# MOTOR2020 competition

## Field management engineer scheduling problem

Igor Vasilyev[1,3], Dong Zhang[2], Jie Ren[3], and Sergey Kudrya[3]

[1]*Matrosov Institute for System Dynamics and Control Theory of Siberian Branch of Russian Academy of Sciences, Irkutsk, Russia*
[2]*Algorithm & Technology Development Dept., Global Technical Service Dept., Huawei Technologies, Co., Ltd., Dongguan, China*
[3]*Moscow Advanced Software Technology Lab, Huawei Russian Research Institute, Moscow, Russia*

March 5, 2020

## 1 Problem statement

Huawei Global Technical Service (GTS) department provides the technical support for a large number on communication networks worldwide. For example, more than 120 mobile operators in more than 75 countries are clients of GTS department and there are more than 28 thousands service employees involved to deliver agile and high-quality services on a global scale, tailored to local needs..

One of the problem, which arise in the maintenance and service of networks, is the effective use of field management engineers (FME) for maintenance of the equipment located in different, spaced apart sites. There is a regional maintenance office, which is responsible for the operation of equipment in various sites of some area. We need to schedule service tasks for FMEs, while minimizing service cost and maximizing the quality of service. More formally, the problem is stated as follows. We are given with:

- $J$ — a set of tasks.
- $c_j$ — a processing time of task $j \in J$.
- $[a_j, b_j]$ — a time window in which task $j \in J$ has to be processed, the interruptions of task processing are not allowed.
- $V$ — a set of sites, where tasks are located and an office is also included, which contains no tasks.
- $J_u \subset J$ — a subset of task located in site $u \in V$.
- $d_{uv}$ — duration (travel time, distance) between site $u \in V$ and $v \in V$.
- $F$ — staff, i.e. a set of persons (FMEs) which can complete tasks.

- $J_f \subset J$ — subset of tasks, which can be completed by engineer $f \in F$ according to its skill.
- $S$ — a set of all shifts.
- $[\alpha_s, \beta_s]$ — a time window of engineer $s \in S$.
- $S_f \subset S$ — a subset of shifts of engineer $f \in F$.

The problem consists of making a schedule for each shift. Let $t_j$ is the beginning time of task $j \in J$. We have to satisfying the following conditions:

1) Engineer $f$ starts his shift $s \in S_f$ from the office at $\alpha_s$.
2) A shift schedule consists of an ordered subsets of tasks to be completed and a route between sites, where corresponding tasks are located.
3) Engineer $f$ can process only tasks $j \in J_f$.
4) Engineer $f$ finishes his shift at the site of the last scheduled task. The shift duration consists of travel time between corresponding sites, waiting time and processing time of corresponding tasks.
5) Waiting time is the time in a shift which is not taken neither processing of tasks nor traveling between sites. Waiting time is also the time between the end of processing the last task in the shift and the end of the shift.
6) Engineer $f$ should finish his shift within the corresponding time window $[\alpha_s, \beta_s]$. If he finishes after $\beta_s$, the overtime is penalized, but the overtime cannot in one shift cannot be more than $L^{over}$ time units.
7) For task $j$, beginning of task processing time cannot be earlier $a_j$, i.e $t_j \geq a_j$. The end of processing time should be not later than $b_j$, i.e. $t_j + c_j \leq b_j$. The tardiness is penalized.
8) Each task $j \in J$ should be completed. Uncompleted tasks are penalized.
9) Task $j \in J_u$ in the same site $u \in V$ can be completed in different shifts but by the same engineer.

