Extended Decomposition for Mixed Integer Programming to Solve a Workforce Scheduling and Routing Problem

Wasakorn Laesanklang, Rodrigo Lankaites Pinheiro, Haneen Algethami, and Dario Landa-Silva

ASAP research group, School of Computer Science, The University of Nottingham, Jubilee Campus, Wollaton Road, Nottingham, NG8 1BB, United Kingdom

Abstract. We propose an approach based on mixed integer programming (MIP) with decomposition to solve a workforce scheduling and routing problem, in which a set of workers should be assigned to tasks that are distributed across different geographical locations. We present a mixed integer programming model that incorporates important real-world features of the problem such as defined geographical regions and flexibility in the workers' availability. We decompose the problem based on geographical areas. The quality of the overall solution is affected by the ordering in which the sub-problems are tackled. Hence, we investigate different ordering strategies to solve the sub-problems. We also use a procedure to have additional workforce from neighbouring regions and this helps to improve results in some instances. We also developed a genetic algorithm to compare the results produced by the decomposition methods. Our experimental results show that although the decomposition method does not always outperform the genetic algorithm, it finds high quality solutions in practical computational times using an exact optimization method.

Keywords: Workforce scheduling, Routing problem, Mixed integer programming, Problem decomposition, Genetic algorithm

1 Introduction

The workforce scheduling and routing problem (WSRP) is a difficult problem that arises in industries like home care, health care, patrol service, meter reading, etc. [9]. One specific example of WSRP is home health care services where nurses or care workers should deliver care services to patients at their home. Solving the problem involves producing a job schedule and a route for each worker while satisfying the business requirements and considering workers qualifications and skills, task requirements, travelling distances, etc. It is usually expected that the solution gives the lowest operational cost.

Developing automated solution methods to solve WSRP scenarios is a current research challenge as reflected by recent published research [17, 22, 23]. Solving

an instance of WSRP often requires the expertise and knowledge of a human planner [4]. In this research, we are working with an industrial partner who provides scheduling services for businesses and other organisations facing this type of problems. The data sets considered here come from real-world scenarios. A particular feature is that 'working areas' or 'regions' are pre-defined and this affects the difficulty of the problem being tackled.

Tackling WSRP with exact optimization methods to produce solutions in practice is still a research challenge. Since obtaining an optimal schedule is the ultimate goal, exact methods like mathematical programming are a suitable approach. However, proven optimality with exact methods has been shown mainly on problem instances of limited size [7, 8]. Problem instances faced in practice are larger and for solving them, heuristic methods are usually considered more efficient in terms of computation time [2, 27]. The motivation for our work is to develop a solution approach based on exact optimization to tackle real-world WSRP instances.

In this paper, we propose a decomposition approach that uses mixed integer programming (MIP) to tackle WSRP instances of practical size. The proposed method splits the problem into sub-problems according to defined geographical areas. Our computational experiments show that the approach allows to explore the trade-off between computation efficiency and solution quality. We also use a process that brings additional workforce to understaffed regions from neighbouring regions. This process is applied when required before further splitting the regions into smaller sub-problems. The results from our experiments suggest that the success of the proposed decomposition varies according to the problem instance, which provides us with ideas for future research.

The main contribution of this paper is to show that the ordering in which sub-problems in workforce scheduling and routing are tackled within a decomposition approach, has an effect on the computational efficiency and achieved solution quality. Following this, some ordering strategies are proposed to achieve solutions of good quality in practical computation time. Moreover, we also compare the results produced with the decomposition method to the results obtained with a standard genetic algorithm and to the optimal solution when available.

Section 2 reviews related works in the literature and establishes the motivation for the research in this paper. Section 3 gives the problem definition and formulation for the WSRP considered here. Section 4 describes the proposed MIP with decomposition approach and the experimental study, including description of test data instances. The final section summarises the paper and outlines some of the proposed future work.

2 Literature Review

Solving integer programming formulations for larger problem instances still has its limitations in terms of computation time. Mathematical programming has been used in the literature to tackle some WSRP. Examples include linear programming [3], integer programming [20] and mixed integer programming [7, 8,

13, 30]. To solve real-world sized problems, works in the literature often resource to heuristic or hybrid algorithms [2, 6, 14]. There are some improved exact methods, like branch and price [8, 13, 30], that can deal with large scenarios. Branch and price requires problem reformulation which usually involves Dantzig-Wolfe decomposition to compute a tighter relaxation bound [33, 15]. The algorithm also requires two steps to repeatedly solve the problem in order to improve the solution.

Decomposition techniques are another good alternative to apply exact optimization methods to large integer programming formulations. The basic idea is to transform or split the problem into smaller sub-problems. This technique has been applied in various problem domains. For example, Benders' decomposition was used to produce solutions for large instances of the aircraft routing and crew scheduling problem [11, 24]. Benders' decomposition is suitable for problems with exclusive sub-problem sets or problems that show some block structures linked by constraints [5]. In another example of decomposition, [26] split the warehouse location-routing problem into three smaller problems: the complete multi-depot vehicle-dispatch problem, the warehouse location-allocation problem and the multi-depot routing-allocation problem. These three smaller problems were solved in phases and each of them was formulated with mathematical programming and solved by an exact solver. For detailed reviews of decomposition approaches see [29, 34].

Decomposition techniques have also been applied within heuristic approaches using some form of clustering. For example, [31] tackled a large vehicle routing problem by decomposing it into sub-problems. Each sub-problem was a cluster of customers assigned to a vehicle which then became a travelling salesman problem. The sub-problem size is controlled by splitting a large sub-problem to shrink the corresponding cluster. Similar ideas were applied in a hybrid heuristic for generating multi-carrier transportation plans [21].

In this work we propose a decomposition approach that uses mixed integer programming (MIP) to tackle workforce scheduling and routing problem instances arising in real-world scenarios. For this, we also present an MIP formulation that incorporates features of the WSRP scenarios faced by our industrial partner. The proposed decomposition technique does not require some formulation structure like in Benders' decomposition neither uses a heuristic solver. Our approach harness the power of exact optimization solvers while decomposing the problem instances in a way that is meaningful to practice.

3 Problem Description and Formulation

The goal in WSRP is to assign each worker to undertake a set of tasks across a set of geographical locations. A path is the series of tasks to be performed by a worker within the planning period. A good quality solution consists of a set of shortest paths representing the task assignments for each worker at the lowest cost. The solution should also respect other conditions such as task pre-requisites,

required travelling time between locations, defined appointment times, workers' skills, workers availability, restricted working regions, working time limits, etc.

Consider a graph $G = (V, E)$ where $V = T \cup D \cup D'$ represents the union of a set of tasks (each task as a location) T, a set of start locations D and a set of end locations D' while E represents a set of links between two locations (e.g. between two task locations or between the worker's home and a task location). The set of workers is denoted by C. Binary decision variable $x^c_{i,j} = 1$ if worker $c \in C$ is assigned to a task $j \in T$ after finishing task $i \in T$, $x^c_{i,j} = 0$ otherwise. Note that elements of T are referred here as tasks but also each task has an associated location.

