INTEGRATED VEHICLE ROUTING AND ROSTERING FOR THE HOME HEALTH CARE SERVICES

Zdeněk Bäumelt[1], *Libor Waszniowski*[2],
Přemysl Šůcha[1], *Zdeněk Hanzálek*[1,2]

Abstract: *The paper presents an algorithm for nurse routing and rostering. The objective is to construct routes and the roster for nurses providing services for geographically spread patients. Nurses are located in the hospital and they travel around patients. The total duration of the set of routes is minimized. Routes must satisfy constraints on time windows defining when the visit can be accomplish and the constraints on the duration of the assigned nurse shift. The roster must satisfy constraints on the nurse workload, balanced shift assignment, forbidden shift combination, etc.*

Keywords: *logistics of health services, planning health services/care, staff scheduling, heuristics, metaheuristics, health providers*

1. Introduction

This paper focuses on the personnel scheduling problem in the home health care services (HHCSP). The scheduled persons are nurses providing services to patients at home. The objective is to design routes of nurses and to schedule shifts of nurses to provide sufficient workforce with respect to the subjective requirements on the schedule. Therefore, the HHCSP consists of two sub-problems, the vehicle routing problem with time windows and the nurse scheduling problem (Burke et al., 2004), (Ernst et al., 2004).

The routing part of the problem considers the time and the costs of travelling to patients. Each visit of the patient must occur at the defined time window, the total time of the route must be shorter than the length of the shift assigned to the nurse and all patients must be served. The solution of the routing sub-problem minimizes the cost of the travelling time.

[1] Department of Control Engineering, Faculty of Electrical Engineering,
Czech Technical University in Prague, Czech Republic
[2] Merica Ltd., Czech Republic

The solution of the rostering part assigns shifts to nurses. The lengths and number of the shifts depend on the set of routes planned for each day. The schedule has to satisfy hard constraints defined by the labor code, the collective agreement and the work regulations. The objective is to balance the total workload of the nurses, to avoid isolated days-on/days-off, to balance the nurses' weekend workload etc.

Due to the complexity of the HHCSP the both parts are solved by heuristic approaches. The patients' needs are taken into account as the input of the routing part. Consequently, the optimal routes are covered by the available nurses with respect to the scheduling part of the HHCSP. The paper shows how the both parts are integrated together.

1.1. Related Works

This paper deals with the personnel scheduling problem in the home health care services (HHCSP). There are a lot of papers focused on the standalone Nurse Rostering Problems or on the Vehicle Routing Problems, while the number of papers that handle the combination of these problems is very small.

A MIP mathematical model of HHCSP is introduced in (Cheng, 1998). This mathematical model is used for the definition of the problem only, while the solution is obtained by a two phase heuristic algorithm that iteratively improves the routes assignment.

The approach depicted in (Everborn, 2006) solves the rostering part and the routing part together by the repeated matching algorithm, where the visits are assigned directly to the nurses with respect to given hard and soft constraints.

Another approach for this problem is described in (Bertels, 2006), where the authors applied the combination of the constraint programming and metaheuristics on the problem. This approach was tested on generated instances of dozens of nurses with hundreds of visits.

1.2. Paper Outline

The rest of the paper is organized as follows. Section 2 provides formal definition of the problem. Section 3 presents the algorithm for nurse routing. Sections 4 and 5 describe algorithms for solving routing and rostering sub-problems. Section 6 concludes the paper.

2. Problem Statement

The problem of the nurse routing and rostering (nurse routing for short) is the combinatorial problem of the routes planning and the roster construction for nurses providing services for geographically spread clients. Nurses are located in the hospital and they travel around visits. Each nurse is supposed to be equipped by a vehicle from a homogenous fleet. The total duration of the set of routes is optimized. Routes must satisfy constraints on time windows defining when the visit can be accomplish and the constraints on the duration of the assigned nurse shift. The roster and the assignment of a concrete nurse to the route must satisfy constraints on the nurse workload, balanced shift assignment, forbidden shift combination, etc.

