Matheuristic optimization for robust home health care services

Thi Viet Ly Nguyen [1], Nihat Engin Toklu [1] and Roberto Montemanni [1]
[1] Dalle Molle Institute for Artificial Intelligence (IDSIA – USI/SUPSI), Galleria 2, 6928 Manno, Switzerland
{vietly, engin, roberto}@idsia.ch

Proc. ICAOR 2015
Vienna, Austria

Keywords:
Genetic algorithm
Home health care
Integer programming
Metaheuristics

Abstract

In modern societies, home health care services are becoming increasingly important. Having optimized solutions on operational issues can play a potential role in offering patients high quality medical service as well as taking in better regard, the needs of care providers. An important issue to consider is the uncertainty on the problem data. In more details, an optimal solution that was obtained under the assumption that the collected problem data are accurate, can turn out to be infeasible when it is implemented in the reality, because the data encountered in the reality might differ from the assumed data in the optimization model. In this paper, we consider the uncertainty on the availability of the nurses (e.g. they might call sick on short notice), and we follow a robust optimization approach to handle the case when nurses are unexpectedly unable to operate. A matheuristic method based on a constructive heuristic combined with a genetic algorithm and mathematical programming is proposed to provide a near-optimal solution both in terms of nurse-patient assignment and nurse scheduling and routing.

Introduction

Medical treatment services that take place at the private places of the locals is called home health care (HHC). In the context of improving life expectancy, HHC services are becoming more and more important to locals. HHC becomes an important alternative to the traditional model of a hospital to maintain the best possible clinical conditions for patients who are discharged from hospital but have not been completely recovered yet, or for locals who are in need of physical support or medical treatment (Caffrey *et al*, 2011). A concise review in Nguyen & Montemanni (to appear) shows us that making decisions on nurse-patient assignment and nurse scheduling and routing are the most important tasks in HHC services planning, and during the last decade, these decisions have been considered in some recent publications in the field of HHC optimization. These studies have inspired us to propose an HHC optimization approach which handles nurse-patient assignment and nurse routing decisions by using a monolithic mixed integer linear programming model, so that all these decisions are made in a single phase and an efficient solution for a HHC service is found in our previous studies (Nguyen & Montemanni, to appear). In reality, the presence of the official nurses is uncertain because they might call sick. Ignoring this uncertainty has a noticeable influence on the operation of the HHC system because it might cause shortage of the current workforce. Therefore, it is worth finding a solution that is practical under different realizations of the perturbations in the availability of nurses. In this study, we consider the uncertainty in nurse availability. When an official nurse is missing, uncovered patients have to be visited by extra nurses who are always available on the market and are more costly to hire.

For solving the uncertainty-unaware version of the problem (to be described in details in the next section), integer linear programming models were proposed in Nguyen & Montemanni (to appear). The experiments presented in Nguyen & Montemanni (to appear) demonstrate that the medium and large sized instances cannot be solved by a pure integer programming approach, because of requirements for too much execution time and/or computer memory. The main novel contribution of this paper is a *matheuristic* (hybridization of mathematical programming and metaheuristics; see Maniezzo *et al* (2009)) algorithm that, like in Nguyen & Montemanni (to appear), handles all nurse-patient assignment and nurse routing decisions in a single phase, and also addresses the uncertainty on nurse availability. Because of its heuristic nature, our proposed algorithm can work on large instances. The objective of our proposed matheuristic is to assign nurses to patients, to make a schedule of treatment, and to design the tours of the nurses, in such a way that the solution does not become impractical in terms of cost even in the worst-case scenario. We define the worst-case scenario as the scenario which imposes the maximum additional expenses because of missing nurses. Evaluating a solution under its worst-case scenario leads the optimization algorithm to robust solutions which cost reasonably even under pessimistic situations.