In real-world scenarios like the ones considered here, the available skilled workforce is often not sufficient to cover all the required tasks. We introduce a dummy worker to address this issue (through an integer decision variable y_j) that takes any uncovered task that cannot be assigned to the available workforce [1, 30]. In our problem some tasks may require more than one worker. We denote the number of workers required to perform a task by b_j. Therefore, the required number of workers must be assigned to each task. The dummy worker can be used for that effect too, even taking the whole task as $y_j = b_j$. Then, the assignment of tasks is represented by (1).

$$\sum_{c \in C} \sum_{i \in D \cup T} x^c_{i,j} + y_j = b_j \qquad , \forall j \in T \tag{1}$$

The sequence of tasks performed by a worker is represented as a path for visiting task locations, hence the number of workers arriving at one location must be equal to the number of workers leaving that task location so that either workers are assigned to the next task or go home. Then, the path constraint is represented by (2).

$$\sum_{i \in D \cup T} x^c_{i,j} = \sum_{k \in D' \cup T} x^c_{j,k} \quad , \forall j \in T, \forall c \in C \tag{2}$$

Workers must start and end their paths from their specific location (e.g. their home or a central office) as given by (3) and (4). Since D and D' are sets of start and end locations respectively, these two constraints indicate the start and end locations for each worker. Also, workers leave their start location and enter their end location at most once (although the start and end locations can be different) as expressed by (5) and (6) respectively. Note that a worker does not leave his start location if he is not assigned to work. This is different from the common case in the literature where all workers leave their start location. In our problem instances, the specific start and end locations are provided for every worker.

$$\sum_{j \in D' \cup T} x^c_{k,j} \geq \sum_{j \in D' \cup T} x^c_{i,j} \qquad , \forall c \in C, \forall i \in T, \exists k \in D \tag{3}$$

$$\sum_{i \in D \cup T} x^c_{i,k} \geq \sum_{i \in D \cup T} x^c_{i,j} \qquad , \forall c \in C, \forall j \in T, \exists k \in D' \tag{4}$$

$$\sum_{j \in D' \cup T} x_{i,j}^c \leq 1 \qquad , \forall i \in D, \forall c \in C \tag{5}$$

$$\sum_{i \in D \cup T} x_{i,j}^c \leq 1 \qquad , \forall j \in D', \forall c \in C \tag{6}$$

Let S be the set of skills and $s \in S$ a particular skill. For worker c the qualification level on skill s is q_s^c and for task j the requirement of skill s is $r_{s,j}$. Hence, worker c can be assigned to task j only if the worker has the required qualifications level for skill s, that is, $q_s^c \geq r_{s,j}$. Then, in our model the multi-skill qualification requirements are represented by (7).

$$x_{i,j}^c r_{s,j} \leq q_s^c \qquad , \forall c \in C, \forall i \in D \cup T, \forall j \in T, \forall s \in S \tag{7}$$

Also, travel time between task locations must be feasible. Decision variable a_i^c takes a positive fractional value that gives the worker arrival time to the location of task i. Note that the maximum arrival time value is 1440 which is equivalent to the 24^{th} hour of the day. Let a_i^c, a_j^c be the arrival times of worker c to the locations of task i and task j respectively. Let $t_{i,j}$ be the travelling time between the locations of tasks i and j. Let δ_i be the duration of task i. Then, if worker c is assigned to perform task j after completing task i, inequality (8) (M is a large constant number) expresses the arrival on time requirement.

$$a_j^c + M(1 - x_{i,j}^c) \geq a_i^c + x_{i,j}^c t_{i,j} + \delta_i \qquad , \forall c \in C, \forall i \in D \cup T, \forall j \in D' \cup T \tag{8}$$

An arrival time window is also defined for task j and the worker should not arrive earlier than w_j^L or later than w_j^U, as expressed by (9).

$$w_j^L \leq a_j^c \leq w_j^U \qquad , \forall j \in T, \forall c \in C \tag{9}$$

An important feature of our WSRP scenarios is that working regulations and availability can be specific for each worker. In the problems considered here, this refers to long breaks between shifts (short breaks within the working shift are not considered), days-off, working shift duration, maximum working hours, and specific worker preferences (e.g. late morning, afternoon only, whole day, overnight). We adopt a flexible availability constraint from an optimization of daily scheduling for home health care services [32]. Any task assignment at time a_j^c including the task duration δ_j should lie in between the shift starting time α_L^c and the shift ending time α_U^c. The availability parameters α_L^c and α_U^c are real numbers defined for each worker c. A task assigned outside the shift is charged as additional expense, hence binary decision variable $\omega_j = 1$ if this is the case and $\omega_j = 0$ otherwise. Then, individual availability constraints are denoted by (10) and (11) while the working hours limit (h^c) constraint is denoted by (12).

$$\alpha_L^c - a_j^c \leq M(1 - x_{i,j}^c + \omega_j) \qquad \forall c \in C, \forall i \in D \cup T, \forall j \in T \tag{10}$$

Table 1: Summary of problem requirements and type of constraints.

	Hard	Soft		Hard	Soft
Job assignment (Cons. (1))		*	Start-end paths (Cons. (3) - (6))	*	
Path constraint (Cons. (2))	*		Travel time feasibility (Cons. (8))	*	
Time windows (Cons. (9))	*		Skill and qualification (Cons. (7))	*	
Working hours (Cons. (12))	*		Worker availability(Cons.(10),(11))	†	*
Working regions(Cons.(13))		*			

† Hard constraints (15) and (16) are described in Section 4.2 and only apply to the decomposition model.

$$a_j^c + \delta_j - \alpha_U^c \leq M(1 - x_{i,j}^c + \omega_j) \qquad \forall c \in C, \forall i \in D \cup T, \forall j \in T \tag{11}$$

$$\sum_{i \in D \cup T} \sum_{j \in T} x_{i,j}^c \delta_j \leq h^c \qquad , \forall c \in C \tag{12}$$

Another important feature of our WSRP scenarios is that workers have preferred geographical areas for working but the decision maker can still request workers to work outside those preferred regions. We formulate this in (13) where binary parameter $\gamma_j^c = 1$ if worker c is willing and able to work at the location of task j, $\gamma_j^c = 0$ otherwise, and binary decision variable $\psi_j = 1$ if worker c is forced to work outside their defined regions.

$$\sum_{i \in D \cup T} x_{i,j}^c - \psi_j \leq \gamma_j^c \qquad , \forall c \in C, \forall j \in T \tag{13}$$

Most of the above constraint formulations exist in literature but not all. Common constraints (see also [9]) such as path constraint (2), skill and qualification (7) and time windows (9) form the basic structure of the scheduling and routing problem [12, 1, 13, 10, 30]. Tailor cut constraints adopted from literature are the availability constraints (10,11) while the constraints that required further adaptation to our problem features are the working region (13) (implemented as soft constraint) and start-end paths (3-6).

