.4	-10.0
m580_s12	825	232	16.4	2.0	45.2
m590_s10	834	223	13.3	4.4	-52.9
m590_s12	841	233	12.1	-7.0	39.5
m590_s17	848	228	17.8	3.3	53.9

Table 5.4: Percent variation between MoILS and MoSimILS results

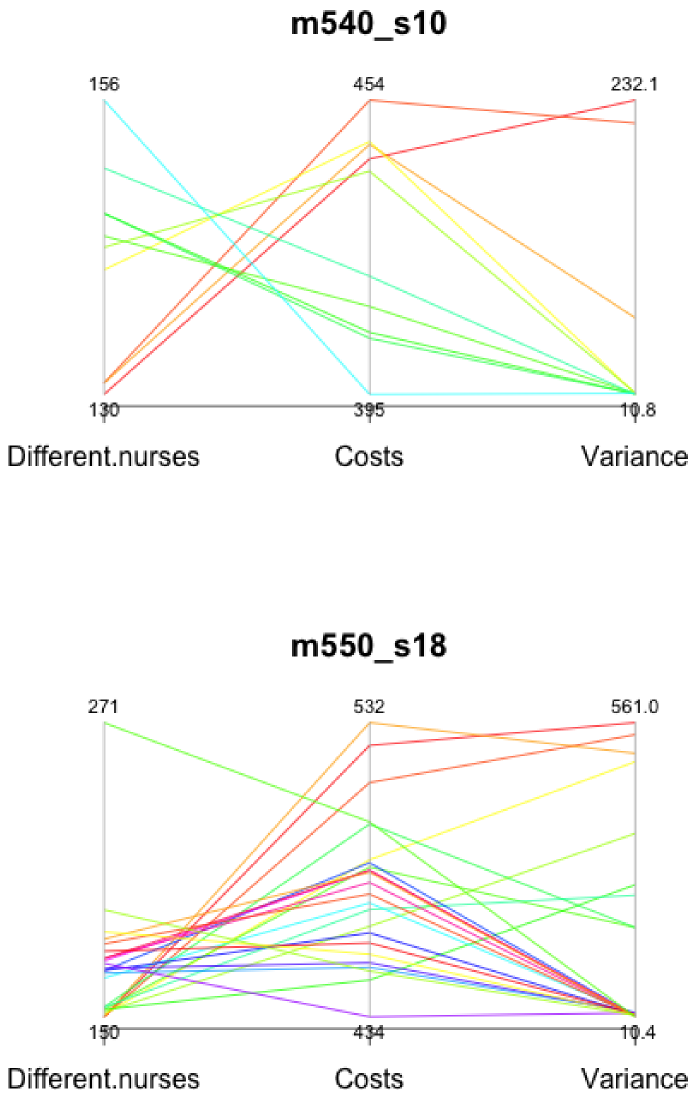

Figure 5.1: Pool of approximate Pareto optimal solutions of two different instances in parallel coordinates using MoSimILS

