

Assigning and Scheduling Service Visits in a Mixed Urban/Rural Setting

Mark Antunes[§], Vincent Armant*, Kenneth N. Brown*, Daniel Desmond*, Guillaume Escamocher*, Anne-Marie George*, Diarmuid Grimes*, Mike O’Keeffe*, Yiqing Lin[†], Barry O’Sullivan*, Cemalettin Ozturk[‡], Luis Quesada*, Mohamed Siala*, Helmut Simonis* and Nic Wilson*

*Insight Centre for Data Analytics, University College Cork, Cork, Ireland

[†]UTRC-I, Cork, Ireland

[‡]UTRC, East Hartford, CT, USA

[§]United Technologies Corporation, Winnipeg, Manitoba, Canada

Abstract—In this paper¹ we describe a complex optimization application arising in maintenance scheduling, developed in close collaboration with an industrial partner. We have to plan and schedule preventive and corrective maintenance activities at customer sites by a group of traveling repair technicians. A specific property of the problem considered here is a mix of customers in both urban centers and rural areas. This means that travel times between customers must be considered when balancing overall workload for each agent. We discuss a problem decomposition compatible with current management practice, describe different solvers for the individual problem steps, and show results on real-world data from the industrial partner.

I. INTRODUCTION

In many industries, the after-sales service and maintenance of installed products is an important part of the overall business [4]. We here consider a problem where service technicians have to visit customer sites to test, maintain and repair machinery permanently installed at the customer location. Different types of products have varying service requirements, both in time required and in frequency of visits. During a working day, technicians visit multiple customer sites to perform required work. The travel between customer locations is part of the working time, but is not productive, so a good schedule minimizes travel, while performing required tests and repairs at the right time point. One special aspect of the problem studied in this paper is a mix of urban and rural customer sites. While travel times inside cities can be quite short, visits to outlying rural areas may require a carefully planned tour of multiple days.

We developed this application in close collaboration with our industry partner, in a team comprising of domain experts, management and end users, as well as optimization experts. The constraints and overall solution approach were defined specifically for the client. Familiarity with individual installations and with specific customers is very important to maintain service quality, so an existing business rule is that all work at a specific customer site should be performed by the

same technician, if at all possible. In our approach we have to follow this principle, which naturally leads to a decomposition of the problem into an assignment and a scheduling phase. Technicians are trained to work on certain types of equipment, and need additional training to deal with unfamiliar equipment. They may also have preferences either to work mainly inside their home town during regular office hours, or spending more time on the road visiting outlying locations. While travel from their home to the first local customer site and from the last local customer of the day back home is not considered working time for the technicians, they do have preferences about starting and ending locations, or places for their lunch break.

In our model we currently consider deterministic task duration and travel times. We describe how to balance service regions for each technician, based on the expected yearly workload of planned and unplanned operations. We use a pre-clustering of customer locations to reduce long distance travel, while at the same time reducing problem complexity. In order to reduce travel times within each service region, we use our optimization model iteratively, minimizing the diameter of each area, while balancing the workload between technicians.

The mobile workforce scheduling problem combines aspects of vehicle routing[11] with personnel scheduling [7]. It is a well studied problem, see [4] for a survey. Some studies focus on a specific industry, for example electricity [9], water supply [3], [28], copier repair [26] or elevator maintenance [20], [2], [29]. Others consider specific solution techniques, like large neighborhood search [5], [17] or genetic algorithms [22]. Heching and Hooker suggest the use of Benders Decomposition to solve a related problem of home-care scheduling [12], which was also considered in [23], [13], but solved using meta-heuristics.

Mobile workforce scheduling can have a large impact on operational cost: A tool for the mobile workforce management of BT Telecom is described in [18], [19]. It combines constraint programming with optimization to handle a multi-skilled workforce of 20,000 technicians, with up to 150,000 tasks per day. A different IT platform for dispatching UPS drivers is described in [14]. The tool provides optimized routes

¹This material is based upon works supported by United Technologies Corporation under UCC Collaboration Project and by Science Foundation Ireland under Grant No. 12/RC/2289 which is co-funded under the European Regional Development Fund.

for all 55,000 drivers collecting and delivering packages for UPS in North America. It is expected to save \$300M to \$400M per year in operational cost.

The technical contributions of our paper are the combination of planned and unplanned work, and their required travel, in a yearly capacity model, reducing problem complexity by aggregation of customer sites, and the combination of one-day and multi-day tours in the scheduling solution, using a variety of tools and algorithms.

The paper is structured as follows: We begin with a description of the overall problem decomposition (Section II). We then describe the different stages of our solver, first a clustering method (Section III), then a discussion of Route Generation (Section IV), a refinement of Aggregated Route Generation in Section V, and finally, the scheduling model in Section VI. This is followed by an evaluation based on real-world end-user data in Section VII.

II. OVERALL DECOMPOSITION

In a complete workflow, an automated procedure should produce the detailed daily assignment of technicians to sites, considering the long-term workload, and achieving the least-cost solutions. But solving this complete problem in one step would be very challenging, so we propose a decomposition that follows existing management practice, and retains control for the different stakeholders. We split the scheduling into four phases, as shown in Figure 1.