Table 1 summarises the constraints in our MIP model. Given our real-world data sets, some are implemented as soft constraints. For example, workers can be forced to work outside their predefined regions and availability. Also, tasks can be left unassigned (assigned to the dummy worker). These features are quite important to maintain the practical applicability of our model and solution approach.

The objective function (14) involves three costs: monetary cost, preferences penalty cost and soft constraints penalty cost.

$$\text{Min} \sum_{c \in C} \sum_{i \in D \cup T} \sum_{j \in D' \cup T} \lambda_1 \left(d_{i,j} + p_j^c\right) x_{i,j}^c + \sum_{c \in C} \sum_{i \in D \cup T} \sum_{j \in D' \cup T} \lambda_2 \rho_j^c x_{i,j}^c + \sum_{j \in T} (\lambda_3(\omega_j + \psi_j) + \lambda_4 y_j) \quad (14)$$

The first term in (14) is the monetary cost and includes the travelling cost $d_{i,j}$ of going from location of task i to the location of task j, and the payment p_j^c for worker c to perform task j. The second term in (14) is the preference penalty cost denoted by $\rho_j^c \geq 0$ and is a summation of penalties for not meeting worker-client preferences, required skill preferences and working region preferences. This penalty value can go from 0 to 2 and $\rho_j^c = 0$ when all preferences are met, while this penalty value grows higher as the preference level of assigning worker c to task j decreases. The third term in (14) is the soft constraints penalty cost due to the violation of the three soft constraints in the model. The job assignment constraint has the highest priority, so a violation of this constraint type costs more than a violation of the the other two constraints. The worker availability and working regions constraints have the same priority. Note that the working regions constraint is involved in two costs. If the worker is assigned a task in a non-preferred region then this is a constraint penalty cost. If the worker is assigned to one of the preferred regions this is quantified as a preference penalty cost according to the degree in which the region is preferred by the worker (several working regions with different preference levels). Note that $\lambda_1, \lambda_2, \lambda_3$ and λ_4 are weights that set the priorities between objectives as $\lambda_1 < \lambda_2 < \lambda_3 < \lambda_4$. The value assigned to these weights are included with the data instances.

The above MIP model corresponds to the integrated scheduling and routing problem. Solving this model with an exact optimization method is not practical considering our real-world problem instances. Hence, we apply a decomposition technique.

4 Decomposition Approach and Study

In order to reduce the overall computational time for solving real-world instances of the integrated workforce scheduling and routing problem, we now present a decomposition method. First, we describe the features of our problem instances as this will help to explain the proposed decomposition approach. Later, the method is described and experimental results are provided.

4.1 Test Instances

For the present work, we prepared some test instances using real-world data corresponding to home care scenarios in the UK, provided by our industrial partner. A problem instance P has a set of nodes V. Recall from Section 3 that $V = D \cup T \cup D'$. Also, some of the tasks $\{j_1, j_2, \ldots, j_n\}$ in T share the same

Table 2: The test data sets.

Instance	$\|C\|$	$\|K\|$	$\|T\|$	$\|A\|$	Instance	$\|C\|$	$\|K\|$	$\|T\|$	$\|A\|$
WSRP-A-01	23	25	31	6	WSRP-B-01	25	27	36	6
WSRP-A-02	22	24	31	4	WSRP-B-02	25	11	12	4
WSRP-A-03	22	28	38	5	WSRP-B-03	34	43	69	6
WSRP-A-04	19	22	28	3	WSRP-B-04	34	14	30	4
WSRP-A-05	19	9	13	3	WSRP-B-05	32	38	61	8
WSRP-A-06	21	22	28	7	WSRP-B-06	32	38	57	7
WSRP-A-07	21	9	13	3	WSRP-B-07	32	38	61	7
WSRP-D-01	164	233	483	13	WSRP-F-01	805	477	1211	45
WSRP-D-02	166	215	454	12	WSRP-F-02	769	496	1243	46
WSRP-D-03	174	279	585	15	WSRP-F-03	898	582	1479	54
WSRP-D-04	174	237	520	15	WSRP-F-04	789	513	1448	47
WSRP-D-05	173	259	538	15	WSRP-F-05	883	626	1599	59
WSRP-D-06	174	291	610	15	WSRP-F-06	783	565	1582	44
WSRP-D-07	173	293	611	15	WSRP-F-07	1011	711	1726	64

$|C|$ = number of workers.
$|K|$ = number of task locations.
$|T|$ = number of required tasks.
$|A|$ = number of working regions.

geographical location $\kappa \in K$, where K is a set of geographical locations. A group of locations are assembled as a geographical region or working region $a \in A$. Note that $a \subseteq K$ and A is a partition. Also, an individual worker c may work on one or several geographical regions. As noted above, a key aspect of our scenarios is that several tasks might be required at one particular location. Each individual task may have different required skills, worker preferences and worker cost.

We took four real-world scenarios and prepared a data set from each. Although the instances in each data set come from the same scenario, each instance is formed from a different planning time giving a variation in the available human resources and task requirements. In our data, the start and end locations of a worker are the same ($d = d'$). Table 2 shows the main features of the test instances: the number of available workers $|C|$, the number of task locations $|K|$, the numbers of tasks $|T|$ and the number of predefined geographical regions $|A|$. In terms of size, instances WSRP-A-(01-07) and WSRP-B-(01-07) are considered small with around 19-34 workers and 13-69 tasks. The optimal solution for each of these instances can be found in less than 5 minutes. Instances WSRP-D-(01-07) and WSRP-F-(01-07) are considered large with more than 100 workers and 400 tasks. These large instances cannot yet be solved to optimality in practical computation time. In our experimental study, we use the small instances to validate the proposed decomposition approach as we can compare to the optimal solutions. Moreover, the experimental results show the suitability of the decomposition approach in tackling the large instances using an exact optimization solver.

Algorithm 1: Geographical decomposition with conflict avoidance (GDCA).

Data: Problem instance $P = \{C, A\}$
1 initialization: For worker $c \in C$, define earliest availability vector $\beta_L = (\beta_L^c)$ and latest availability vector $\beta_U = (\beta_U^c)$;
2 Split problem P by regions denoted as $P_i = \{C, a\}, a \in A, i = 1...|A|$;
3 **forall the** $P_i \in P$ **do**
4 Solve P_i with availability α_L and α_U by CPLEX solver$\rightarrow \Phi_i$;
5 Update availability vector β_L and β_U;
6 **end**
7 Combine sub-problem solutions;

4.2 Geographical Decomposition with Conflict Avoidance

In this paper, the workforce scheduling and routing problem is decomposed into working regions as this is a key feature of the scenarios provided by our industrial partner. Since we decompose the problem into sub-problems to deal with the larger size more efficiently, by solving the sub-problems one at a time in a given sequence, we can no longer guarantee overall optimality.

Basically, the decomposition method generates a sub-problem for each working region and solves each sub-problem in sequence. Worker assignment conflicts (i.e. a worker being assigned to different task locations at the same time) are avoided because each sub-problem is solved using only the reduced available workforce after solving the previous sub-problem.

