Portuguese Study Groups' Reports

Report on
Service scheduling in garden maintenance

Problem presented by NEOTURF at the
86^{th} European Study Group with Industry
7^{th}–11^{th} May 2012
Laboratório de Engenharia Matemática
Instituto Superior de Engenharia do Porto
Portugal

March 8, 2013

Problem presented by:	Paulo Palha (Neoturf)
Study group contributors:	Paulo Bárcia, Manuel Cruz, Alzira Faria, Ana Moura, Jorge Orestes Cerdeira, Paulo Vasconcelos, Ana Viana
Report prepared by:	Manuel Cruz (e-mail: mbc@isep.ipp.pt), Ana Moura (e-mail: aim@isep.ipp.pt), Jorge Orestes Cerdeira (e-mail: orestes@isa.utl.pt)

Abstract

Neoturf is a Portuguese company working in the area of project, building and garden's maintenance. Neoturf would like to have a procedure for scheduling and routing efficiently the clients from garden maintenance services. The company has two teams available during the whole year and an additional team during summer to handle all the maintenance jobs. Each team consists of two or three employees with a vehicle fully equipped with the tools that allow to carry out every kind of maintenance service. In the beginning of each year, the number and frequency of maintenance interventions to conduct during the year, on each client, are accorded. Each client is assigned to the same team and, usually, time windows are established so that visits to the client should occur only within these periods. As the Neoturf costumers' are geographically spread over a wide region, the total distance on visiting clients is a factor that has a heavy weight on the costs of the company. Neoturf is concerned with reducing these costs, while satisfying the agreements with the clients.

1.1 Introduction

Neoturf is a Portuguese company working in the area of project, building and garden's maintenance. One of their services is the maintenance of private gardens of residential customers (about 60), whose demands are mainly periodic short time interventions (usually 1 to 3 hours). In the beginning of each year, the number and frequency of maintenance interventions to conduct during the year are accorded with each client. Thus, a minimum and maximum periods of time separating two consecutive interventions on the same client are settled.

The amount of work highly depends on seasonality. The company allocates to this service, in almost full-time, two teams (each consists of two or three employees) in the winter's period and three teams during summer. Each team has a van fully equipped with the tools needed to perform the maintenance jobs. Each client is assigned to the same team and, usually, time windows were established so that visits to the client should occur only within these periods. The clients are geographically spread along an area around Oporto of approximately 10000 km^2. In 2011, these teams traveled more than 60000 km, with a significant impact on the costs.

Neoturf aims at finding a procedure to scheduling and routing clients efficiently so to reduce costs, while satisfying the agreements with the clients. The scheduling of clients for each day should be planed on a basis of short periods of time (say ten consecutive working days), since unforeseeable events (e.g., rainy days, client not available at the time previously arranged) may force to postpone planned interventions and to re-settle the designed scheduling.

1.2 Formulation

We propose to address the problem partitioning the year into consecutive short periods of time (say 10 consecutive working days) and identifying, for each period, (i) which customers should be served and (ii) how these costumers should be routed.

We formulate (i) as a 0-1 knapsack problem and (ii) as a multiple time windows travelling salesman problem.

1.2.1 Identifying clients to visit in each period

The identification of which clients should be served on a given period P, of m consecutive days, takes into account the interventions carried out on previous periods. Clients are classified as

- mandatory, those for which an intervention has to take place during period P, i.e., the number of days since the last visit till the end of period P exceeds the maximum number of consecutive days which can elapse without any intervention taking place, according to what has been agreed with the client;

- discarded, those for which no intervention is expected to take place during period P, i.e., the number of days since the last visit till the end of period P is lower than the number of consecutive days that were agreed to elapse before a new intervention takes place;
- admissible, those for which an intervention may or may not take place during period P.