To deal with uncertainties, stochastic optimization approaches which rely on probability distribution information were developed and have long been explored in both application and theory (Birge, J. R. and Louveaux, F. (1997)). However, obtaining reliable information about probability distributions can be very difficult. Therefore, as an alternative, a more recent field called robust optimization emerged. In robust optimization, an uncertain coefficient is represented as a set of possible realizations (usually in the form of an interval), instead of a random variable with a known probability distribution. In the field of robust optimization, an important concept is the conservativeness degree (Ben-Tal & Arkadi, 1998; Bertsimas & Sim, 2003). The conservativeness degree represents how much robustness we want in our solution. The ability to configure the conservativeness degree is important, because it allows the decision maker to run the optimization model with various conservativeness degrees, and therefore obtain and examine alternative solutions with various robustness levels (see, for example, Toklu *et al* (2013)). During the optimization procedure, the conservativeness degree affects how pessimistic our assumptions are about the scenario in which we evaluate solutions. In our approach, we follow the school of robust optimization, and the conservativeness degree represents our assumption on how many official nurses will be missing.

Problem Description

We work on a model which is defined on a complete directed network containing a single central office called depot. Two main elements of the HHC services are the patients and the nurses. We define P as the set of performances of medical treatments which can be handled by the nurses working for the HHC service. We also define *D* as the set of working days during the planning horizon. The patients are represented via a set of patients *J*, in which a patient $j \in J$ needs a performance $p \in P$ on a required day $d \in D$. A set of shifts *K* is given, in which each $k \in K$ is a shift of a working day. In our data, *K* stores three shifts representing morning, afternoon, and evening. Let s_k and e_k be the starting time and ending time of shift $k \in K$, respectively. In addition, the starting time of each performance *p* required by patient *j* on day *d* is constrained by a soft time window $[\underline{st}_{dj},\ \overline{st}_{dj}]$ and a hard time window $[lbv_{dj}, ubv_{dj}]$. Penalty costs $\underline{c}_j$ and $\bar{c}_j$ are paid if the service starts before time $\underline{st}_{dj}$ or after time $\overline{st}_{dj}$, respectively. This penalty-based approach makes the soft time window constraint violations avoided when possible, but accepted if necessary. When it comes to hard time window constraints, however, the model has a strict behavior: the service cannot start before time lbv_{dj} or after time ubv_{dj}. This modeling choice is designed to provide a level of flexibility which is desirable in practice: certain treatments have necessarily to be done within a given time windows, and preferably within a tighter time window. The requirement of a patient is modeled as follows. If performance *p* is required by patient *j* on day *d*, and the soft time window $[\underline{st}_{dj},\ \overline{st}_{dj}]$ belongs to the time interval of shift *k*, then $b_{jp}^{dk} = 1$, otherwise $b_{jp}^{dk} = 0$. The duration of the performance *p* on patient *j*, also called the service time, is denoted by z_j. Finally, the traveling time between patients *i* and *j* is t_{ij}. The cost for every unit of traveling time is denoted by β.

The set of nurses available in the team is denoted by O. Each nurse $n \in O$, called official nurse, is able to cover a set of performances $P_n \subset P$ for which she/he has received a full training. If the official nurse *n* is able to operate the performance *p*, then $f_{np} = 1$, otherwise $f_{np} = 0$. If nurse *n* is available on shift *k* on day *d* then $a_n^{dk} = 1$, otherwise $a_n^{dk} = 0$. Each nurse is available on only one shift *k* per day. The starting time of her/his working day is the starting time of her/his shift. We allow the case that nurse *n* does not need to visit any patient on a working day if there is no demand to handle, even though that nurse is available on that day. Together with the official nurses, we define external nurses E_1 who are available on the market to operate in case the official nurses are not enough to cover the given demand, but with a very high cost δ_e according to the number of kind of treatments that she/he can efficiently operate.

The availability and costs of nurses are modeled as follows. Each day, the initial cost C_0 is paid for her/his whether she/he visits patients or stays at the office. In addition, nurse *n* will get an extra payment for working after the usual working time, *T* hours. The cost for each overtime unit is c_o. A maximum allowed value of the working time on a day for a nurse is *L* hours. Nurse *n* can only work on performance *p* required by patient *j* on shift *k* on day *d* if the following three conditions are met: (*i*) nurse *n* is able to operate the performance *p*; (*ii*) she/he is available on shift *k* on day *d*; and finally, (*iii*) the associate hard time window of performance *p* required on day *d* by patient *j* belongs to the time interval of shift *k*. We assume that each hard time window of performance *p* required on day *d* by patient *j* belongs to the time interval of only one shift.