Algorithm 1 presents the proposed geographical decomposition with conflict avoidance approach (GDCA). A problem instance P is split into several sub-problems P_i (step 2). A sub-problem P_i corresponds to a geographical region or working region $a \in A$. Some regions may generate a sub-problem that is too large. Hence, we further split them until the sub-problem has no more than around ten locations. Then, the sub-problems are solved in a given sequence (steps 3-6) and different solving sequences can lead to different solution quality. This is because the first sub-problem has access to the most workforce resources but subsequent sub-problems will have access to limited available workforce. Since worker assignment conflicts are avoided, this means that the hard constraints expressed by equations (15) and (16) are enforced in this algorithm.

$$a_j^c + \delta_j - \beta_L^c \leq M(2 - x_{i,j}^c - \zeta^c) \qquad \forall c \in C, \forall i \in D \cup T, \forall j \in T \cup D' \tag{15}$$

$$\beta_U^c - a_j^c \leq M(1 - x_{i,j}^c + \zeta^c) \qquad \forall c \in C, \forall i \in D \cup T, \forall j \in T \cup D' \tag{16}$$

Here, β_L^c denotes the start of unavailable time and β_U^c denotes the end of unavailable time for worker c. Since the original model generates a continuous path for a worker, a path created under hard availability conditions is allocated either before or after the unavailability period. That is, a path which overlaps with the unavailability period defined by β_L^c and β_U^c is not allowed. The control variable ζ^c is applied for selecting only one side of the availability period. When

$\zeta^c = 1$ the time interval before β_L^c is selected and if the $\zeta^c = 0$ the time interval after β_U^c is selected.

In our data, we know that the start location d and end location d' for a worker are the same. Therefore, we designed a sub-problem solutions combination process based on this assumption. During the **Combine sub-problem solutions** process (step 7), sub-problem solutions are combined together by connecting the worker's paths from each sub-problem to get a long single path. After this process, a worker leaves his start location and arrives to his end location only once. Suppose that $\Phi_1 = \{(x^c_{d,t_1}, a^c_1), (x^c_{t_1,d'}, a^c_{d_1})\}$ is a solution to sub-problem P_1 representing the assignment of worker c from start location d to work on task t_1 and returning to end location d' and $\Phi_2 = \{(x^c_{d,t_2}, a^c_2), (x^c_{t_2,d'}, a^c_{d_2})\}$ is a solution to sub-problem P_2 representing the assignment of the same worker c from starting location d to work on task t_2 and returning to ending location d'. Assume without loss of generality that $a^c_1 < a^c_{d_1} < a^c_2 < a^c_{d_2}$. The combining process redirects the arriving assignment to end location d' to task t_2 which gives a global solution as $\Phi = \{(x^c_{d,t_1}, a^c_1), (x^c_{t_1,t_2}, a^c_2), (x^c_{t_2,d'}, a^c_{d_2})\}$. It is possible than in other scenarios of the WSRP, the start location and end location for a worker are different, we leave this for future work as it is not a feature of the scenarios tackled at present.

4.3 Experimental Study of the Decomposition Method

We conduct an experimental study to gather insights into the performance of the proposed geographical decomposition method. The flow of the study is depicted in Figure 1. The figure outlines the three parts of the experimental design. First, on the left-hand side of the figure, the **permutation study** refers to solving the sub-problems in different order given by all the different permutations of the geographical regions. However, trying all permutations is practical only in small problems. Therefore, finding an effective ordering pattern is the second part of the experiment, **observation step** in the figure. This second part solved each sub-problem using all available workforce, i.e. ignoring if some workers were assigned in previous sub-problems. The third part analysed the results from the observation step in order to define some strategies to tackle the sub-problems. Based on this **strategies study**, some solving strategies were conceived. Listed in the figure are these ordering strategies: Asc-task, Desc-task, Asc-w&u, etc. More details about these ordering strategies are provided when describing the **Observation step** below. Finally, the solutions produced with the different ordering strategies are compared to the solutions produced by the permutation study to evaluate the performance of these ordering strategies.

Permutation Study. Since the number of permutations grows exponentially with the number of geographical regions, we performed the permutation study using only the instances with $|A| = 3$ and $|A| = 4$ geographical regions. Figure 2 shows the relative gap obtained for the small instances that have 3 regions. Each sub-figure shows the results for one instance when solved using the different permutation orders of the 3 regions. Each bar shows the relative gap between the solution by the decomposition method and the overall optimal solution. The figure

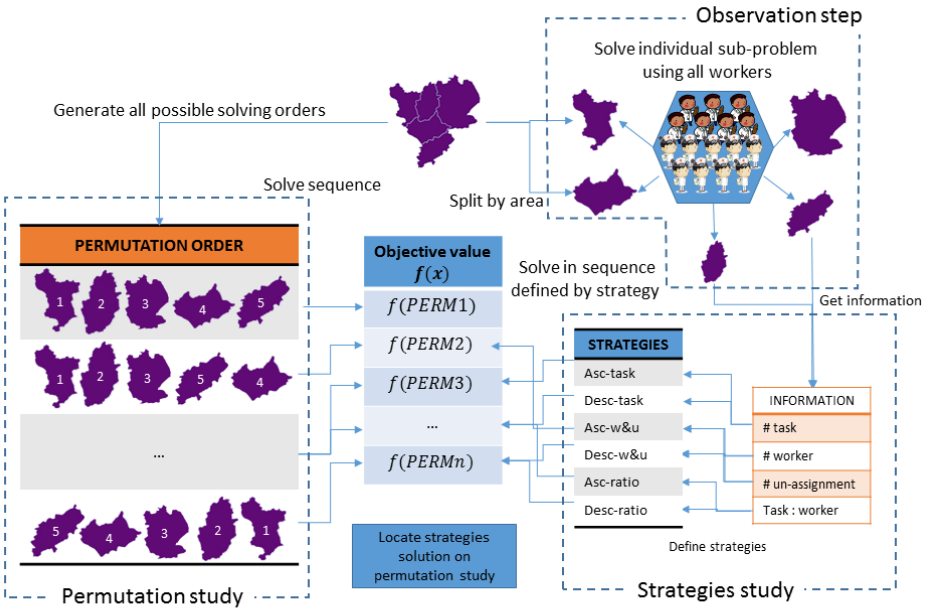

Fig. 1: Outline of the experimental study in three parts: permutation study, observation step and strategies study.

shows that the quality of the obtained solutions for the different permutations fluctuates considerably. Closer inspection reveals that in these instances the geographical regions are very close to each other and sometimes there is an overlap between them. The result also reveals that some permutations clearly give better results. For example, permutation "1-2-3" for instance WSRP-A-04, permutations "1-2-3" and "2-1-3" for instance WSRP-A-05 and permutations "1-3-2" for instance WSRP-A-07.