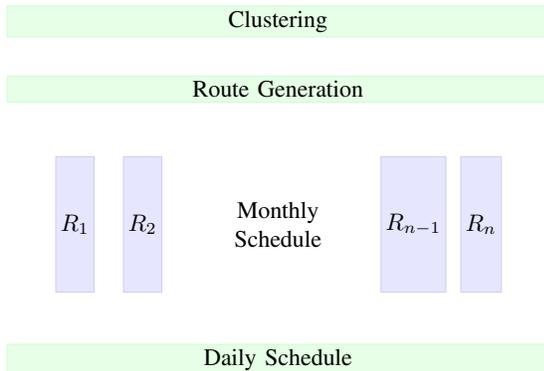


Fig. 1: Overall Problem Decomposition

Clustering Visits. As a first step, sites in close proximity should be clustered together, to make sure that we are visiting these locations at the same time.

Route Generation. The second step is to partition sites into sets (called routes in the industry), to produce a balanced assignment of sites to technicians, which respects the constraints and preferences involved, but also allows for schedules that require the least amount of travel. A core feature is stability of the assignment over time, as familiarity of technicians with their assigned sites improves service quality, and often increases customer satisfaction.

Monthly Schedule. In the next step, we find monthly schedules for each technician, considering all manda-

tory and optional work that should be performed in this time period. For each day, a tour visiting one or multiple sites should be defined, that satisfies the working time rules, yet minimizes travel time. As activities have due dates, we also must avoid scheduling visits too early (lost time), and too late (penalties). Each technician is scheduled independently, considering the work assigned to them in the route assignment.

Daily Rescheduling. At the day of operation, the predefined monthly schedule may require modifications. There may be unfinished work from previous days, or faults have been reported by customers, that require an unplanned repair visit. The daily scheduler therefore should reconsider the existing assignment, and modify it accordingly. As part of this step, work may be shifted between technicians, in order to handle urgent requests in a more timely manner. The module is similar to the Monthly Scheduler, but we do not cover the details of this element in the current paper.

We will now describe each of the modules in turn.

III. CLUSTERING

The clustering operation aims at grouping sites together that are close to each other to get a better estimate of the travel required. The intention is to visit all nodes in a cluster at the same time, requiring only one trip instead of multiple trips to each site. Note that this can work only for planned visits; unplanned repairs occur independently from each other, and do require a separate trip to each site.

Our clustering method is based on a simple graph-based representation. We build an undirected graph consisting of one node for each site. Two nodes are connected if either

- they are located in the same city
- or their distance is less than d km
- or the user has entered a connection manually

We find all connected components in this graph, and group the sites based on their component number. The parameter d gives us control over how easily items should be placed together.

IV. ROUTE GENERATION

The route generation phase is based on the generated components, and tries to partition the sites into sets (called routes), that have a similar yearly workload. As input components we use the clusters obtained in the previous step. At the core this is a bin packing problem where we want to find the correct bin size to pack all items (sites) into a fixed number of bins (routes), thereby balancing the workload assigned. We extend this core concept with an estimate of the amount of travel that will be assigned to each route, balancing not only the working time, but the total of work and travel time. We choose to use an off-the-shelf MIP solver to solve this problem instead of a bespoke CP solution [25], [24], [21], as there are few large items to be placed, resulting in poor constraint propagation.

We propose three models for this problem.

- The core model (described in Section IV-A) assigns individual sites to routes, and handles the main constraints and preference requirements.
- The virtual route center model (Section IV-B) also considers the distance between sites in the same route, and tries to find a compromise between balanced workload and small intra-route travel times. We use the concept of a virtual route center to group sites in each route more closely together. This method uses an iterative process that stops when the virtual centers stabilize. In each iteration, a version of the core MIP model is solved. The solution may be only a local minimum, so multiple runs with different starting conditions can be used.
- The aggregated route generation (Section V) first assigns (fractions of) components to each route, allowing a split of large components between multiple routes. Once the best overall split is known, it then assigns sites in the split components proportionally to all routes involved.

A. Core Model

We consider a problem with a set of sites B , which should be assigned to routes R from a set of depots D . The set of components C forms a partition of B . Lower case symbols b, c, d, r denote members of these sets. In addition, we use the following constants:

t_{ab}	travel time between two locations a and b
h_r	home location for route r
$l_{b/c/d}$	location of site b , component c , or depot d
$v_{b/c}$	yearly planned visits to site b or component c ; $v_c = \max_{b \in C} v_b$
u_b	yearly unplanned visits to site b , using a forecast based on historical data
n_d	number of routes allocated to depot d
w_{rb}	yearly work for site b when performed by route r , includes travel time for unplanned, but not for planned visits
$q_b^{p/u}$	average time needed on a planned (p) or unplanned (u) visit to site b
c_b	component to which site b belongs
$p_{r/b}$	administrative region to which route r or site b belongs
$d_{r/b}$	depot to which route r or site b belongs in current assignment
m_b	route to which site b is assigned in current, manual assignment
s_{rb}	true if route r is skilled to perform work in site b

We define the yearly work w_{rb} for site b assigned to route r as the total of all unplanned visits to the location, considering the working time and an estimate of the travel time, and the work performed in the site during all planned visits.

$$w_{rb} := v_b q_b^p + u_b (t_{h_r, l_b} + q_b^u) \quad (1)$$

Note that this is an approximation, as unplanned trips may start during the day in a location different from the home location.