In order to proceed with the selection of the clients to be visit, besides the expected time t_c for the intervention on every client c, for which reliable estimates exist, there is need of an estimate e_c of the traveling time to arrive at client c. There are several possibilities for settle e_c from the estimates ρ_{uv} of the time to travel directly from location u to v, where u and v represent either two clients, or one client and the depot (the location where vehicles are parked, and from where the vehicles depart to daily maintenance services and return when services are accomplished). A possibility, which may be reasonable if P is the initial period, or there is lack of information from the previous periods, is just to define e_c as the mean or some quartile of ρ_{uc}, for appropriate u. If several visits to client c have already took place, then e_c could be defined as the mean of the times to arrive at client c from the clients visited immediately before, on the previous periods. In addition, let T be the estimate working time for the whole period P, not including the time of the arrivals at depot, which can be estimated as m times the estimate of each arrival calculated as for clients.

We first have to check whether T is enough to serve all mandatory clients, i.e., if the following inequality holds

$$\sum_{c \in M} t_c + e_c \leq T, \tag{1.1}$$

where M denotes the set of mandatory clients for period P.

In case the inequality is not verified, the current version of our implementation produces a warn on this fact, and the decision maker should decide whether to extend period P, or to postpone the interventions on certain costumers.

If inequality (1.1) is satisfied, admissible clients are selected to fill as much as possible the time that remains from serving the mandatory clients. This can be formulated as follows.

$$\text{Max} \quad \sum_{c \in A} p_c x_c, \tag{1.2}$$

subject to

$$\sum_{c \in A} p_c x_c \leq T', \tag{1.3}$$

where A denotes the set of admissible clients, $p_c = (t_c + e_c)$ and $T' = T - \sum_{c \in M}(t_c + e_c)$ is an estimate of the time that remains from serving the mandatory clients.

This problem is known in the literature as the 0-1 *Knapsack Problem* (see, e.g., [7, 9]). Although the 0-1 knapsack is NP-hard, it is readily computable for the sizes of sets A that arises on Neoturf' instances.

In the computational application that we developed the above problem was coded using Octave and solved to optimality.

1.2.2 Routing clients assigned to the same period

The *Travelling Salesman Problem* (TSP) is described as the problem of find a minimum-cost route that visits all the cities of a set exactly once. It was first publicised by Flood [6], but its origins are unclear. Due to its several applications in people's daily lives, the study of this problem has been growing steadily. See Lawler et al [8] for a survey on this matter.

Usually, the spatial routing problem is mixed with the temporal aspect of scheduling, where the visits to the cities must respect certain time-windows constraints. This extension of the initial problem is called the *Travelling Salesman Problem with Time Windows* (TSPTW), and it is in this variant that our problem is framed.

As the TSP is NP-hard, so is the TSPTW. Savelsbergh [10] showed that even the problem of finding a feasible solution for the TSPTW is NP-complete. However, because of the small sizes of the sets of clients in Neoturf's problem, the TSPTW can be implemented. See, e.g., [2, 5] for some models of programming.

We based our formulation in the first one given by [4].

Suppose C is the set of clients (assigned to some team) that are to be routed in a given period of m days. We construct a directed weighted graph $G = (V, A, \rho)$ as follows. The set of vertices V is equal to $C \cup B$, where vertices of C represent clients and each vertex b_i of B, $i = 0, \cdots, m$, is the i "day (fictitious) copy" of the depot. There is an arc (u, v) linking client u to client v if there is any possibility (w.r.t. travelling time between u and v and time windows) to serve v immediately after visiting u. Arcs with both directions link each vertex b_i, $i = 1, \cdots, m-1$, with every client. Vertex b_0 is also incident to every vertex in C, but there will be no arcs entering b_0. Every vertex in C is incident to b_m , but no arcs exist leaving b_m. The other arcs in set A are $(b_0, b_1), (b_1, b_2), \cdots, (b_{m-1}, b_m)$, and no more arcs exist linking pairs of vertices in B. For $v \in V$, we use V_v^+ and V_v^- to denote the set of vertices leaving and entering vertex v, respectively, i.e., $V_v^+ = \{u \in V : (v, u) \in A\}$ and $V_v^- = \{u \in V : (u, v) \in A\}$.