Figure 3 shows the relative gap obtained for the small instances that have 4 regions. Each sub-figure shows the result for one instance when solved using the permutation orders of the 4 regions. Each bar shows the relative gap between the solution by the decomposition method and the overall optimal solution. The figure reveals an interesting result from instance WSRP-B-02. The optimal solution value is obtained for every permutation. Closer inspection reveals that the decomposition method works very well on this instance because its geographical regions are well separated from each other. Therefore, the sub-problem solutions are part of the complete overall solution and not many worker assignment conflicts arise when solving the sub-problems. For the other instances, WSRP-A-02 and WSRP-B-04, the quality of the obtained solutions fluctuates in the same way as in Figure 2. Results in Figure 3 indicate that some solutions obtained with the decomposition approach using some permutations have a considerable gap in quality compared to the overall optimal solution. The figure also shows that some

permutations clearly give better results than others. For example, permutation "2-4-3-1" and "3-1-2-4" for instance WSRP-A-02, permutation "1-2-3-4", "1-2-4-3", "2-1-3-4", "2-1-4-3" and "2-3-1-4" for instance WSRP-B-02 and permutation "4-3-1-2" and "4-3-2-1" for instance WSRP-B-04.

The conclusion from this permutation study is that the order in which the sub-problems are solved matters differently according to the problem instance. More importantly, the results confirm our assumption that some particular permutation could produce a very good result in the decomposition approach. Hence, the next part of the study is to find a good solving order.

Observation step. Here we solve each of the sub-problems using all available workers and collect the following values from the obtained solutions: number of tasks in the sub-problem (# task), minimum number of workers required in the solution (# min worker), number of unassigned tasks in the solution (# unassigned task) and the ratio of tasks to worker in the solution (task/worker ratio). Then, we defined six ordering strategies as follows. Increasing number of tasks in the sub-problem (Asc-task); decreasing number of tasks in the sub-problem (Desc-task); increasing sum of minimum workers required and unassigned tasks (Asc-w&u); decreasing sum of minimum workers required and unassigned tasks (Desc-w&u); increasing ratio of tasks to worker (Asc-ratio) and decreasing ratio of tasks to worker (Desc-ratio).

Strategies study. The GDCA approach is again executed using the 6 ordering strategies listed above to tackle the sub-problems in each problem instance. The results are presented in Figure 4 which shows the relative gap for the 14 small instances in the WSRP-A and WSRP-B groups. Note that each bar represents the relative gap obtained with each strategy.

From Figure 4, the decomposition technique with ordering strategies gives solutions with relative gaps up to 70%. On average, the decomposition technique produces relative gap at 30.77%. Moreover, we can see that some of the ordering strategies are more likely to produce better solutions than others. The best performing ordering strategy is Asc-w&u that gives 8 best solutions considering all 14 small instances. The average gap for the ordering strategies Asc-task,

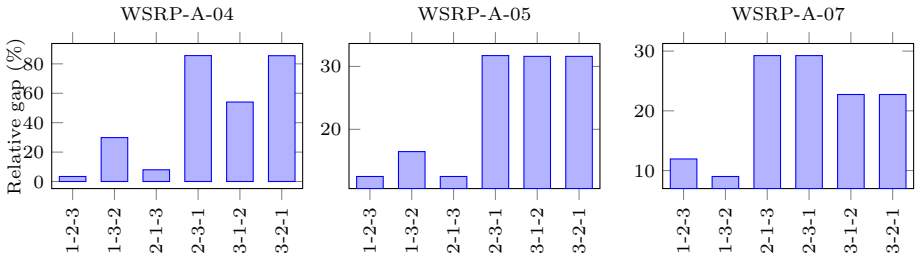

Fig. 2: Relative gap obtained from solving the 3 instances (WSRP-A-04, WSRP-A-05 and WSRP-A-07) with $|A| = 3$ using the different permutation orders. Each graph shows results for one instance. The bars represent the relative gap between the solution obtained with the decomposition method and the overall optimal solution.

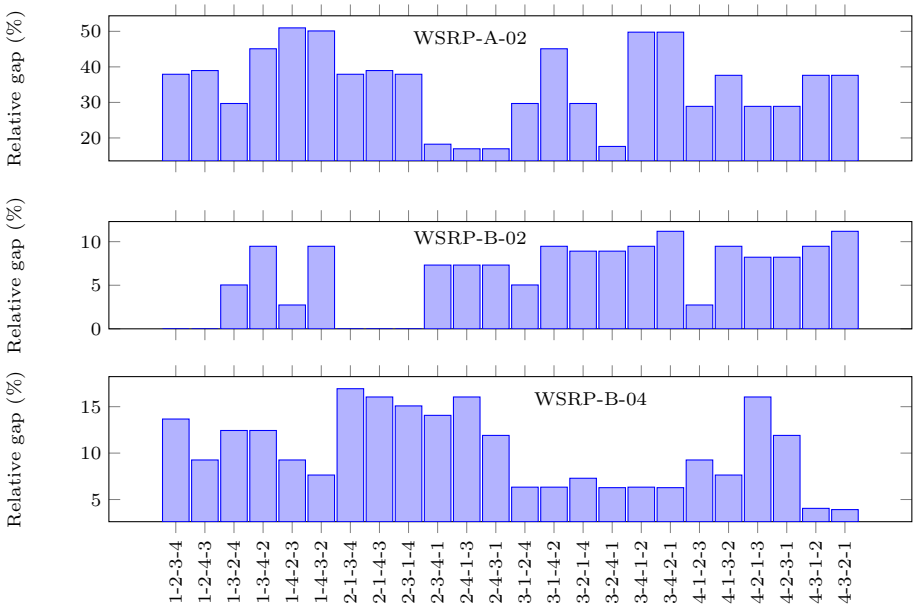

Fig. 3: Relative gap obtained from solving the 3 instances (WSRP-A-02, WSRP-B-02 and WSRP-B-04) with $|A| = 4$ using the different permutation orders. Each graph shows results for one instance. The bars represent the relative gap between the solution obtained with the decomposition method and the overall optimal solution.

Table 3: Relative gap (%) of best permutation VS. best strategy.

Instance	B.Permutation	B.Strategy	Instance	B.Permutation	B.Strategy
WSRP-A-04	3.41	7.97	WSRP-A-02	16.95	29.51
WSRP-A-05	12.50	12.50	WSRP-B-02	0	0
WSRP-A-07	9.01	11.95	WSRP-B-04	3.91	6.28

Desc-task, Asc-w&u, Desc-w&u, Asc-ratio and Desc-ratio are 27.14%, 32.62%, 27.39%, 33.26%, 31.61% and 32.62% respectively. Table 3 shows a comparison of relative gap between the best permutation order (see **Permutation study**) and the best ordering strategy. There are differences between the best strategies and the best permutation, the maximum being 12.56% for instance WSRP-A-02. Two out of six solutions (instance WSRP-A-05 and WSRP-B-02) of the best ordering strategy match the solution from the best permutation. This shows that the ordering strategies are able to work well in other problem instances.