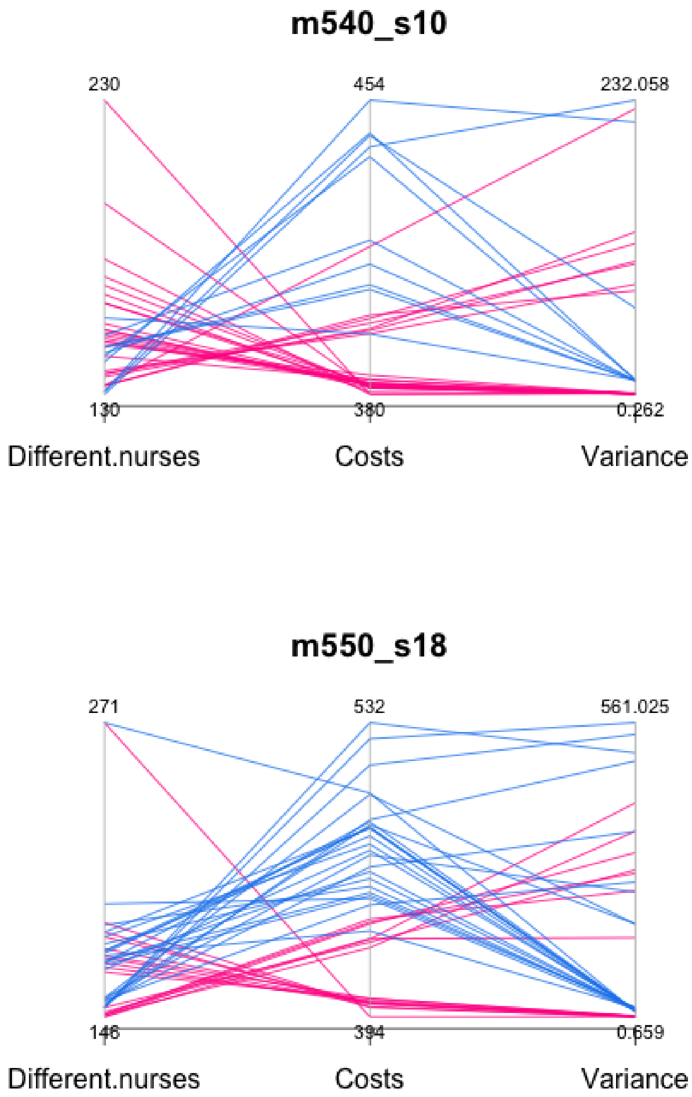

Figure 5.2: Pool of approximate Pareto optimal solutions of two different instances in parallel coordinates using MoILS (pink) and MoSimILS (blue)

Chapter 6

Conclusions and Future Research

Home Health Care service consists in medical and para medical services performed, by medical stuff, to patients at home. The considered problem in this thesis consists in a weekly request of services by a set of patients. Each service is described by three variables: time to perform it, frequency in a week and days that must pass between two repetitions of the same service. Services are performed by a set of nurses.
The aim is to solve the problem without simplifying it too much and maintaining it closer as possible to real life situations.
In the first part of this work a single objective function of HHCP is considered as the summation between costs, in terms of needed time, and loyalty, as the number of different nurses visiting the same patient. Then to make the solution more realistic the problem is considered as multiobjective introducing also the balance of workload between nurses. In the last chapter stochastic service times are introduced in order to be closer to reality.

A mathematical model for HHCP is first developed to explain better the problem. As the problem is NP-hard a metaheuristic method is the best approach for its resolution.
First an Iterated Local Search algorithm is applied. The initial solution method is based on the Clarke and Wright algorithm, then a Local Search is applied to improve the solutions. This Local Search is based on a combination of different operators with the aim of minimizing costs and maximize the loyalty, i.e. minimize the different number of nurses visiting the same patients. The perturbation consists in removing services from 20% of patients, ordered by increasing loyalty, from the obtained tours, and putting them back in the tours where it costs less. With these three parts the ILS algorithm is then built.
Considering HHCP as multiobjective problem the ILS is modified introducing the concept of Pareto optimal solutions, building a new algorithm: MoILS. The bal-

ance of workload between nurses is also introduced as a new objective function, in terms of variance between the average workload for each nurse in a week. To consider three different objective functions, at each iteration of ILS algorithm, a pool of approximate Pareto optimal solutions is updated. The pool is then presented and best solution should be chosen by decision makers.
In the last chapter a new scheme for stochastic and multiobejctive problems is developed. MoILS is then modified using the idea of SimILS algorithm, the new algorithm is called: Multiobejctive Simulated Iterated Local Search, MoSimILS. A simulation is then introduced to service times, making them from deterministic to stochastic parameters and making the problem closer to reality.
The developed algorithms can be applied in reality to obtain a weekly plan for nurses. Knowing the weekly request it can be decided the scheduling part for nurses and also the routing part to obtain the daily tours for each nurse.

All these algorithms are then applied to 60 instances generated, in previous works, from real data furnished by Ferrara hospital. As we can see ILS algorithm obtains good results comparing them to results obtained applying ALNS algorithm on the same instances.
MOILS makes us able to present a set of solutions, that is the approximate pool of Pareto optimal ones, without deteriorating the single values of the objective function, comparing them to the ones obtained with ILS method. The presented set of solutions can be presented to the decision makers that can choose the most appropriate solution considering their preferences.
Introducing uncertainty we can present, using the pool of approximate Pareto optimal solutions, more realistic results.

In future similar HHCPs can be studied, as for example considering also drivers to transport the nurses, in order to not loose time finding parking. More real data can be also obtained from other regions and the developed methods can be applied.
It has to be remembered that scheduling problems for an hospital that is delivering home health care services are the same problems that can appear in other situations, such as in a cleaning company. This means that all the algorithms developed in this thesis can be also applied in other different contexts.
Finally, to extend the developed work, MoSimILS can be applied to solve different multiobjective and stochastic problems. This allows to not simplify too much the problems deteriorating the quality of the results in terms of veracity.