A scheduling of clients will read on graph G as a directed path Q from b_0 to b_m, including every other vertex of V exactly once. The clients that are to be visited on day i are the vertices of C on the subpath of Q linking b_{i-1} to b_i. The order of vertices on that path specifies the order by which the corresponding clients should be visited. If arc (b_{i-1}, b_i) is included in path Q it means that no interventions on clients of set C will occur on day i.

We now define the weights on the arcs of A. The weight ρ_{uv} is the time to travel on arc $(u, v) \in A$, if u and v are both in C, and $\rho_{uv} = 0$, for arcs $(u, v) \in A$, with $u, v \in B$. If one vertex of arc $(u, v) \in A$ is in C and the other in B, the weight ρ_{uv} is the time of traveling along arc (u, v) plus some large positive constant so that ρ_{uv} exceeds the weight of every arc with both vertices in C.

For each vertex $v \in C$, let $T_v^i = [e_v^i, l_v^i]$ be the i-th time-window of client v, $i = 1, \cdots, |T_v|$, with $e_v^i < l_v^i < e_v^{(i+1)}$, where e_v^i and l_v^i are the release time and the deadline time of the i-th time-window of client v, respectively. For vertices of B, define $T_{b_0}^1 = [ST, ST]$ and $T_{b_i}^1 = [EN+24(i-1), EN+24(i-1)]$, for $i = 1, \cdots, m$, where ST and EN are, respectively, the daily service start hour and the daily service end hour.

Recall that, for $v \in C$, t_v is the processing time on client v, and set $t_b = 0$, for every $b \in B$.

To formulate the problem we use the following variables.

- $x_{uv} = \begin{cases} 1, & \text{if arc } (u,v) \in A \text{ is selected,} \\ 0, & \text{otherwise;} \end{cases}$
- $y_v^i = \begin{cases} 1, & \text{if client } v \in V \text{ is served in the time-window } T_v^i, \text{ with } i \leq |T_v|, \\ 0, & \text{otherwise;} \end{cases}$
- $s_v \geq 0$ is the start time, i.e., time-instant in which the service starts at client $v \in V$;
- $w_v \geq 0$ is the waiting-time to start the service at client $v \in C$, if the vehicle arrives early than the release time.

We deem minimise the sum of travel-time and waiting-time on clients. We thus have the following objective function.

$$\text{Min} \sum_{(u,v)\in A} \rho_{uv} x_{uv} + \sum_{v \in C} w_v \tag{1.4}$$

The following equations

$$\sum_{u \in V} x_{vu} = 1, \quad \forall v \in V \backslash \{b_m\}, \tag{1.5}$$

$$\sum_{u \in V} x_{uv} = 1, \quad \forall v \in V \backslash \{b_0\}, \tag{1.6}$$

ensure there will be exactly one arc leaving every vertex $v \neq b_m$, and exactly one arc entering every vertex $v \neq b_0$.

To force that each client is visited exactly in one of its time-windows, we add equations

$$\sum_{i \leq |T_v|} y_v^i = 1, \quad \forall v \in V. \tag{1.7}$$

To guarantee that the start time occurs within the selected time-window and that vehicle has enough time to travel from u to v, we use the following constraints

$$\sum_{i \leq |T_v|} e_v^i y_v^i \leq s_v \leq \sum_{i \leq |T_v|} l_v^i y_v^i, \quad \forall v \in V, \tag{1.8}$$

$$s_u + t_u + \rho_{uv} - (1 - x_{uv})M \leq s_v, \quad \forall (u, v) \in A, \tag{1.9}$$

where $M > 0$ is large enough to guarantee that the left hand side is non positive whenever $x_{uv} = 0$, and thus making constraint (1.9) not active when $x_{uv} = 0$.

Note that constraints (1.5),(1.6) together with (1.9), ensure that the set of selected arcs defines a directed path linking b_0 to b_m, which passes through every vertex of V exactly once.

The following inequalities define upper bounds on the waiting times on clients.

$$w_v \geq \sum_{i \leq |T_v|} e_v^i y_v^i - (s_u + t_u + \rho_{uv}) - (1 - x_{uv})M', \quad \forall (u, v) \in A, v \in C, \tag{1.10}$$

where $M' > 0$ is large enough to guarantee that the right hand side is non positive whenever $x_{uv} = 0$, thus turning the constraint (1.10) redundant when $x_{uv} = 0$.