The decomposition method is also able to find solutions for the large instances whilst solving those problems as a whole is not practical in terms of computation time. The results from using the decomposition technique with the 6 ordering strategies on the large instances are presented in Table 4. The table shows the objective values of the obtained solutions as relative gaps cannot be computed

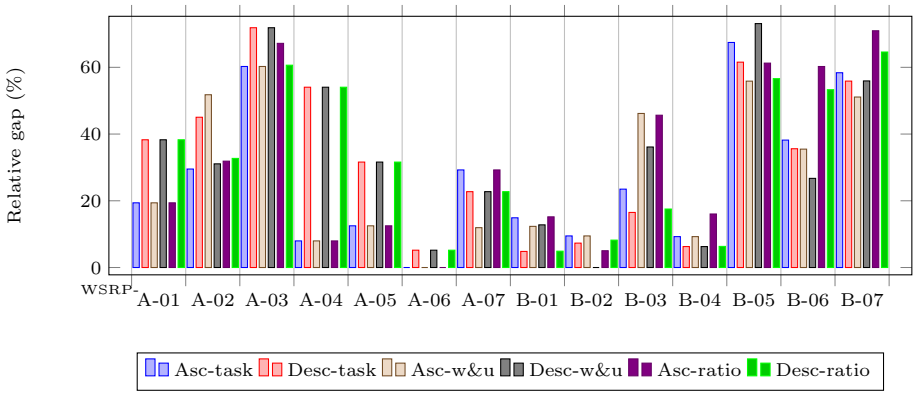

Fig. 4: Relative gap obtained from solving the 14 small instances using the 6 ordering strategies. Each bar for an instance represents the relative gap between a solution by the decomposition method using an ordering strategy and the overall optimal solution.

because the optimal solutions are not known. The values in **bold** are the lowest cost (best objective value) obtained among the six strategies. The table shows that as a whole, Desc-task gives six best solutions, Desc-ratio gives four best solutions, Asc-w&u gives two best solutions, Desc-w&u and Asc-task give one best solution while the Asc-ratio gives no best solution. On average, the Desc-task strategy gives the lowest cost solution, around 15.42% less than the highest average cost strategy (Asc-task).

Figure 5 shows, according to the problem size, the computation times used by the decomposition approach using the different ordering strategies and the time used to find the overall optimal solution. Each sub-figure presents the problem instances classified by their size (number of items is $|T|+|C|$). Each line represents the time used by the ordering strategy in solving the group of 14 problem instances. As noted before, the time to find the optimal solution represented by [- - -] is available only for the small instances. For the smaller instances which smaller than instance B-06 (has 89 items), the computation time used by the decomposition method is not much different from the time used to find the optimal solution. The computation time used to find the optimal solution grows significantly for instances B-06 to B-03. Note that instance WSRP-B-03 which has 109 items uses 5,419 seconds for finding optimality. For the large instances, it is shown that the computation time used by the decomposition method starts from 17 minutes (1,060 seconds) to above 6 hours (22,478 seconds). Also, for the large instances the average computation time used by six strategies are 4,620, 3,098, 7,451, 6,348, 7,640 and 7,048 seconds respectively. The average processing time shows Asc-task and Desc-task use significantly less computation time. This is because these ordering strategies do not require an additional process to retrieve information about the problem. Hence, considering both solution quality and computation time, it can be concluded that Asc-task and Desc-task (the second best known on average) should be selected for large instances because they produce solutions

Table 4: Objective value obtained from solving large instances using six ordering strategies.

Instance	Asc-task	Desc-task	Asc-w&u	Desc-w&u	Asc-ratio	Desc-ratio
D-01	1,688.07	496.45	1,549.04	765.48	1,301.04	**240.98**
D-02	860.50	**372.94**	496.47	495.44	984.98	732.97
D-03	2,624.84	3,213.32	2,619.33	3,836.84	**1,690.81**	3,839.34
D-04	312.43	418.89	303.45	**283.91**	314.42	420.41
D-05	408.42	243.89	1,113.47	253.91	401.45	**241.89**
D-06	**307.55**	1,411.27	945.60	1,582.52	634.05	1,729.29
D-07	1,112.80	753.28	**292.55**	604.01	293.53	1,077.28
F-01	73,286	64,305	71,430	72,040	75,760	**63,680**
F-02	81,852	**73,291**	76,460	80,569	86,906	74,860
F-03	141,060	115,235	140,258	120,715	148,092	**116,011**
F-04	111,671	102,994	105,262	109,411	113,557	**91,670**
F-05	127,476	**101,438**	113,403	105,284	112,995	103,156
F-06	105,595	**76,007**	88,702	84,050	107,281	84,050
F-07	199,160	**176,541**	194,525	178,387	218,058	178,387
Average	60,529.69	**51,194.39**	56,954.29	54,162.73	62,019.34	51,435.43

Bold text refers to the best solution.

which are not much different from the other strategies but requiring significantly less computational time (48% less on average).

4.4 Geographical Decomposition with Neighbour Workforce

The study on the proposed decomposition method shows some limitation on tackling the working regions constraint. Recall that in our problem we consider the regions constraint to be soft. Assigning a worker to a region other than his/her own is allowed by penalised. However, the geographical decomposition technique described above enforces the regions constraint because each sub-problems corresponds to one region. When solving each sub-problem, the solver can only use the workers included in the sub-problem. Adding neighbour workers from nearby regions increases the feasible region by treating the regions constraint as soft. We only add workers to those regions in which the number of workers is less than the number of locations. Also, in order to maintain the size of the sub-problem manageable we add only enough workers to fill that difference. These additional workers are selected based on the distance from their departure location to the region, hence considered neighbour workforce. But also, the additional workers are selected from those with the highest set of skills and qualifications to be eligible to work in most of the tasks. This process of adding neighbour workforce is done before the process of further splitting a sub-problem. The only instances that require this process of adding neighbour workforce are instances WSRP-D and WSRP-F as presented in Table 5. For each instance, the table shows in columns two and six, the number of regions that required additional workers. Columns three and seven give the average ratio between the number of available worker and the number of locations. Columns four and eight show the improvement obtained in the objective function value when using this process of adding neighbour workforce. The result shows that additional

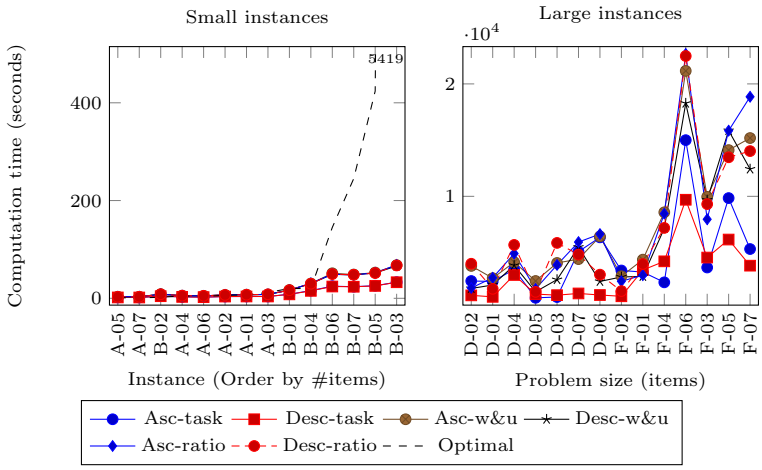

Fig. 5: Computation time (seconds) used in solving small and large instances. Each sub-figure corresponds to a problem size category (small and large). Instances are ordered by the problem size (#items) which is the summation of #workers and #tasks. Each graph presents the computation time used by the decomposition method with the different ordering strategies (line with markers) and the time used for producing the overall optimal solution (dashed line) when possible.

neighbour workforce is more beneficial to the WSRP-F instances for which the cost decreased by up to 75.63% from the solution without additional neighbour workforce. On the other hand, some of the WSRP-D instances did not benefit from the additional workforce, an indication that such instances have the right amount of workforce for the demand. This experimental result suggests that in the WSRP-F instances, the workforce might not be distributed well across regions according to the demanded tasks, which then causes problems when decomposing the problem by regions.