The problem of nurse routing can be defined formally as n-tuple

$$\langle h, \boldsymbol{V}, \boldsymbol{N}, TT, O, C, Dur, Rel, Dead, \boldsymbol{Days}, NR \rangle, \tag{1}$$

where:

- h is the hospital
- $\boldsymbol{V}$ is the set of visits
- $\boldsymbol{N}$ is the set of nurses
- $TT_{i,j}$ is the travelling time between visits (or hospital) i and j
- $O_{i,w}$ is the opening time of visit i on day of week w
- $C_{i,w}$ is the closing times of visit i on day of week w
- Dur_i is the duration of visit i
- Rel_i is the release date of visit i
- $Dead_i$ is the deadline of visits i
- ***Days*** is the set of days for which the roster is designed (planning horizon)
- NR represents the initial roster containing e.g. the holidays of nurses

The solution of the nurse routing problem is the set of opened routes $\boldsymbol{OR}$ and the roster NR. For each route $i \in \boldsymbol{OR}$ we define:

- D_i is the date of the route i
- N_i is the nurse assigned to the route i
- R_i is the trip from the hospital to a sequence of n visits and back to the hospital. The route R_i is represented as (0, 1, 2, ..., j, ..., n+1) where 0 and n+1 both refer to the hospital. $R_{i,j}$ is the visit at the j-th position of the route i.
- A_i is the sequence of arrival times of the route i. The time $A_{i,j}$ refers to the arrival time to the visit at the j-th position of the route i.

Constraints of the HHCSP are divided into two groups. The first group is stated as hard constraints that have to always be fulfilled. On the other hand, soft constraints can be violated, but their non-fulfillment is penalized in the cost function. All constraints in the routing part of the HHCSP are hard. The cost function optimized by routing is

$$Z_{VRP}(\boldsymbol{OR}) = \sum_{i \in R} (A_{i,n+1} - A_{i,0}). \tag{2}$$

The rostering part of the HHCSP contains following hard constrains:

- required number of nurses for each shift type

- nurses' requests consideration
- one shift assignment per day
- forbidden shift combinations

The constraints in the list below are soft constraints:
- balanced nurses' workload
- nurses' weekend workload balance
- avoiding isolated days-on/days-off

Their violations are projected to the cost function of the rostering part

$$Z_{NRP}(NR) = \sum_{\forall i} w_i \cdot p_i \tag{3}$$

where *w* is a vector of the weights of the particular soft constraints and *p* is a vector of the penalizations of violated soft constraints. Finally, the objective of the algorithm is to minimize the following cost function

$$Z(\boldsymbol{OR}, NR) = \alpha \cdot Z_{VRP}(\boldsymbol{OR}) + \beta \cdot Z_{NRP}(NR). \tag{4}$$

3. Nurse Routing

The nurse routing problem is solved by interaction of routing algorithm (referred as Router in this paper) and rostering algorithm (referred as Roster in this paper).

The Router designs the sets of routes satisfying restrictions of the Roster. If these opened routes are not able to cover all visits, Router prolongs the opened routes (while violating constraints of the Roster) or designs additional routes (unfeasible from the Roster point of view) needed to cover unassigned visits. For this purpose, the Router can use the extra capacity only, specified by the Roster on each day of the planning horizon.

The Roster defines for each day of the planning horizon the possible number of routes that can be opened and can be assigned to a nurse. Moreover, it defines the earliest allowed beginning and the latest allowed end of each opened route. Furthermore, the Roster can split unsuitable routes to the surrounding days.

The process is repeated until the stop condition is met. Formally is the algorithms defined by Algorithm 1. We define the following additional variables providing interface between the Roster and the Router:

- $B_{r,d}$ is the earliest allowed beginning of the route $r \in \boldsymbol{OR}$ on the day $d \in \boldsymbol{Days}$
- $E_{r,d}$ is the latest allowed end of the route $r \in \boldsymbol{OR}$ on the day $d \in \boldsymbol{Days}$
- NC_d is the extra capacity on the day $d \in \boldsymbol{Days}$ (in person hours) that can be used to prolong any opened route or to open a new route