We allow the following optional settings to enable/disable certain constraints and preferences as follows:

respectProvince. Work in each administrative area should only be assigned to technicians that are registered to work in that administrative area. If this is not enforced, then technicians missing the accreditation must perform the required training/exams, at extra cost.

respectDepot. Work currently assigned to a depot should not be assigned to workers from outside that depot. Any changes in depot assignment need agreements of multiple stakeholders, and are really management decisions. It can be useful to remove this restriction to see if any changes of the assignment would improve other cost factors, and which sites would be reassigned.

keepAssignment_b. Site b should stay with the current assignment m_b . This may be required by the technician or the owner/operator of the site, and would be considered only for some sites.

enforceSkills. The equipment in site b requires training that not all technicians may have. When enforcing this constraint, the current training is taken into account; If it is not enforced, the optimal solution may require additional training for some technicians, but may also allow for less travel.

We define a Boolean function `infeasible` which, based on the options and constant values, decides if site b can be assigned to route r .

$$\text{infeasible}(r, b) := p_r \neq p_b \wedge \text{respectProvince} \quad (2)$$

$$\vee d_r \neq d_b \wedge \text{respectDepot} \quad (3)$$

$$\vee r \neq m_b \wedge \text{keepAssignment}_b \quad (4)$$

$$\vee \neg s_{rb} \wedge \text{enforceSkills} \quad (5)$$

We use three sets of variables in our core model. The Boolean variables x_{rb} denote if site b is assigned to route r . The Boolean variables y_{rc} denote if component c is visited by route r ; there may be multiple routes visiting a component. The non-negative variables z_d denote an upper bound on the work load of all routes in depot d . They are used to balance the workload between the routes of each depot.

The objective function (6) of the model consists of three parts. The first is the sum of the upper work load bounds for each depot, used to balance the workloads of the routes in the depot. We weight each term with the number of routes in the depot to give more importance to large depots. The second term considers the travel time used to visit the components for regular (scheduled) service. As we plan to perform all work in a component at the same time, we only have to consider this travel once for the component. The third cost element is the sum of the assigned workload for each route, which consists of the working at the site (both planned and unplanned), and the travel time required for unplanned visits.

Note that all cost terms are expressed as time durations, making them directly comparable.

$$\min \sum_{d \in D} z_d n_d + \sum_{r \in R} \sum_{c \in C} y_{rc} t_{h_r l_c} v_c + \sum_{r \in R} \sum_{b \in B} x_{rb} w_{rb} \quad (6)$$

$$\forall r \in R \forall b \in B : x_{rb} \in \{0, 1\} \quad (7)$$

$$\forall r \in R \forall c \in C : y_{rc} \in \{0, 1\} \quad (8)$$

$$\forall d \in D : z_d \geq 0 \quad (9)$$

$$\forall r \in R \forall b \in B : x_{rb} = 0 \text{ if } \text{infeasible}(r, b) \quad (10)$$

$$\forall r \in R \forall b \in B : x_{rb} \leq y_{rc_b} \quad (11)$$

$$\forall b \in B : \sum_{r \in R} x_{rb} = 1 \quad (12)$$

$$\forall r \in R : \sum_{b \in B} x_{rb} w_{rb} + \sum_{c \in C} y_{rc} t_{h_r l_c} v_c \leq z_{d_r} \quad (13)$$

We cannot assign a site to a route if that assignment is infeasible given the current options (Eq 10). If a site in a component is assigned to a route, the route must visit the component (Eq 11). Each site must be assigned to exactly one route (Eq 12). The work load assigned to each route in a depot is limited by the upper workload bound for the depot (Eq 13).

B. Virtual route centers, minimizing intra route travel

One of the problems with the basic model is that it does not consider any travel that may be required between sites if the technician travels to multiple locations in the same trip. Trying to express such constraints directly in the model was not successful. Instead, we introduce the concept of a virtual route center and introduce another cost term, the sum of the distances between the virtual route center and all assigned sites. We iteratively update the centers to the average of the assigned sites, and repeat until the virtual centers stabilize. For this we need a virtual center location of route r at iteration k , denoted by l_r^k . Iteration 0 corresponds to an initial assignment, that can be chosen randomly, or can be based on the existing, manual assignment.

The objective function is based on (6), adding a term minimizing the distances from the virtual route centers to the assigned sites.

$$\min \sum_{d \in D} z_d n_d + \sum_{r \in R} \sum_{c \in C} y_{rc} t_{h_r l_c} v_c + \sum_{r \in R} \sum_{b \in B} (x_{rb} w_{rb} + t_{l_r^k l_b} v_b) \quad (14)$$

The overall solver is described by the procedure `findAssignment` in Figure 2, which returns true if a solution is found. The notation x^* is used to refer to the value of variable x in the optimal solution of the MIP model. The MIP model itself consists of the objective (14) together with constraints (7) to (13). We determined the values for the parameters $limit=20$ and $cutoff=0.05$ empirically.