Bibliography

- BACHOUCH, R.B., A. GUINET, S. HAJIRI-GABOUJ, (2011). *A Decision-Making Tool for Home Health Care Nurses' Planning*. In: Supply Chain Forum, Vol. 12.1, pp 14-20.

- BAUM E. B., (1986). *Iterated descent: A better algorithm for local search in com- binatorial optimization problems*. Technical report, Caltech, Pasadena, CA. manuscript.

- BÄUMELT, Z. et al. *Integrated Veichle Routing and Rostering for the Home Health Care Services*. S.l.

- BEECH, A.J., (2011). *Market-based demand forecasting promotes informed strategic financial planning*. In: Healthcare Financial Management 55, 11, 4656.

- BEGUR, S. V., MILLER, D. M., WEAVER, J. R., (1997). *An integrated spatial decision support system for scheduling and routing home health care nurses*. In: Institute of Operations Research and the Management Science, 27(4), 35-48.

- BELLANTI, F., G.CARELLO, F. DELLA CROCE, R. TADEI, (2004). *A greedy-based neighborhood search approach to a nurse rostering problem*. In *European Journal of Operational Research* 153, pp. 28-40.

- BERTELS, S. and T. FAHLE, (2006). *A hybrid setup for a hybrid scenario: combining heuristics for the home health care problem*. In *Computers & Operations Research* 33.10, pp. 2866-2890.

- BOCCAFOLI M., (2012). *Assistenza sanitaria a domicilio: problemi multi-obiettivo d'instradamento di veicoli con bilanciamento di carico e fidelizzazione paziente-infermiera*. Phd thesis Mathematical and Computer Science department, University of Ferrara.

- BORSANI, V., A. MATTA, G. BESCHI and F. SOMMARUGA, (2006). *A Home Care Scheduling Model For Human Resources*. In: *2006 International Conference on Service Systems and Service Management*. Vol. 1, pp. 449-454.

- Boussaïd, I., J. Lepagnot and P. Siarry, (2013).*A survey on optimization metaheuristics*. In: *Information Science: an International Journal*, vol.237, pp. 82-117.

- Burke., E.K., P. De Causmaecher, G. Vanden Berghe and H. Van Landenghem, (2004). *The State of the Art of Nurse Rostering*. In: *Journal of Scheduling* 7 (6),pp. 441-499.

- Burke, E.K., T. Curtois, G. Post, R. Qu and B. Veltman, (2008). *A hybrid heuristic ordering and variable neighbourhood search for the nurse rostering problem*. In: European Journal of Operational Research 188.2, pp. 330-341.

- Caramia, M. and P. Dell'Olmo, (2008). In: Multi-objective Management in Freight Logistics Increasing Capacity, Service Level and Safety with Optimization Algorithms, pp. 11-36.

- Cattafi, M., R. Herrero, M. Gavanelli, M. Nonato, F. Malucelli and J.J. Ramos, (2012). *Improving Quality and Efficiency in Home Health Care: an application of Constraint Logic Programming for the Ferrara NHS unit*. In: *Technical Communications of the 28th International Conference on Logic Programming* (ICLP'12).

- Cheng. E., and J.L. Rich, (1998). *A Home Health Care Routing and Scheduling Problem*. Technical Report TR98-04. Houston, TX.

- Clarke, G. and J.W. Wright, (1964). *Scheduling of Vehicles from a Central Deopt to a Number of Delivery Points*. In: *Operations Research* 12.4, pp.568-581.

- Cordeau J.-F., Gendreau M., Laporte G., Potvin J.-Y., Semet F., (2002). *A guide to vehicle routing heuristics*. J. Oper. Res. Soc. 53:512522 CrossRef.

- Cote, M.J., Tucker, S.L., (2001). *Four methodologies to improve healthcare demand forecasting*. In: *Healthcare Financial Management* 55, 5, 5458.

- Daskin, M.S., Dean, L.K., (2004). *Location of health care facilities*. In: Brandeau, M.L., Sainfort, F., Pierskalla, W.P. (eds) *Operations Research and Health Care*. *A Handbook of Methods and Applications*. Kluwers International Series, Dordrecht, pp. 4376.

- Ehrgott, M. and X. Gandibleux, (2000). *A survey and annotated bibliography of multiobjective combinatorial optimization*. In: OR-Spektrum, vol.22, pp. 424-460.

- EVEBORN, P., P. FLISBERG, M. RÖNNQVIST, (2006). *Laps Care an operational system for staff planning of home care*. In :*European Journal of Operational Research* 171.3, pp.962-976.