The range of the variables is established as follows.

$$x_{uv} \in \{0, 1\}, \quad \forall (u, v) \in A \tag{1.11}$$

$$y_v^i \in \{0, 1\}, \quad \forall v \in V, \text{ and } i \leq |T_v| \tag{1.12}$$

$$s_v \geq 0, \quad \forall v \in V \tag{1.13}$$

$$w_v \geq 0, \quad \forall v \in C \tag{1.14}$$

The above model (1.4)-(1.14) gives a mixed integer linear programming formulation for the problem of routing clients assigned to the same team on a given period of m days. Note that weights we assigned to the arcs entering in the fictitious depots $b_1, \cdots, b_m$, where some large constant has been added to the time of traveling along each of these arcs, make these arcs unattractive for solutions with low values of the objective function. As a result, good solutions tend to include, as much as possible, arcs (b_{i-1}, b_i), which means not assigning services on day i. Thus, optimal solutions assign services to clients in the minimum number of days for which feasible schedulings exist. Clearly, the days for which no interventions were planned will be considered for the planning of the subsequent period.

1.3 Implementation details and results

A tester version of the proposed was made during the ESGI'86, in order to test it with real data kindly provided by Neoturf. On Figure (1.3) a simplified workflow of the implemented version is presented.

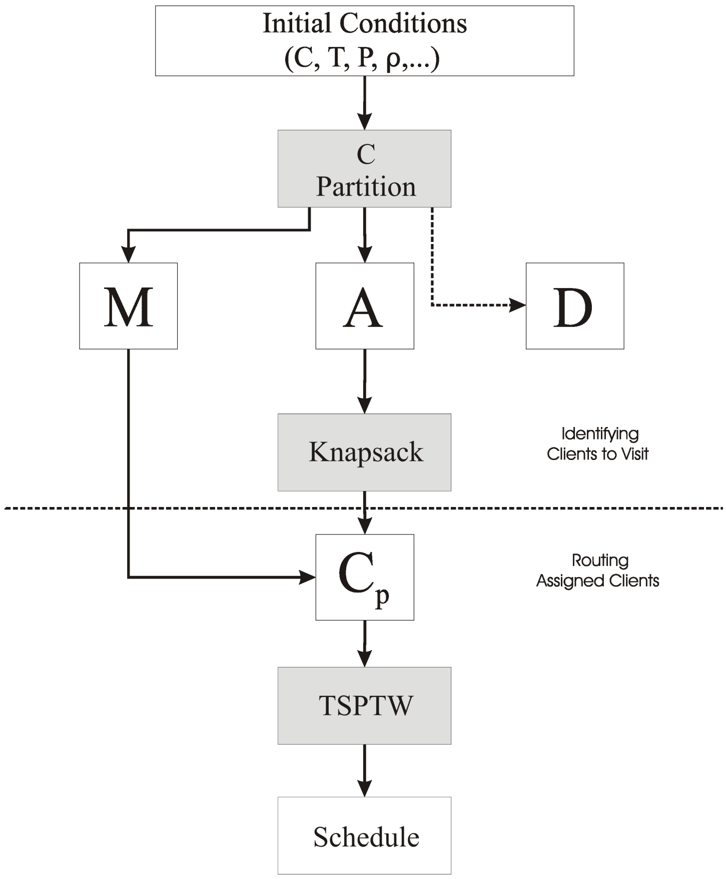

Figure 1.1: Neoturf algorithm workflow.

In a 5-day ESGI context the following assumptions were made:

- The definition of ρ in 1.4 in terms of time instead of distance unit is justified by the fact that, accordingly with Neoturf, the time unit cost is higher than the kilometric cost, in this context. As so, although ρ_{uv} is not constant, as it depends on factors like traffic or weather among others, a rough measure of the time was considered as the preferential measure.
- The lunch time is not considered explicitly. This is because Neoturf's employees have some flexibility regarding that subject. Thus, this

version was implemented by closing the time-window of the depot one hour earlier, that is, instead of a 8-hour working day we used a 7-hour working day.