V. AGGREGATED ROUTE GENERATION

One of the challenges of the model in Section IV-B is that we need to solve a medium-sized MIP problem at each step of multiple runs. In an interactive problem solving setting, the run-time of the solver is no longer acceptable. In order

```

boolean findAssignment() {
    status := true; k := 0; for (r:R)  $l_r^k := \frac{\sum_{b \in B: m_b=r} l_b}{|\{b \in B: m_b=r\}|}$ ;
    do {
        k++;
        status &= solveMIP();
        for (r:R)  $l_r^k := \frac{\sum_{b \in B} x_{rb}^* l_b}{\sum_{b \in B} x_{rb}^*}$ ; % recomputeCenters(k);
    } while (k < limit &&  $\sum_{r \in R} |l_r^k - l_r^{k-1}| > cutoff$ );
    return status;
}

```

Fig. 2: Pseudocode for virtual route center optimization

to reduce the size of the model, we introduce the decomposition shown in Figure 3. We first solve a partial component assignment, where we decide which components are visited by which route, and how much of their workload is performed by each route. If a component is visited only by a single route, then all sites in the component are handled by the route. If the work is split, we need a second stage solver (shown in red in Figure 3) which handles the assignment of individual sites for that component.

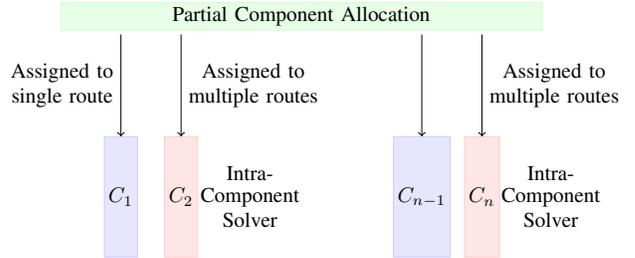


Fig. 3: Aggregated Route Generation Model, Overall Design

A. Partial Component Allocation

The first step of the aggregated solver is the (partial) assignment of components to routes. We cannot simply assign each component to a single route since

- this would reduce flexibility too much and might remove better solutions
- some components are too large to assign to a single route, and must be split.

We need a few more symbols for constants for the partial component assignment as follows:

$$f_c \quad \text{smallest fractional assignment of component } c$$

$$w_{c/b} \quad \text{yearly workload for site } b \text{ or component } c; w_c = \sum_{b \in c} w_b$$

We define the workload for a component to be the sum of the workloads of its sites. We redefine the workload of a site as the sum of the planned and unplanned work, plus the unplanned travel (using the depot center as starting point). This slightly redefines the workload from the w_{rb} definition used in Section IV-A to remove dependency on the assigned route.

$$w_b := u_b(t_{l_d l_b} + q_b^u) + v_b q_b^p \quad (15)$$

If we want to keep some site assigned to a specific route, we have to make sure that the component is assigned to the route. For this we introduce a function `assigned`:

$$\text{assigned}(r, c) := \exists b \in B \text{ s.t.} \\ c_b = c \wedge m_b = r \wedge \text{keepAssignment}_b \quad (16)$$

The overall solver uses virtual route centers, and the overall iterative process defined in Figure 2. We now define the MIP model that is solved at each iteration k .

The objective function in our aggregated model is the sum of the workload bounds for each depot, the assigned work to each route, and the distances from the virtual route centers to the assigned components. The new model is as follows:

$$\min \sum_{d \in D} z_d n_d + \sum_{r \in R} u_r + \sum_{r \in R} \sum_{c \in C} x_{rc} t_{rl_c}^k \quad (17)$$

$$\forall r \in R \forall c \in C : x_{rc} \in \{0, 1\} \quad (18)$$

$$\forall r \in R \forall c \in C : y_{rc} \in [0, 1] \quad (19)$$

$$\forall r \in R : u_r \geq 0 \quad (20)$$

$$\forall d \in D : z_d \geq 0 \quad (21)$$

$$\forall r \in R \forall c \in C : y_{rc} \leq x_{rc} \quad (22)$$

$$\forall r \in R \forall c \in C : y_{rc} \geq x_{rc} f_c \quad (23)$$

$$\forall c \in C : \sum_{r \in R} y_{rc} = 1 \quad (24)$$

$$\forall r \in R \forall c \in C : x_{rc} = 1 \text{ if } \text{assigned}(r, c) \quad (25)$$

$$\forall r \in R : u_r = \sum_{c \in C} x_{rc} v_c t_{h_r, l_c} + \sum_{c \in C} y_{rc} w_c \quad (26)$$

$$\forall r \in R : u_r \leq z_d \quad (27)$$

The integer variable x_{rc} encodes whether or not we visit component c as part of route r . We use fractional values y_{rc} to describe how much of the total work in component c is assigned to route r . We then introduce non-negative, continuous variables u_r , to describe the work assigned to route r . The non-negative variable z_d gives an upper bound on the work load assigned to each route in depot d . If we assign some work in the component to a route, then we must visit the component. If, on the other hand, we do not visit the component, then the assigned work fraction must be zero (Eq 22). If we visit a component, then we must assign a certain minimum amount of work to the route (Eq 23). The smallest value could, for example, correspond to one site. The work in a component must be split between the assigned routes, so their percentages must add up to one (Eq 24). If we want to pre-assign a site in a component, we also have to pre-assign the component to which it belongs (Eq 25). The work load for a route consists of the regular travel time to visit all assigned components and the proportion of the workload of the visited components (Eq 26). The workload of each route is bounded by the work load limit of the depot (Eq 27). If a component is assigned to a single route, then all sites in that component are automatically assigned to the route.