In this study, uncertainty about the effective presence of the official nurses (e.g., they might call sick) is taken into account. In case it is revealed that an official nurse is missing, an external nurse, from the emergency external nurses set E_2, is hired with a hiring cost δ_e to handle the patients which were assigned to the missing nurse, by following the route which was originally planned. We assume that each external nurse is able to efficiently operate any performance. She/he is available to work at any time and operate up to *L* hours per day.

The objective function of the problem is to provide a nurse-patient assignment and nurse scheduling and routing solution which minimizes the total operational cost when handling the uncertainty on the availability of the official nurses.

A Robust Optimization Approach

We now give the details of the robust optimization approach we use for handling the uncertainty on the availability of the nurses. Namely, we follow the principles of robust optimization approach introduced by Bertsimas & Sim (2003). By applying this approach, the manager has ability to configure the level of conservativeness degree in order to make a clever decision in terms of trade-off between the operational cost in the best case and robustness. In our study of HHC services, the conservativeness degree constant Γ, which is a non-negative integer, denotes the budget of missing nurses on a working day. Each realization of the perturbations in the effective presence of the official nurses on a working day is a possible scenario. A non-conservative manager might assume that no official nurse will be missing while making decisions (i.e. $\Gamma = 0$). On the other hand, a more conservative manager might expect that some official nurses will be missing (i.e. $\Gamma > 0$). In this study, we consider every scenario in which up to Γ number of official nurses are missing. It is obvious that the size of set of all possible scenarios becomes extremely large when we slightly increase the value of Γ or $|O|$. There will be a large number of scenarios when there are a lot of nurses. Evaluating a solution under every scenario would make a solution approach computationally expensive. Usually, in robust optimization, we focus only the worst-case scenario from the set of all possible scenarios. This is also how we handle the uncertainty in our current approach.

Finding the worst-case scenario. If an official nurse $o \in O$ is missing, an associated external nurse $e \in E_2$ with a very high hiring cost δ_e must be called to handle unvisited patients. For each official nurse $o \in O$, the potential additional cost λ_o is the subtraction of her/his initial cost C_o from her/his replacement cost Δ_o which is equivalent to the hiring cost of an associated external nurse $\Delta_o = \delta_e$. In our study, we assume that δ_e increases with the number of skills of the external nurse $e \in E_2$. Therefore, if an official nurse $o \in O$ with an assigned route requiring a high number of different skills is missing, the external nurse to be hired to handle the patients of the nurse *o* will be paid a very high cost too. So, we define the criticality of an official nurse as the number of skills required in her/his assigned route. The worst-case scenario in this study is the scenario in which the care provider has to pay the most, therefore, considering the conservativeness degree parameter Γ, the worst-case scenario is the scenario in which Γ number of nurses are missing, and these missing nurses are the most critical ones (i.e. nurses who would cause the highest replacement costs).

Uncertainty-aware solutions evaluation. We evaluate the quality of a solution based on its solution cost which is the sum of the base cost and potential additional cost under the worst case scenario. The base cost is the sum of total initial cost, total travel cost, and total penalty cost over all nurses. The potential additional cost under the worst case scenario is the additional cost that would be encountered if Γ number of most critical nurses would be missing.

Let us define the procedure of calculating the cost of a given solution, in terms of a solution vector χ. Here, $\chi[n,d]$ represents a given route decided for the nurse *n* on day *d*, where *n* can be an official nurse ($n \in O$), or an extra nurse ($n \in E_1$), and $|\chi[n,d]|$ represents the length of that route (i.e. how many locations exist within that route). Also, $\chi[n,d,q]$ represents the *q*-th location visited by the nurse *n* on day *d*. Note that, each nurse begins and ends her/his tour at the office, i.e.: $\chi[n,d,1] = \chi[n,d,|\chi[n,d]|] = 0$. Now let us define the components of the cost function for a given solution χ. The total initial cost of a given solution is presented as follows:

$$\textsc{InitialCost}(\chi) = \sum_{n\in O} \sum_{d\in D} C_n + \sum_{n\in E_1} \sum_{d\in D} \delta_n$$