4.5 Comparing to a Genetic Algorithm

We also compare the results obtained by our proposed techniques to the results obtained with a genetic algorithm (GA) [16]. We have chosen a GA because it is a well-known meta-heuristic that has been proven to provide good solutions for both scheduling [18, 25] and routing [28, 19] problems.

We developed a straightforward GA implementation with uniform mutation crossover using a mutation rate $\alpha = \frac{1}{|T|}$ [16], binary tournament selection, population of 100 individuals where the 10% best individuals are kept on the offspring population and time limit as stopping condition. Additionally, to avoid getting stuck in local optima and early convergence, we introduced a reset mechanism, that after 10 generations without improvement, the bottom half (the less fit individuals) of the population is randomly re-generated acting as a population diversity procedure.

Table 5: Instances with understaffed regions. The second and sixth columns show the number of understaffed regions. The third and seventh columns show the average workforce/locations ratio. The fourth and eight columns show average decrease in the objective function value (improvement) obtained from having the additional neighbour workforce.

Instance	#Regions	Ratio	Decrease	Instance	#Regions	Ratio	Decrease
D-01	5	73.76%	41.17%	F-01	13	39.51%	70.57%
D-02	4	75.04%	50.78%	F-02	18	40.92%	66.79%
D-03	5	67.13%	-5.87%	F-03	22	31.87%	67.33%
D-04	5	76.77%	0%	F-04	17	40.09%	70.04%
D-05	4	73.08%	0%	F-05	19	37.97%	56.29%
D-06	5	64.72%	0.10%	F-06	13	44.58%	75.63%
D-07	7	69.31%	7.34%	F-07	23	33.73%	53.57%

#Regions is number of understaffed regions.
Ratio is average of proportion between workers and locations.
Decrease is the average improvement on the objective function value and is calculated as $\frac{(originalObj - addedWorkerObj)}{originalObj}$

Problem-specific knowledge was introduced only at the chromosome encoding. In order to eliminate time conflicts and restrict the exploration to the feasible region of the solution space, we employed an indirect encoding. This encoding is composed of an array of integers such that the indexes correspond to the tasks (note that a task requiring multiple workers would have multiple indexes). The contents of each element in the array represents the k_{th} worker that is skilled to perform the task but that also represents a feasible assignment with no time conflict. If there is no such worker, than the task is left unassigned.

We now compare the results obtained with the proposed decomposition methods and the GA. Because of the stochastic nature of the GA, we ran the GA eight times on each instance and calculated the average solution quality obtained from all these runs. For the instance sets WSRP-A and WSRP-B, we employed the strategies Asc-task and Desc-task without additional neighbour workforce and we set a 15 minutes time limit for the GA. For the larger instance sets WSRP-D and WSRP-F, the additional neighbour workforce process was applied to the decomposition methods and the GA was ran for two hours. Table 6 presents the summary of the experimental results. Overall, we can see the GA shows better performance. This is clearer on the larger instances where the GA gives all best solutions on WSRP-F. However, it is important to mention that on such scenarios the decomposition technique took less computational time than the time limit given to the GA, except for instances WSRP-F-05 and WSRP-F-06 (see Figure 5).

The results obtained for the WSRP-F instances indicate that there is a drawback when the problem is decomposed into too small sub-problems. Our decomposition methods control the maximum size of each sub-problem to be the same for every instance, hence more tasks were left unassigned. As we

Table 6: Comparison of results obtained by the decomposition algorithms and the genetic algorithm on 28 instances.

	Asc-task	Desc-task	GA		N-A-task	N-D-task	GA
A-01	4.33	5.65	**3.92**	D-01	1168.56	**240.44**	337.09
A-02	3.53	4.53	**3.44**	D-02	259.99	**254.44**	324.82
A-03	7.54	10.65	**4.55**	D-03	2,933.84	3,212.32	**434.00**
A-04	**1.54**	3.09	1.75	D-04	**312.44**	418.89	386.26
A-05	2.77	3.54	**2.47**	D-05	408.42	**243.89**	356.47
A-06	**3.55**	3.74	3.70	D-06	307.05	1,410.77	**435.03**
A-07	5.25	4.81	**3.84**	D-07	949.55	753.28	**421.93**
B-01	2.00	**1.79**	1.82	F-01	18,046	22,020	**2,592.56**
B-02	1.94	1.89	**1.79**	F-02	25,712	25,654	**2,652.67**
B-03	2.25	**2.06**	2.27	F-03	50,944	33,686	**1,372.24**
B-04	2.29	**2.21**	2.49	F-04	29,965	34,070	**2,064.93**
B-05	5.60	4.74	**3.38**	F-05	59,363	41,433	**1,092.35**
B-06	2.62	2.52	**2.16**	F-06	20,853	22,038	**1,439.86**
B-07	4.30	4.06	**2.37**	F-07	109,398	66,969	**4,419.73**

Asc-task and **Desc-task** - GDCA ordered by ascending and descending task.
N-A-task and **N-D-task** - decomposition with neighbour workforce ordered by ascending and descending task.
GA - genetic algorithm.

mentioned before, the larger regions were split further into several small sub-problems which share the same workforce. The decomposition can be improved by generating a better cluster based on location and also uniformly distributed. Another challenge is choosing and clustering workforce which would improve the computation efficiency and workforce utilisation.

5 Conclusion and Future Work

A tailored mixed integer programming model for real-world instances of a workforce scheduling and routing problem is presented. The model is constructed by incorporating various constraints from the literature while also adding working region constraints to the formulation. It is usually the case that models in the literature for this type of problem are presented but their solution is provided using alternative methods such as heuristics because solving the model using mathematical exact solvers is computationally challenging. An approach using geographical decomposition with conflict avoidance is proposed here to tackle workforce scheduling and routing problems while still harnessing the power of exact solvers. The proposed decomposition method allows us to tackle real-world sized problems for which finding the overall optimal solution requires extensive computation time. However, the solution quality fluctuates when changing the order to tackle the sub-problems defined by the geographical regions. Exploring all permutation orders to find the one producing the best results is not practical

for larger problems (e.g. more than 6 geographical regions). In this work, six ordering strategies are proposed for obtaining high-quality solutions within acceptable computation time. We also implemented a process of adding neighbour workforce to understaffed regions which helped to obtain better results. We also implemented a standard genetic algorithm to compare the results produced by the proposed decomposition methods. The experimental results indicate that although the decomposition methods produce some best results, it is outperformed by the GA on several instances. However, the decomposition methods are still faster in terms of computation time. Our future research will explore ways to improve the decomposition methods by re-defining sub-problems through automated clustering in order to to find well separated regions that improves the solution procedure.