B. Intra-Component Solver

For each component c that is split between two or more routes, we have to solve the detailed site assignment inside the component. We introduce the work f_r that should be given to route r in component c , based on the fractional value assigned in the optimal solution to (17), and the total workload of the component:

$$f_r := y_{rc}^* w_c \quad (28)$$

The objective is to group sites in the component assigned to the same route together, again using the virtual route centers.

$$\min \sum_{r \in R} \sum_{b \in B} x_{rb} t_{rl_b}^k \quad (29)$$

$$\forall r \in R \forall b \in c : x_{rb} \in \{0, 1\} \quad (30)$$

$$\forall b \in c : \sum_{r \in R} x_{rb} = 1 \quad (31)$$

$$\forall r \in R : f_r(1 - \epsilon) \leq \sum_{b \in B} x_{rb} w_b \leq f_r(1 + \epsilon) \quad (32)$$

$$\forall r \in R \forall b \in c : x_{rb} = 0 \text{ if } \text{infeasible}(r, b) \quad (33)$$

The decision variables x_{rb} state whether site b in component c is handled by route r . Each site must be assigned to exactly one route (Eq 31). The workload assigned to each route is determined by the fractional values obtained from the optimal solution of the MIP in Section V-A. We allow an ϵ variation of the workload to avoid infeasible sub-problems (Eq 32). We restrict the assignment of sites to feasible routes (Eq 33). If no solution can be found, we either have to increase ϵ or have to introduce more constraints in the partial assignment model. This MIP is also run inside a virtual route center loop, as defined in Figure 2.

VI. MONTHLY SCHEDULING

In this part of the problem solving process, we consider all activities that should be performed by one technician within one month. Some of those activities are pre-defined (monthly training sessions, planned holidays), whereas the visits to customer sites can be moved in time. Each planned visit has a due date on which it should ideally be completed. If it is performed earlier, then we lose some efficiency due to a shortened service interval; if the activity is performed after the due date, increasing penalties for non-performance must be paid. We can impose hard time-windows around the due date during which tasks must be performed, at the risk of creating infeasible problems.

The working hours for the technicians usually are office hours, 08:00 to 17:00, with one hour lunch breaks. To visit remote locations, multi-day trips may be required, which incur overnight-stay costs, and additional allowances. If the technician is working in their home location, then travel from their home to the first customer of the day, and travel from the last customer back home is not considered as working time. For multi-day trips, that travel is considered working time.

To solve the monthly scheduling problem, we use a decomposition into two stages. In a first stage, we create feasible

configurations, sequences of visits to customers that satisfy the work rules and fill a working day. Ideally, we are able to produce all relevant configurations initially, otherwise a column generation [8] scheme is required, where we add missing configurations during the second stage. Alternatively, we could use a portfolio approach to improve performance of the set partitioning solver [15].

The second stage selects a single configuration for each day, so that the demand is satisfied, and early and late penalties are minimized. This is a version of a set-partitioning problem [10]. This use of configuration generation and set partitioning is well known for vehicle routing, going back at least to [6], [16], [27], we extend it here to also include the temporal, scheduling aspect.

A. Configuration Generation

A configuration is a sequence of activities with their start times, that describes a potential day of work for the technician. We distinguish single-day and multi-day configurations. A single day configuration will be performed within the home town of the technician. Work starts at 08:00 at the first site visited, and ends at the end of work at the last location visited. Travel between site locations counts as working time; traveling from and to home is not included.

Any trip that covers one or more activities, and allows enough time for travel to fit into the working time would be a valid configuration. We produce a set of possible configurations by dynamic programming [1], starting with configurations consisting of single activities, and travel tasks from and to home. Given a configuration, we then try to insert activities into the configuration, replacing one travel task with travel to the location of the inserted activity, then performing the added activity, and finally the travel to continue the trip. We only have to allow one insertion, for which we minimize the resulting travel time. We stop the process when we can no longer add new activities without exceeding the working time. We can reject many configurations that, for example, visit some component multiple times, or do not contain enough work.

To further improve quality, we can re-optimize the configuration after each step by solving a small traveling salesman problem, possibly re-ordering the activities to reduce total travel time. At the moment this step is skipped to enhance interactive performance.

B. Configuration Selection

We use the set of all activities A , the set of generated configurations C , and the set of days D in the planning period. We also use a set E of excluded days, where the person is assigned to other activities (training, holidays), and cannot be scheduled. This might also include the first days of the current scheduling period, if the technician finishes a multi-day trip started in the previous planning period.