The total travel cost of a given solution is expressed as follows:

$$\textsc{TravelCost}(\chi) = \sum_{n\in(O\cup E_1)} \sum_{d\in D} \sum_{q\in\{2,3,\dots,|\chi[n,d]|\}} \beta t_{\chi[n,d,q-1],\chi[n,d,q]}$$

In addition to TRAVELCOST, we need to consider the penalty costs that we will pay for the time window violations and overtime. These penalties depend not only on the order of visits stored within the solution χ, but also on the starting time decisions for all the visits. Therefore, at this point, we need to make those decisions, in such a way that these penalties will be minimized. After making these decisions, these minimized penalties are to be added to the solution cost. For this purpose, we first define a linear programming model called MINPENALTY(n,d), which finds the minimum penalty cost for the route of the nurse $n \in (O \cup E_1)$ in day $d \in D$. In the model MINPENALTY(n,d), we define the following variables: st_{ndj} as the starting time at patient *j*; $etwv_{ndj}$ as the early time window violation penalty cost at patient *j* (i.e. penalty cost caused by starting the service earlier than the beginning time of the soft time window); $ltwv_{ndj}$ as the late time window violation penalty cost at patient *j* (i.e. penalty cost caused by starting the service later than the ending time of the soft time window); e_{nd} as the ending time of the tour; O_{nd} as the overtime penalty cost of the tour. By using these variables, we are now ready to formulate MINPENALTY(n,d) as follows:

MinPenalty(n,d): minimize $\sum_{q\in\{2,3,\ldots,|\chi[n,d]|-1\}}\left(etwn_{n,d,\chi[n,d,q]} + ltwv_{n,d,\chi[n,d,q]}\right) + O_{nd}$ (1)

subject to:

$etwv_{n,d,\chi[n,d,q]} \geq \underline{c}_{\chi[n,d,q]}(\underline{st}_{n,\chi[n,d,q]} - st_{n,d,\chi[n,d,q]})$ $\quad \forall q \in \{2,3,\ldots,|\chi[n,d]|-1\}$ (2)

$ltwv_{n,d,\chi[n,d,q]} \geq \bar{c}_{\chi[n,d,q]}(st_{n,d,\chi[n,d,q]} - \overline{st}_{n,\chi[n,d,q]})$ $\quad \forall q \in \{2,3,\ldots,|\chi[n,d]|-1\}$ (3)

$s_{nd} + t_{0,\chi[n,d,2]} \leq st_{n,d,\chi[n,d,2]}$ (4)

$e_{nd} = st_{n,d,\chi[n,d,|\chi[n,d]|-1]} + z_{\chi[n,d,|\chi[n,d]|-1]} + t_{\chi[n,d,|\chi[n,d]|-1],0}$ (5)

$st_{n,d,\chi[n,d,q-1]} + z_{\chi[n,d,q-1]} + t_{\chi[n,d,q-1],\chi[n,d,q]} \leq st_{n,d,\chi[n,d,q]}$ $\quad \forall q \in \{2,3,\ldots,|\chi[n,d]|-1\}$ if $|\chi[n,d]| \geq 4$ (6)

$e_{nd} \leq s_{nd} + L$ (7)

$O_{nd} \geq c_o[(e_{nd} - s_{nd}) - T]$ (8)

$st_{n,d,\chi[n,d,q]}, etwv_{n,d,\chi[n,d,q]}, ltwv_{n,d,\chi[n,d,q]} \geq 0$ $\quad \forall q \in \{2,3,\ldots,|\chi[n,d]|-1\}$ (9)

$e_{nd}, O_{nd} \geq 0$ (10)

where objective function (1) aims to minimize the penalty costs of both early and late time window violations as well as the penalty cost of overtime for the given route $\chi[n,d]$. The penalty costs of early time window violation, late time window violation, and overtime are calculated in constraints (2), (3) and (8) respectively. The starting times of the services at patients are guaranteed to be feasible in constraints (4) and (6). Constraints (5) and (7) together make sure that the total working time of a nurse on a day will not exceed the limit *L*. Constraints (9) and (10) present the variable domains.