- FINARELLI JR, H.J., JOHNSON, T., (2004). *Effective demand forecasting in 9 steps*. In: *Healthcare Finance Management* 58, 11, 5258.

- GEIGER,M.J., (2008). *Proposition of Interactive Pareto Iterated Local Search Procedure - Elements and Initial Experiments*. In: The Fourth International Conference on Evolutionary Multi-Criterion Optimization: Late Breaking Papers, Matsushima, Japan, March 2007, pp. 19-23.

- GEIGER, M.J., (2006). *Foundations of the Pareto Iterated Local Search Metaheuristic*. In: Proceedings of the 18th international conference on multiple criteria decision makingMCDM 2006, Crete, Greece.

- HODGSON, M.J., LAPORTE, G., SEMET, F., (1998). *A covering tour model for planning mobile health care facilities in Suhum District, Ghana*. In: Journal of Regional Science 38, 4, 621638.

- JOHNSON, S., (1990). *Local optimization and the travelling salesman problem*. In: Proceedings of the 17th Colloquium on Automata, Languages, and Programming, volume 443 of LNCS, pages 446461. Springer Verlag, Berlin.

- JOHNSON, D.S., AND L. A. MCGEOCH, (1997). *The travelling salesman problem: A case study in local optimization*. In E.H.L. Aarts and J.K. Lenstra, edi- tors, Local Search in Combinatorial Optimization, pages 215310. John Wiley and Sons, Chichester, England.

- JONES, S.S., THOMAS, A., EVANS, R.S., WELCH, S.J., HAUG, P.J., SNOW, G.L., (2008). *Forecasting daily patient volumes in the emergency department*. In: *Academic Emergency Medicine* 15, 2, 159170.

- LANZARONE, E., A. MATTA AND G. SCACCABAROZZI, (2009). *A patient stochastic model to support human resource planning in home care*. In: Prod Plan Control 2010; 21(1):3-25; ISSN 0953-7287.

- LOURENÇO, H.R., O.C. MARTIN, T. STÜZLE, (2003). *Iterated Local Search*. In: *Handbook on MetaHeuristics*, ser. International Series in Operations Research & Management Science, F. Glover and G. A. Kochenberger, Eds. Boston, Dordrecht, London: Kluwer Academic Publishers, 2003, vol. 57, ch. 11, pp. 321353.

- LOURENÇO, H.R., O.C. MARTIN, T. STÜZLE, (2010). *Iterated Local Search: Framework and applications*. In: *Handbook on MetaHeuristics*, ser. International Series in Operations Research & Management Science, F. Glover and G. A. Kochenberger, Eds. M. Gendreau, Y. Potvin, second ed. Springer, ch. 12, pp. 363-397.

- LOURENÇO, H.R. A. GRASAS, A.A. JUAN, (2014). *SimILS: a simulation-based extansion of the iterated local search metaheuristic for stochastic combinatorial optimization*. In: Journal of Simulation advance online publication.

- MARTIN, O., S. W. OTTO, AND E. W. FELTEN, (1991). *Large-step Markov chains for the traveling salesman problem*. Complex Systems, 5(3):299326.

- MARTIN, O., S. W. OTTO, AND E. W. FELTEN, (1992). *Large-step Markov chains for the TSP incorporating local search heuristics*. Operations Research Letters, 11:219224.

- MURAWSKI, L., CHURCH, R.L., (2009). *Improving accessibility to rural health services: The maximal covering network improvement problem*. In: Socio-Economic Planning Sciences 43, 2, 102110.

- RAHMAN, S., SMITH, D.K., (1999). *Deployment of rural health facilities in a developing country*. In: Journal of the Operational Research Society 50, 9, 892902.

- RAHMAN, S., SMITH, D.K., (2000). *Use of location-allocation models in health service development planning in developing nations*. In: European Journal of Operational Research 123, 437452.

- RAIS, A. AND A. VIANA, (2011). *Operations Research in Health Care: a survey*. In: International Transaction in Operational Research, vol.18, pp.1-31.

- . SMITH-DANIELS, V.L., SCWEIKHART, S.B., SMITH-DANIELS, D.E., (1988). *Capacity management in health care services: Review and future research directions*. In: Decision Sciences 19, 889919.

- SMITH, H.K., HARPER, P.R., POTTS, C.N., THYLE, A., (2009). *Planning sustainable community health schemes in rural areas of developing countries*.In: European Journal of Operational Research 193, 3, 768777.