We require the following constants:

p_{ca} Boolean which states if configuration c contains activity a

n_a cost of not performing operation a in the planning period
 q_{cd} cost of running configuration c on day d
 m_c duration (in days) of configuration c , used to handle multi-day trips

$$\min \sum_{c \in C} \sum_{d \in D} v_{cd} q_{cd} + \sum_{a \in A} u_a n_a \quad (34)$$

$$\forall c \in C \forall d \in D : v_{cd} \in \{0, 1\} \quad (35)$$

$$\forall a \in A : u_a \in \{0, 1\} \quad (36)$$

$$\forall a \in A : \sum_{c \in C} \sum_{p_{ca}} v_{cd} + u_a = 1 \quad (37)$$

$$\forall d \in D : \sum_{c \in C} \sum_{\substack{d' \in D \\ d' \leq d \\ d-d' < m_c}} v_{cd'} \leq 1 \quad (38)$$

$$\forall c \in C : \sum_{d \in D} v_{cd} \leq 1 \quad (39)$$

$$\forall d \in E \forall c \in C \forall \substack{d' \in D \\ d' \leq d \\ d-d' < m_c} : v_{cd'} = 0 \quad (40)$$

We use Boolean decision variables v_{cd} to state that we schedule (or start, for multi-day trips) configuration c on day d . We may not be able to handle all activities in the given time period. Instead of just reporting an infeasible problem, we use Boolean variables u_a to postpone activities until the next time period. The objective (34) is to find the best selection of configurations to run on each day, so that their cost (earliness and lateness), together with the penalties for non-performed activities, is minimized. Each activity must be either performed once in one configuration on one day, or must be postponed until after the current planning period (Eq 37). Finally, we state that on each day d we can run at most one configuration, allowing for a case with so little work that we can have a free day as well. To handle multi-day configurations, we consider all starting days d' for a configuration c so that the configuration is still running on day d (Eq 38). We also state that each configuration can be used at most once (Eq 39). For all excluded days in set E , we cannot be running any configurations, including multi-day trips starting earlier, but not finished yet on day d (Eq 40).

VII. EXPERIMENTS

For the experiments we use a real-world data-set provided by an end-user, which includes 5 depots in 3 administrative regions, handling 18 routes for 1,252 customer sites. Over the one year planning horizon, we have to consider 79,454 activities for assignment and scheduling. All experiments were run on a 2.9GHz Intel i7 laptop with 64Gb of memory, using Cplex 12.63 as the MIP solver.

Figure 4(a) shows the number of clusters found as a function of the distance d between neighbors selected. As d increases, the number of clusters decreases rapidly. Enforcing the clustering together for sites in the same city (dotted) only

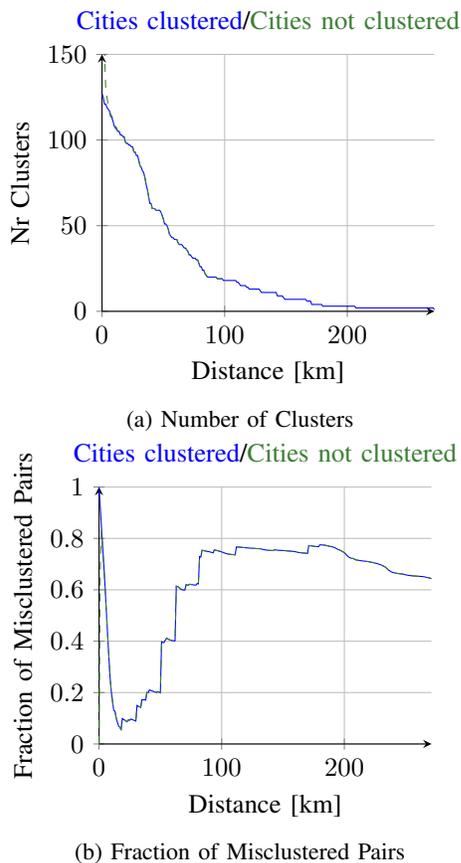


Fig. 4: Connected Component Based Clustering: Impact of Parameter d

has an impact for small values of $d < 10km$. We use a value of $d = 20km$ to generate close to 100 clusters in our experiments.

In the connected component based clustering, some sites may be in the same component, even though their distance is greater than distance limit d , as they are connected through a chain of other sites, each at a distance of less than d . Figure 4(b) considers the fraction of all pairs of sites that are in the same cluster, but whose distance is greater than d . We see that there is a local minimum for $d = 18km$, where less than 6% of clustered pairs exceed the distance limit. This further justifies our choice of $d = 20km$ in our experiments.

Table I compares the three models presented in Sections IV and V on the same problem instance. We report the amount of work scheduled, the total travel time estimated, the standard deviation of the total time allocated to each route, the average time of a trip within a route, the number of iterations of the model run, and the total run-time, marking with TO when a timeout of 300 seconds was limiting the search. All three models allocate the fixed work in all customer sites, while the travel time and the balance between the route workloads is quite similar. As the core model ignores the travel within each route, its average trip duration is more than twice the value of the moving route center model. The average trip duration of the aggregated model is slightly worse than for the moving

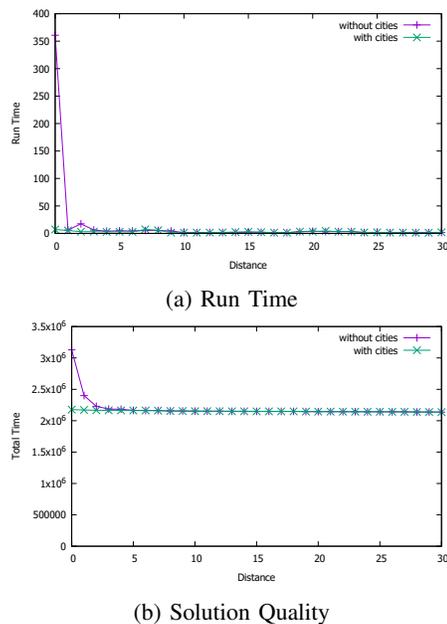


Fig. 5: Impact of Clustering Distance (X-axis) on Results of the Aggregated Route Generation Algorithm

route centers due to the independent execution of the intra-component models, but the aggregated model converges much more rapidly, while solving each sub-problem to optimality.