Note that MinPenalty(n, d) is formulated with the assumption that there is at least one patient on the given tour. If the tour is empty, we just set all the penalty decision variables (O_{nd}, $etwv_{ndj}$, $ltwv_{ndj}$) to 0. By using the resulting O_{nd}, $etwv_{ndj}$, $ltwv_{ndj}$, we can find the total penalty cost:

$$\text{PenaltyCost}(\chi) = \sum_{n\in(O\cup E_1)} \sum_{d\in D}\left(O_{nd} + \sum_{q\in\{2,3,\ldots,|\chi[n,d]|-1\}}\left(etwv_{n,d,\chi[n,d,q]} + ltwv_{n,d,\chi[n,d,q]}\right)\right)$$

Let us define $\Phi(\chi,\Gamma)$ as a set holding the Γ number of official nurses, these Γ nurses being the most critical ones according to the solution χ. We can now define the potential additional cost under the worst-case scenario as:

$$\text{PotentialAddition}(\chi,\Gamma) = \sum_{n\in\Phi(\chi,\Gamma)}(\Delta_n - C_n)$$

Finally, we define the uncertainty-aware solution cost as:

$$\text{SolutionCost}(\chi,\Gamma) = \text{InitialCost}(\chi) + \text{TravelCost}(\chi) + \text{PenaltyCost}(\chi) + \text{PotentialAddition}(\chi,\Gamma)$$

A Matheuristic Approach

In this study, we develop a matheuristic approach for solving the HHC problem, also considering the uncertainty in nurse availability. In this matheuristic approach, considering the set of days *D*, $|D|$ number of self-adaptive genetic algorithms (GA) (see, for example, Pellerin *et al* (2004)) are executed concurrently, each GA being focused on self-adaptation of its execution parameters (crossover and mutation probabilities) and finding a nurse-patient assignment and routing plan for a different day within *D*. Every iteration of the matheuristic algorithm involves the execution of a single iteration of each GA and the construction of a best-known solution, which is the combination of the best solution of each GA's population.

In a GA focused on $d \in D$, a solution vector stores a list of patients to visit (in specified order) for each nurse *n* within day *d*. The objective of the GA is the minimization of SolutionCost. As explained in Section 3, a component of SolutionCost named PenaltyCost is affected by the visiting times at the patients. Therefore, when evaluating a solution vector based on its SolutionCost, we use the linear programming model MinPenalty(*n*, *d*) embedded into the objective function to evaluate tours and decide the visiting times at patients. The population of a GA at the beginning is prepared by a heuristic initial population generator. A binary linear programming-based cross-over (i.e. construction of new "offspring" solution vectors by using two "parent" solution vectors) operator is also built.

Heuristic initial population generator. For generating an initial solution, for each day, for each nurse $n \in (O \cup E_1)$ (nurses being picked in random order), we construct tours starting from the depot. At each step of the tour generation, considering the current location of the nurse, we compute the set of nearest visitable patients whose time windows clash. If $n \in E_1$, this set will also be restricted to the patients which minimizes the number of skills required by the current nurse. From that set, the next patient to visit is picked randomly. When there is no visitable patient left, the next visit is assigned as the depot, and the tour of the current nurse becomes complete. Repeating this initial solution generation procedure multiple times, we obtain an initial population.

The cross-over operator. Because of the many practical constraints of the HHC problem at hand, our cross-over operator differs from the traditional GA cross-over operators. The goals of our cross-over operator are: (i) to increase the solution diversity; (ii) to achieve offspring solutions with the sum of total potential replacement costs and total extra-nurse-hiring costs are minimized; and (iii) to keep the offspring solutions away from infeasibility as much as possible. To satisfy these goals, we developed a two-stage approach. At the first stage, by using the same technique as in the heuristic initial population generator, a set of possible tours is generated for each nurse, with an additional constraint limiting the number of skills required on a tour, to avoid large potential nurse replacement costs. To these sets, the tours from the parent solutions are also added. In the second stage, a combination of tours is chosen for a new offspring solution with the help of a binary linear programming model. For this model, let us define: R as the set routes; R^j as the set of routes that visit patient j; R_n as the set of routes that can be operated by the nurse $n \in (O \cup E_1)$; ω_{rn}^1 as the weight of replacement cost and ω_{rn}^2 as the weight of extra-nurse-hiring cost, where $r \in R_n$ and $n \in (O \cup E_1)$; σ_r as the cost of route $r \in R$; ζ as the maximum allowed number of extra nurse routes allowed. We also define the decision variable $y_r \in \{0,1\}$: $y_r = 1$ means that the route $r \in R$ is involved in the offspring solution, otherwise $y_r = 0$. Considering day $d \in D$, the model is defined as follows:

$$\text{minimize} \quad \sum_{n\in O}\sum_{r\in R_n}\omega_{rn}^1\sigma_r y_r + \sum_{n\in E_1}\sum_{r\in R_n}\omega_{rn}^2\sigma_r y_r \tag{11}$$

$$\text{subject to} \quad \sum_{r\in R^j} y_r \geq 1 \qquad \forall j \in J \tag{12}$$

$$\sum_{r\in R_n} y_r \leq 1 \qquad \forall n \in (O \cup E_1) \tag{13}$$

$$\sum_{n\in E_1}\sum_{r\in R_n} y_r \leq \zeta \qquad \text{if } |E_1| \geq 1 \tag{14}$$

$$y_r \in \{0,1\} \qquad \forall r \in R \tag{15}$$

where the objective function (11) minimizes the total potential cost on day d of a solution vector $\chi[d]$. Constraints (12) means that each patient is assigned to at least one tour. Constraints (13) mean each nurse can only be in charge of at most one tour. A limit on the allowed number of tours operated by extra nurses is stated in constraints (14). Constraints (15) present domains of variables.

Constraints (12) potentially make the created offspring likely an infeasible solution vector because there may be multiple visits at a patient. To fix the infeasibility of the new solution vector, for each patient visited more than once, we remove visits from the tours which require a larger number of skills or store a larger number of locations to visit.

Experiments

To make sure that this study is close to reality, we modeled the HHC services according to the activity of a non-profit organization called SCuDo (Servizio Cure a Domicilio) <http://www.scudo.ch/>. This organization operates in Canton Ticino, Switzerland and provides HHC services to people who are in need of medical treatments. For obtaining preliminary results, we considered two instances based on real data provided by SCuDo. The instances consider 190 patients requiring total 760 medical treatments in one week, 15 official nurses, 22 types of medical treatment, and three working shifts. The proposed matheuristic was implemented in C++ and the solver GUROBI <http://www.gurobi.com/> 5.5.0 C++ interface was used to optimally solve sub-problems where mathematical models are embedded. All experiments were performed on a computer with Intel Core 2 Duo P9600 @ 2.66GHz with 4GB of RAM. For the GA, some preliminary tests suggested to set the population size to 200 and the number of generations to 10. On each instance, our GA was executed 30 times, and average results are reported.

Testing the GA. In these experiments, we compare our GA against the constructive heuristic we used in the heuristic initial population generator. The goal of these experiments is to see if the generations of our GA indeed improve the solution quality. The population size being 200, over 10 generations, our GA processes 2000 solutions. Therefore, for the comparison, we independently execute the constructive heuristic 2000 times and take its best value. After repeating this test 30 times, the results (with $\Gamma = 1$) in Table 1 are obtained. In the table, over the 30 tests, the minimum ("min"), the maximum ("max"), and the average solution costs found by the GA and by the constructive heuristic are presented. The standard deviations ("stdev") over the 30 tests are also presented. In Table 1, it can be seen that the solutions found by the GA have lower costs. This means that the GA works as expected, and it is not equivalent to an undirected random search. How the best-known solution cost (averaged over 30 tests, with Γ set as 2) decreases with each generation of GA is reported in Figure 1, from which it can be seen that, even few generations are enough to increase the solution quality. Also note that, within the execution times of GA reported in Table 1, it would not be possible to solve these datasets by using pure mathematical programming. This shows the practicality of our GA in terms of execution speed.