Acknowledgements

Special thanks to the Development and Promotion for Science and Technology talents project (DPST, Thailand) who providing partial financial support.

References

1. Combined vehicle routing and scheduling with temporal precedence and synchronization constraints. European Journal of Operational Research 191(1), 19 – 31 (2008)
2. Akjiratikarl, C., Yenradee, P., Drake, P.R.: PSO-based algorithm for home care worker scheduling in the UK. Computers & Industrial Engineering 53(4), 559–583
3. Angelis, V.D.: Planning home assistance for AIDS patients in the City of Rome , Italy. Interfaces 28, 75–83 (1998)
4. Barrera, D., Nubia, V., Ciro-Alberto, A.: A network-based approach to the multi-activity combined timetabling and crew scheduling problem: Workforce scheduling for public health policy implementation. Computers & Industrial Engineering 63(4), 802–812 (2012)
5. Benders, J.: Partitioning procedures for solving mixed-variables programming problems. Numerische Mathematik 4(1), 238–252 (1962)
6. Bertels, S., Torsten, F.: A hybrid setup for a hybrid scenario: Combining heuristics for the home health care problem. Computers & Operations Research 33(10), 2866–2890 (2006)
7. Borsani, V., Andrea, M., Giacomo, B., Francesco, S.: A home care scheduling model for human resources. 2006 International Conference on Service Systems and Service Management pp. 449–454
8. Bredstrom, D., Ronnqvist, M.: A branch and price algorithm for the combined vehicle routing and scheduling problem with synchronization constraints. NHH Dept. of Finance & Management Science Discussion Paper No. 2007/7 (February) (2007)
9. Castillo-Salazar, J., Landa-Silva, D., Qu, R.: Workforce scheduling and routing problems: literature survey and computational study. Annals of Operations Research (2014)

10. Castro-Gutierrez, J., Landa-Silva, D., Moreno, P.J.: Nature of real-world multi-objective vehicle routing with evolutionary algorithms. Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on pp. 257–264 (2011)
11. Cordeau, J.F., Stojkovic, G., Soumis, F., Desrosiers, J.: Benders decomposition for simultaneous aircraft routing and crew scheduling. Transportation Science 35(4), 375–388 (2001)
12. Dantzig, G.B., Ramser, J.H.: The truck dispatching problem. Management Science (pre-1986) 6(1) (1959)
13. Dohn, A., Esben, K., Jens, C.: The manpower allocation problem with time windows and job-teaming constraints: A branch-and-price approach. Computers & Operations Research 36(4), 1145–1157 (2009)
14. Eveborn, P., Ronnqvist, M., Einarsdottir, H., Eklund, M., Liden, K., Almroth, M.: Operations research improves quality and efficiency in home care. Interfaces 39(1), 18–34 (2009)
15. Feillet, D.: A tutorial on column generation and branch-and-price for vehicle routing problems. 4OR 8(4), 407–424 (2010), http://dx.doi.org/10.1007/s10288-010-0130-z
16. Goldberg, D.: Genetic Algorithms
17. Hart, E., Sim, K., Urquhart, N.: A real-world employee scheduling and routing application. In: Proceedings of the 2014 Conference Companion on Genetic and Evolutionary Computation Companion. pp. 1239–1242. GECCO Comp '14, ACM, New York, NY, USA (2014)
18. Husbands, P.: Genetic algorithms for scheduling. Intelligence and the Simulation of Behaviour (AISB) Quarterly(89) (1994)
19. Jeon, G., Leep, H.R., Shim, J.Y.: A vehicle routing problem solved by using a hybrid genetic algorithm. Computers & Industrial Engineering 53(4), 680 – 692 (2007)
20. Kergosien, Y., Lente, C., Billaut, J.C.: Home health care problem, an extended multiple travelling salesman problem. In: Proceedings of the 4th multidisciplinary international scheduling conference: Theory and applications (MISTA 2009), Dublin, Ireland. pp. 85–92 (2009)
21. Landa-Silva, D., Wang, Y., Donovan, P., Kendall, G., Way, S.: Hybrid heuristic for multi-carrier transportation plans. In: The 9th Metaheuristics International Conference (MIC 2011). pp. 221–229 (2011)
22. Liu, R., Xie, X., Garaix, T.: Hybridization of tabu search with feasible and infeasible local searches for periodic home health care logistics. Omega 47(0), 17 – 32 (2014)
23. Mankowska, D., Meisel, F., Bierwirth, C.: The home health care routing and scheduling problem with interdependent services. Health Care Management Science 17(1), 15–30 (2014)
24. Mercier, A., Cordeau, J.F., Soumis, F.: A computational study of Benders decomposition for the integrated aircraft routing and crew scheduling problem. Computers & Operations Research 32(6), 1451 – 1476 (2005)
25. Mesghouni, K., Hammadi, S.: Evolutionary algorithms for job shop scheduling. International Journal of Applied Mathematics and Computer Science 2004, 91–103 (2004)
26. Perl, J., Daskin, M.S.: A warehouse location-routing problem. Transportation Research Part B: Methodological 19(5), 381 – 396 (1985)
27. Pillac, V., Gueret, C., Medaglia, A.: On the dynamic technician routing and scheduling problem. In: Proceedings of the 5th International Workshop on Freight Transportation and Logistics (ODYSSEUS 2012). p. id: 194. Mikonos, Greece (May 2012)

28. Potvin, J.Y.: Evolutionary algorithms for vehicle routing. Tech. Rep. 48, CIRRELT (2007)
29. Ralphs, T.K., Galati, M.V.: Decomposition methods for integer programming. Wiley Encyclopedia of Operations Research and Management Science (2010)
30. Rasmussen, M.S., Justesen, T., Dohn, A., Larsen, J.: The home care crew scheduling problem: Preference-based visit clustering and temporal dependencies. European Journal of Operational Research 219(3), 598–610 (2012)
31. Reimann, M., Doerner, K., Hartl, R.F.: D-Ants: Savings based ants divide and conquer the vehicle routing problem. Computers & Operations Research 31(4), 563 – 591 (2004)
32. Trautsamwieser, A., Hirsch, P.: Optimization of daily scheduling for home health care services. Journal of Applied Operational Research 3, 124–136 (2011)
33. Vanderbeck, F.: On Dantzig-Wolfe decomposition in integer programming and ways to perform branching in a branch-and-price algorithm. Operations Research 48(1), 111 (2000)
34. Vanderbeck, F., Wolsey, L.: Reformulation and decomposition of integer programs. In: Junger, M., et al. (eds.) 50 Years of Integer Programming 1958-2008, pp. 431–502. Springer Berlin Heidelberg (2010)