TABLE I: Comparison of Models

Model	Total Work (min)	Total Travel (min)	Route Std Dev (min)	Av Trip (min)	Iterations	Run-Time (s)
Core	1,647,834	499,799	5,381.56	242.04	-	TO 301.85
Moving Center	1,647,834	500,219	6,073.00	110.17	9	TO 359.92
Aggregated	1,647,834	497,123	5,629.13	136.21	5	6.09

Figure 5 considers the impact of varying the clustering distance between 1 and 30km on the aggregated route generation algorithm, (a) shows the run-time, (b) shows solution quality. We again distinguish the case where we also automatically cluster cities (shown in blue, with x markers) or where we only use the distance to clustering locations (red, with $+$ markers). We see that both run time and solution quality vary only very little with the choice of parameter, with the exception of values smaller than 5km, where, without clustering cities, many small clusters lead to larger problems to solve, while over-estimating the travel times. Similar results are obtained when using other clustering methods.

We evaluated the Schedule Model with the results of the Aggregated Route Generation model, by varying the time window in which a task can be scheduled before and after its due-date. As the time window increases, more and more tasks must be considered together in a configuration, therefore the number of configurations increases rapidly with the parameter value. Table II shows results for different time window sizes. We list the run-time (in seconds), the value of the objective function, the distance traveled for all routes, and the number of configurations generated.

TABLE II: Schedule Results for One Month Based on Time Window Parameter

Time Window	Run-Time (s)	Cost	Travel (km)	Configurations
4	62.38	100,006.17	25,621.99	71,050
5	77.68	98,429.33	24,724.10	97,913
6	172.73	94,463.00	22,365.61	194,697
7	505.93	91,480.83	20,568.48	562,268
8	2,273.85	89,372.17	19,299.16	1,192,461

We see that the run time required increases with the number of configurations generated, while both total cost and distance traveled decrease with increasing time window. On the other hand, an increase in the time window increases penalties for both early and late performance of tasks.

VIII. CONCLUSIONS

In this paper we have shown a complex assignment and scheduling problem from industry that has been solved by decomposition into sub-problems following current business practice. Our approach uses a variety of algorithms and optimization methods, from inter-site travel time through route planning on OSM data, clustering, partial component allocation inter-component assignment, configuration generation using dynamic programming with travel and task times, and finally scheduling via set partitioning. Future work will investigate more stochastic elements of the problem (especially travel times and task duration), as well as integrating more, potentially conflicting, preferences of the various stakeholders.

REFERENCES

[1] R. Bellman. The theory of dynamic programming. *Bull. Amer. Math. Soc.*, 60(6):503–515, 11 1954.

[2] F. Blakeley, B. Argüello, B. Cao, W. Hall, and J. Knolmayer. Optimizing periodic maintenance operations for Schindler elevator corporation. *Interfaces*, 33(1):67–79, 2003.

[3] N. Bostel, P. Dejax, P. Guez, and F. Tricoire. Multiperiod planning and routing on a rolling horizon for field force optimization logistics. In B. Golden, S. Raghavan, and E. A. Wasil, editors, *The vehicle routing problem: latest advances and new challenges*, pages 503–525. Springer, New York, 2008.

[4] J. A. Castillo-Salazar, D. Landa-Silva, and R. Qu. Workforce scheduling and routing problems: literature survey and computational study. *Annals of Operations Research*, 239(1):39–67, 2016.

[5] J.-F. Cordeau, G. Laporte, F. Pasin, and S. Ropke. Scheduling technicians and tasks in a telecommunications company. *Journal of Scheduling*, 13(4):393–409, 2010.

[6] F. H. Cullen, J. J. Jarvis, and H. D. Ratliff. Set partitioning based heuristics for interactive routing. *Networks*, 11(2):125–143, 1981.

[7] J. V. den Bergh, J. Belin, P. D. Bruecker, E. Demeulemeester, and L. D. Boeck. Personnel scheduling: A literature review. *European Journal of Operational Research*, 226(3):367 – 385, 2013.

[8] L. Ford and D. Fulkerson. A suggested computation for maximal multicommodity network flows. *Management Science*, 5:97101, 1958.

[9] A. Froger, M. Gendreau, J. E. Mendoza, E. Pinson, and L.-M. Rousseau. Maintenance scheduling in the electricity industry: A literature review. *European Journal of Operational Research*, 251(3):695 – 706, 2016.

[10] R. S. Garfinkel and G. L. Nemhauser. The set-partitioning problem: Set covering with equality constraints. *Oper. Res.*, 17(5):848–856, Oct. 1969.