Table 1. Comparison between constructive heuristic and GA with Γ = 1

		Constructive heuristic				*GA*			
		min	*max*	*stdev*	*average*	*min*	*max*	*stdev*	*average*
Data 1	cost	22812.60	23111.60	70.90	22995.70	18080.60	18453.70	88.16	18276.50
	Time (sec)	416.79	442.32	5.39	420.98	832.09	903.93	18.87	869.78
Data 2	cost	21128.00	21489.80	70.34	21390.50	17760.30	18166.50	102.16	17965.30
	Time (sec)	400.77	430.41	6.98	405.79	821.35	892.84	21.55	859.05

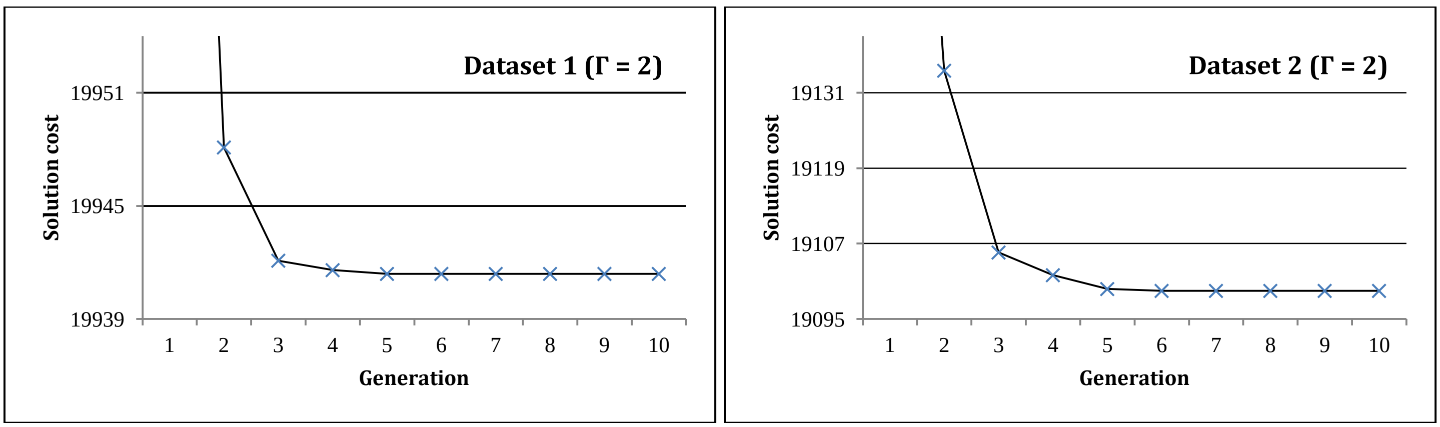

Fig. 1. Average solution cost over 30 executions for each generation

Analysis of the solutions. In these experiments, we run the GA with each considered conservativeness degree setting $\Gamma \in \{0, 1, 2, 4, 8\}$ (i.e. we find solutions optimized for the situation in which Γ number of official nurses are missing). Then, for each obtained solution χ, we calculate $\textsc{SolutionCost}(\chi, \Upsilon)$ for each $\Upsilon \in \{0, 1, 2, 4, 8\}$ (i.e. we calculate what would be the maximum potential cost of an obtained solution χ if ϒ number of official nurses were missing). This shows us, by using our GA, how alternative solutions can be obtained, each with a different protection level against the uncertainty (with different Γ), and how each of these solutions behaves under different scenarios (with different ϒ). This experiment was done over the two datasets. The results averaged over 30 executions of the experiment are shown in Table 2.

Table 2. Costs of alternative solutions under different scenarios

	Dataset 1					*Dataset 2*					
	ϒ=0	*ϒ=1*	*ϒ=2*	*ϒ=4*	*ϒ=8*		*ϒ=0*	*ϒ=1*	*ϒ=2*	*ϒ=4*	*ϒ=8*
Γ=0	16524.6	18315.2	19999.9	23140.6	29023.9	Γ=0	16660.9	17995.9	19188.2	21352.5	25397.2
Γ=1	16548.5	18276.5	19937.8	23071.8	28953.8	Γ=1	16713.6	17965.3	19127.6	21257.0	25279.6
Γ=2	16616.0	18326.0	19941.4	23039.4	28915.4	Γ=2	16738.2	17969.9	19099.5	21221.5	25203.5
Γ=4	16631.1	18341.8	19948.4	23016.4	28873.1	Γ=4	16788.5	18013.8	19132.2	21230.2	25197.2
Γ=8	16688.9	18429.5	20046.9	23092.9	28671.5	Γ=8	16922.9	18123.2	19225.2	21301.2	25143.5