Algorithm 1. Nurse Routing

```
Input: (h, V, N, TT, O, C, Dur, Rel, Dead, Days, NR)
Output: (OR*, D*, B*, E*, N*, R*, A*, NR*)

⟨OR, D, B, E, NC⟩ = CreateInitialOpenedRoutes(N, Days, NR)
OR* = OR
NR* = NR
do
    ⟨OR', D', B', E', R', A'⟩ = DesignRoutes(
                              h, V, TT, O, C, Dur, Rel, Dead, OR, D, B, E, NC)
    ⟨OR, D, B, E, N, NR, NC⟩ = DesignRoster(N, Days, OR', D', B', E', NR)
    //update the best known solution
    if Z(OR', NR) < Z(OR*, NR*)
        OR* = OR'; D* = D'; B* = B'; E* = E'; R* = R'; A* = A'; N* = N
        NR* = NR
    end if
while   max number of iteration not reached AND
        criterion improvement during the last 10 iterations is over threshold
```

Function *CreateInitialOpenedRoutes* makes the initial set of routes according to the initial roster *NR*. Function *DesignRoutes* is described in Section 4. Function *DesignRoster* is described in Section 5. Function *Z* is the cost function defined by equation (4).

4. Routing

The routing sub-problem is the vehicle routing problem with time windows. Notice, however, that the capacity constraint is not considered. The algorithm is formally defined at Algorithm 2. The inputs of the routing algorithm are sets of opened routes for each day of the planning horizon and time constraints on these routes. The router inserts each visit to one route from the set of opened routes on a day between its release date and deadline (function *InsertToRoutes* defined by the Algorithm 3). If there remains a visit that cannot be inserted to an opened route, the Router tries to cover the uninserted visits by allowing overtime (function *InsertToRoutesWithOverTime*). The total overtime of all routes for day *d*, however, can not exceed the allowed extra capacity NC_d. If an uninserted visit still remains, a new route is created via function *OpenNewRoute* (the allowed extra capacity NC_d is considered) and the process is repeated until all visits are inserted or the allowed extra capacity *NC* is exhausted.

Algorithm 2. Design Routes

```
Input: (h, V, TT, O, C, Dur, Rel, Dead, OR, D, B, E, NC)
Output: (OR, D, B, E, R, A)

U = V                    // U is the set of uninserted visits
while U ≠ ∅
    ⟨OR, D, B, E, R, A, U⟩ = InsertToRoutes(
                                    h, U, TT, O, C, Dur, Rel, Dead, OR, D, B, E)
    if U ≠ ∅
        ⟨OR, D, B, E, R, A, U, NC⟩ = InsertToRoutesWithOverTime(
                                    h, U, TT, O, C, Dur, Rel, Dead, OR, D, B, E, NC)
        if U ≠ ∅
            if NC > 0
                ⟨OR, NC, D, B, E⟩ = OpenNewRoute(OR, NC, D, B, E)
            else
                break
            end if
        end if
    end if
end while
for r∈OR
    B_{r,D} = A_{r,0}
    E_{r,D} = A_{r,n+1}
end for
```

The routing algorithm is based on the insertion heuristic (Campbell and Savelsbergh, 2004). It is implemented in function *InsertToRoutes* defined by Algorithm 3.

Algorithm 3. Insert to Routes

```
Input: (h, U, TT, O, C, Dur, Rel, Dead, OR, D, B, E)
Output: (OR, D, B, E, R, A, U)

while U ≠ ∅
    cost* = ∞
    for j∈U
        for r∈OR
            for (i-1, i)∈r
                if Feasible(r, i, j) AND Cost(r, i, j) < cost*
                    r* = r; i* = i; j* = j
                end if
            end for
        end for
    end for
    if cost* == ∞   // U contains visits that can not be feasibly inserted
        break
    else
        Insert(r*, i*, j*)
        U = U \ j*
        Update(r*)
    end if
end while
for r∈OR
    B_{r,D} = A_{r,0}
    E_{r,D} = A_{r,n+1}
end for
```

The function *Cost* evaluates the cost of the partial solution according to the cost function (2). Function *Feasible* checks feasibility of insertion visit j to the position i of the road r. Function *Insert* inserts the visit j^* to the position i^* of the route r^* and the function *Update* updates internal data after insertion. Details are described in (Campbell and Savelsbergh, 2004).