- All the clients are previously assigned to one of the teams (actually this really happens for all Neoturf clients except two of them), so routine runs the algorithm in parallel.

We made a simplified implementation of the workflow presented in Figure (1.3) using Octave. For the problem adressed in Subsection 1.2.1 we used a classical Knapsack implementation. In order to have an initial solution for the routing problem our implementation used the Clark-Wright savings algorithm [3]. The computation time for both problems was in the order of seconds, for a sample with 64 clients, two teams and periods of few days with different initial conditions. This was one of the reasons for the Neoturf satisfaction, because it usually takes one or two days for their logistic department to get the monthly plan ready.

1.4 Conclusions and recommendations

The Neoturf challenge was successfully completed, although with some simplifications. It is strongly recommended a further development post-ESGI as future achievements may be completed with a small cost/benefit ratio. Nevertheless, Neoturf may use this first version to test and greatly simplify their work in planning their team's schedule. One first change recommendation is related with an Neoturf internal procedure: At this moment Neoturf does a primary scheduling at the beginning of every month that is afterwards communicated to their clients. Our previous calculations and modelling show us that, from the optimisation point of view, the scheduling should be done in shorter periods of time as the frequent imponderables make necessary a most adaptive procedure. As so, and based on the sample data provided by Neoturf, a week prior to intervention seems a good compromise between the client comfort and a Neoturf manageable and optimised scheduling procedure.

Regarding future development on this work, the following points should be addressed:

- As stated in (1.1), when T is not enough for all the Mandatory-services the user receives a warning and should decide whether to extend period P, or to postpone the interventions at certain costumers. An interactive procedure with multi-objective (age of the last visit and/or distance), adaptative or recurrence criteria may be study and implemented.

- As a consequence of our modelling using the Knapsack, in some days there is some available time in the end of the day. Neoturf representative says that it is positive, as the teams must do some maintenance on the equipment almost every day. Nevertheless, some post-optimisation may be made. For example, an iterative process to try to include more clients in the schedule may improve the Neoturf productivity.
- In the case that bigger simulation periods are demanded by Neoturf, a process that include more than one visit to each client should be study.
- The company is aware that the solution provided in the computational implementation is, most certainely, not optimal. However, as the problem is formulated, an efficient implementation of TSPTW may be made with a relative small effort.

The authors would like to thank to all the ESGI86 participants, as their suggestions and remarks throughout those five days were of major importance in order to have a first sketch of this solution.

Bibliography

[1] E. Alba, C. Blum, P. Asasi, C. Leon and J.A. Gomez, Optimization Techniques for Solving Complex Problems, Wiley Publishing (2009).

[2] E. K. Baker, An exact algorithm for the time-constrained travelling salesman problem, Operations Research 31 (5) (1983) 938-945.

[3] G Clarke and J.W. Wright, Scheduling of Vehicles from a Central Depot to a Number of Delivery Points, Operations Research, 12 (4) (1964) 568–581.

[4] S. Dash, O. Gnlk, A. Lodi and A. Tramontani, A Time Bucket Formulation for the Traveling Salesman Problem with Time Windows, INFORMS Journal on Computing 24 (1) (2011) 132-47.

[5] Y. Dumas, J. Desrosiers, E. Gelinas, M.M. Solomon, An optimal algorithm for the travelling salesman problem with time windows, Operations Research 43 (1995) 367-371.

[6] M. M. Flood, The Traveling-Salesman Problem, J. Opns. Res. Soc. Amt. 4 (1955) 61-75.

[7] H. Kellerer, U. Pferschy and D. Pisinger, Knapsack Problems, Springer, Berlin (2004).

[8] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan and D. B. Shmoys, The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization, John Wiley & Sons, 1985.

[9] S. Martello and P. Toth, Knapsack Problems: Algorithms and Computer Implementations, Wiley, New York (1990).

[10] M. W. P. Savelsbergh. Local search in routing problems with time windows. Annals of Operations Research 4 (1) (1985) 285-305.