[11] B. Golden, S. Raghavan, and E. A. Wasil, editors. *The vehicle routing problem: latest advances and new challenges*. Springer, New York, 2008.

[12] A. R. Heching and J. N. Hooker. Scheduling home hospice care with logic-based Benders decomposition. In C. Quimper, editor, *Integration of AI and OR Techniques in Constraint Programming - 13th International Conference, CPAIOR 2016, Banff, AB, Canada, May 29 - June 1, 2016, Proceedings*, volume 9676 of *Lecture Notes in Computer Science*, pages 187–197. Springer, 2016.

[13] G. Hiermann, M. Prandtstetter, A. Rendl, J. Puchinger, and G. R. Raidl. Metaheuristics for solving a multimodal home-healthcare scheduling problem. *CEJOR*, 23(1):89–113, 2015.

[14] C. Holland, J. Levis, R. Nuggehalli, B. Santilli, and J. Winters. UPS optimizes delivery routes. *Interfaces*, 47(1):8–23, 2017.

[15] S. Kadioglu, Y. Malitsky, and M. Sellmann. Non-model-based search guidance for set partitioning problems. In J. Hoffmann and B. Selman, editors, *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada*. AAAI Press, 2012.

[16] J. P. Kelly and J. Xu. A set-partitioning-based heuristic for the vehicle routing problem. *INFORMS Journal on Computing*, 11(2):161–172, 1999.

[17] A. A. Kovacs, S. N. Parragh, K. F. Doerner, and R. F. Hartl. Adaptive large neighborhood search for service technician routing and scheduling problems. *Journal of Scheduling*, 15(5):579–600, 2012.

[18] D. Lesaint, C. Voudouris, and N. Azarmi. Dynamic workforce scheduling for British Telecommunications plc. *Interfaces*, 30(1):45–56, 2000.

[19] D. Lesaint, C. Voudouris, N. Azarmi, and B. Laithwaite. Dynamic workforce management. In *IEE Colloquium on AI for Network Management Systems*, pages 1/1–1/5, Apr 1997.

[20] Y. Lin, A. Hsu, and R. Rajamani. A simulation model for field service with condition-based maintenance. In *Proceedings of the Winter Simulation Conference*, volume 2, pages 1885–1890 vol.2, Dec 2002.

[21] F. Pelsner, P. Schaus, and J. Régis. Revisiting the cardinality reasoning for binpacking constraint. In C. Schulte, editor, *Principles and Practice of Constraint Programming - 19th International Conference, CP 2013, Uppsala, Sweden, September 16-20, 2013. Proceedings*, volume 8124 of *Lecture Notes in Computer Science*, pages 578–586. Springer, 2013.

[22] S. Remde, P. Cowling, K. Dahal, and N. Colledge. Exact/heuristic hybrids using rVNS and hyperheuristics for workforce scheduling. In C. Cotta and J. van Hemert, editors, *Evolutionary Computation in Combinatorial Optimization: 7th European Conference, EvoCOP 2007, Valencia, Spain, April 11-13, 2007. Proceedings*, pages 188–197. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[23] A. Rendl, M. Prandtstetter, G. Hiermann, J. Puchinger, and G. R. Raidl. Hybrid heuristics for multimodal homecare scheduling. In N. Beldiceanu, N. Jussien, and E. Pinson, editors, *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems - 9th International Conference, CPAIOR 2012, Nantes, France, May 28 - June 1, 2012. Proceedings*, volume 7298 of *Lecture Notes in Computer Science*, pages 339–355. Springer, 2012.

[24] P. Schaus, J. Régis, R. V. Schaeren, W. Dullaert, and B. Raa. Cardinality reasoning for bin-packing constraint: Application to a tank allocation problem. In M. Milano, editor, *Principles and Practice of Constraint Programming - 18th International Conference, CP 2012, Québec City, QC, Canada, October 8-12, 2012. Proceedings*, volume 7514 of *Lecture Notes in Computer Science*, pages 815–822. Springer, 2012.

[25] P. Shaw. A constraint for bin packing. In M. Wallace, editor, *Principles and Practice of Constraint Programming - CP 2004, 10th International Conference, CP 2004, Toronto, Canada, September 27 - October 1, 2004. Proceedings*, volume 3258 of *Lecture Notes in Computer Science*, pages 648–662. Springer, 2004.

[26] S. Souyris, C. E. Cortés, F. Ordóñez, and A. Weintraub. A robust optimization approach to dispatching technicians under stochastic service times. *Optimization Letters*, 7(7):1549–1568, 2013.

[27] É. D. Taillard. A heuristic column generation method for the heterogeneous fleet VRP. *RAIRO - Operations Research*, 33(1):1–14, 1999.

[28] F. Tricoire. *Vehicle and personnel routing optimization in the service sector: application to water distribution and treatment*. Theses, Université de Nantes, Feb. 2006. Thèse réalisée en convention CIFRE avec Veolia Eau.

[29] S. Velicu, I. Paunescu, and M. D. Paraschiv. Transition from the classic service operations to the decision support in maintenance activity. *Proceedings in Manufacturing Systems*, 10(1):45, 2015.