In the table, each row is a separate solution found with a different conservativeness degree Γ, and each column shows the solution's cost under different scenario assumptions, with different ϒ values. In the results, it can be seen that a non-conservative solution ($\Gamma = 0$) has always the lowest cost in the best-case scenario ($\Upsilon = 0$), but under worse scenarios with higher ϒ assumptions, the cost of the non-conservative solution increases more quickly compared to its alternatives. On the other hand, the most conservative solution considered ($\Gamma = 8$), compared to its alternatives, has the highest cost when $\Upsilon = 0$, and the lowest cost when $\Upsilon = 8$. Compromise solutions can also be seen with $\Gamma = 1$, $\Gamma = 2$, and $\Gamma = 4$. A pattern to notice in the results is that, Q being an integer, a solution optimized with $\Gamma = Q$ is the solution which has indeed the lowest cost under the scenario $\Upsilon = Q$. Two exceptions to this are the solution the $\Gamma = 2$ (which looks dominated by the solution $\Gamma = 1$ under $\Upsilon = 2$, considering the averaged costs) in the first dataset, and the solution $\Gamma = 4$ (which looks dominated by the solution $\Gamma = 2$ under $\Upsilon = 4$) in the second dataset. Such exceptions can be seen in a collection of solutions because of the heuristic nature of the GA, and can be easily overcome by the manager by simply going for the more practical one among the solutions.

In general, our experimental results show that our GA can generate different solutions with different robustness levels. By analyzing these solutions, the manager can see the trade-off between the operational cost in the best-case scenario and robustness, and then pick the alternative that is the most practical one.

Conclusion

In this study, we applied robust optimization principles to deal with the uncertainty in nurse availability and proposed a matheuristic that is a MILP based Genetic Algorithm to solve the studied problem within reasonable execution time. Scheduling and routing decisions in the Home Health Care services were addressed in a single phase of the algorithm. Preliminary computational results demonstrated that our approach works properly and can be used to help the manager examine the trade-off between robustness and solution cost when making decisions.

Acknowledgments—Thi Viet Ly Nguyen was supported by the Swiss National Science Foundation through project 200020-140315/1: “Robust Optimization II: applications and theory”. The authors would like to thank Marco Cinus for his useful explanation about the operations of SCuDo and for having provided the real data.

References

Ben-Tal, A., & Arkadi, N. (1998). Robust convex optimization. Mathematics of Operations Research, 23 (4), 769-805.

Bertsimas, D., & Sim, M. (2003). Robust discrete optimization and network flows. Mathematical Programming, 98 (1-3), 49–71.

Birge, J. R. and Louveaux, F. (1997). Introduction to stochastic programming, Springer Verlag.

Caffrey C., Sengupta M., Moss A., Harris-Kojetin L., & Valverde R. (2011). Home health care and discharged hospice care patients: United States, 2000 and 2007. National Center for Health Statistics, 38, 1–27.

Maniezzo, V., Stutzle, T., & Voß, S. (2009). Matheuristics: Hybridizing metaheuristics and mathematical programming. Annals of Information Systems, 10.

Nguyen, T. V. L., & Montemanni, R. (to appear). Mathematical programming models for home health care service optimization. International Journal of Operational Research.

Pellerin, E., Pigeon, L., & Delisle, S. (2004). Self-adaptive parameters in genetic algorithms. In Proceeding of Conference on Data Mining and Knowledge Discovery: Theory, Tools, and Technology VI (pp. 53–64).

Toklu, N. E., Montemanni, R., & Gambardella, L. M. (2013). An ant colony system for the capacitated vehicle routing problem with uncertain travel costs. In Proceedings of the IEEE Symposium on Swarm Intelligence (pp. 32–39).