Function *InsertToRoutesWithOverTime* is almost equal to function *InsertToRoutes* defined by Algorithm 3. The only difference is that it calls function *FeasibleWithOverTime* instead of *Feasible* and function *UpdateWithOverTime* instead of *Update*. Functions *FeasibleWithOverTime* takes the allowed extra capacity *NC* as an argument and considers overtime when checking feasibility. Function *UpdateWithOverTime* decrease the remaining extra capacity *NC* by the used overtime.

5. Rostering

The Nurse Rostering Problem is solved by the function *DesignRoster*. The input to the function is the set of opened routes ***OR*** defining set of shifts to be assigned to nurses ***N***. The outstanding feature of the set of shits ***OR*** is that there are hardly ever two shifts with the same beginning $B_{r,d}$ and the same end $E_{r,d}$. Therefore, this rostering problem can be classified as ETPHD, i.e. Employee Timetabling Problem with High Diversity of shifts (Bäumelt et al., 2010), (Burke et al., 2006). The function *DesignRoster* is outlined in Algorithm 4. In the first step ETPHD problem defined by ***OR*** is solved by algorithm described in (Bäumelt et al., 2010). Consequently, the resulting roster *NR* is passed to *UpdateOpenedRoutes* function. The aim of this function is to reduce workload on critical days, i.e. days where opened routes ***OR*** cannot be covered by available nurses or the coverage causes the cost function deterioration. Therefore, visits of such routes are moved to surrounding days according to workload on these days. Finally, function returns vector *NC*, where NC_d is the extra capacity on the day $d\in$***Days*** (in person hours) that can be used to prolong any opened route or to open a new route.

Algorithm 4. Design Roster

Input: (***N***, ***Days***, ***OR***, *D*, *B*, *E, NR*)
Output: (***OR***, *D*, *B*, *E*, *N*, *NR, NC*)

⟨*NR, N*⟩ = *SolveRosteringProblem*(***N***, ***Days***, ***OR***, *D*, *B*, *E*)
⟨***OR***, *D*, *B*, *E*, *NC*⟩ = *UpdateOpenedRoutes*(***OR***, *D*, *B*, *E, NR*)

6. Conclusion

We have presented integration of algorithms for solving vehicle routing problem and nurse rostering problem in this paper. Both sub-problems algorithms have been implemented in C#. They strictly employ object oriented principles which allows flexible modification of considered constraints, cost function and the way in which heuristics search the solution. The algorithms can be therefore customized to concrete user needs.

Acknowledgement
This work was supported by the Ministry of Education of the Czech Republic under the Project 2C06017 and Research Programme MSM6840770038.

References

Burke E.K., De Causmaecker P., Berghe G.V., Landenghem H.V. (2004), "*The State of the Art of Nurse Rostering*", *Journal of Scheduling*, 7, 6: 441-499.

Ernst A.T., Jiang H., Krisnamoorthy M., Sier D. (2004), "*Staff Scheduling and Rostering: A Review of Applications, Methods and Models*", *European Journal of Operational Research*, 153, 1: 3-27.

Burke E.K., De Causmaecker P., Petrovic S., Berghe G.V (2006), "*Metaheuristics for Handling Time Interval Coverage Constraints in Nurse Scheduling*", *Applied Artificial Intelligence: An International Journal*, 20, 9: 743-766.

Bertels S., Fahle T. (2006) "*A Hybrid Setup for a Hybrid Scenario: Combining Heuristics for the Home Health Care Problem.*" *Computers & Operations Research*, 33, 10: 2866–2890.

Eveborn, E., Flisberg, Rönnqvist M., (2006), "*Laps Care - An Operational System for Staff Planning of Home Care*", *European Journal of Operational Research*, 171, 3:962–976.

Bäumelt Z., Šůcha P., Hanzálek Z., (2010), "*An Evolutionary Algorithm in a Multistage Approach for an Employee Rostering Problem with a High Diversity of Shifts*", PATAT 2010, Belfast, Northern Ireland.

Campbell A. M, Savelsbergh M. (2004), "*Efficient Insertion Heuristics for Vehicle Routing and Scheduling Problems*", *Transportation Science*, 38, 3: 369-378.