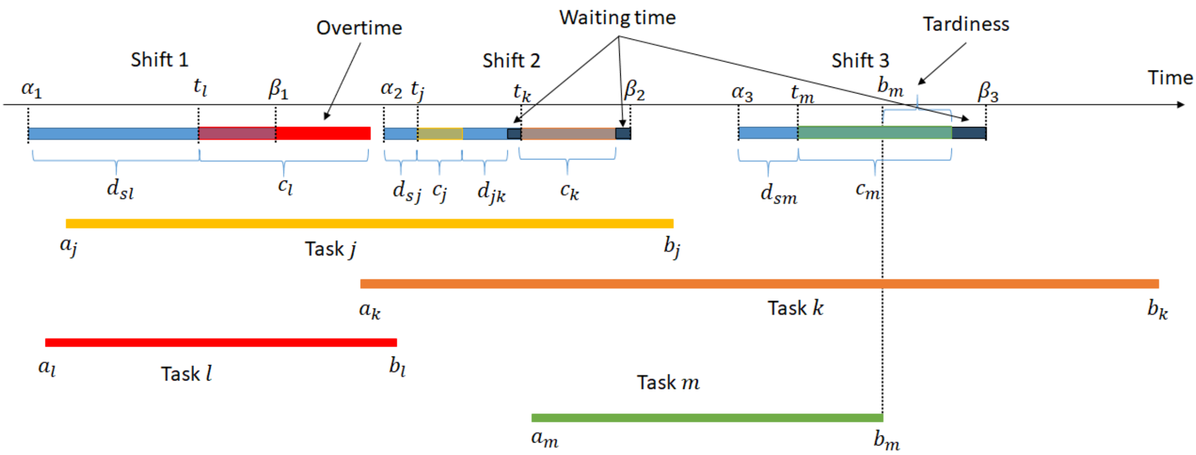


Figure 1: Small example of problem statement

An example of a part of scheduling is illustrated in Figure 1. The schedule has to be made reaching the following objectives:

(i) Maximize the number of task to be processed. $C_j^{task}$ is the cost if task $j \in J$ is not completed. If $C_j^{task} = \infty$ then task $j$ must be completed.

(ii) Minimize the number of staff to be involved in the schedule. If engineer $f$ is involved, cost (penalty) $C_f^{staff}$ is paid.

(iii) Minimize the total travel time, $C^{travel}$ is the cost of travel time per time unit.

(iv) Minimize the total waiting time, $C^{wait}$ is the cost of waiting time per time unit.

(v) Minimize the total overtime, $C^{over}$ is the overtime cost per time unit.

(vi) Minimize the total tardiness, $C_j^{tard}$ is the tardiness cost of task $j \in J$. If $C_j^{tard} = \infty$ then task $j$ must be completed within its time window.

(vii) Maximize the staff preferences. $P_{fv}^{pref}$ is the cost of processing tasks of site $v$ by engineer $f$, the less its value the tasks of this site is more preferable for the corresponding engineer.

(viii) Numbers of tasks and sites for each engineer have to be balanced. $C_{task}^{balance}$ and $C_{site}^{balance}$ are the corresponding costs if the numbers are different from average per unit.

# 2 Test instances and solution evaluation

The data input for each instance consists of 6 files: **staff.csv**, **costs.csv**, **shifts.csv**, **tasks.csv**, **traveltime.csv** and **preferences.csv**. Time unit is one minute. The file formats are Comma-Separated Values (csv) and described below.

- **costs.csv**
  - *First line:* Cost of travel time, i.e. $C^{travel}$.
  - *Second line:* Cost of waiting time, i.e. $C^{wait}$.
  - *Third line:* Cost of overtime, i.e. $C^{over}$.
  - *Fourth line:* Cost of task balance, i.e. $C_{task}^{balance}$.
  - *Fifth line:* Cost of site balance, i.e. $C_{site}^{balance}$.
  - *Sixth line:* Overtime limit, i.e. $L^{over}$.
- **staff.csv**
  - *First line:* Number of engineers.
  - *Second line:* Column titles, i.e. Engineer's ID, Skills, Cˆstaff ($C_f^{staff}$).
  - *Other lines:* For each engineer in each line there are corresponding Engineer's ID, its skills. There are three skills, which are denoted by A, B and C. Engineer's skills is a string, which consists of these three letters.
- **shifts.csv**
  - *First line:* Number of shifts.

- *Second line:* Column titles, i.e. Shift ID ($s$), Engineer's ID, alpha ($\alpha_s$), beta ($\beta_s$).
- *Other lines:* For each shift in each line there are data which corresponds to column titles.

- **tasks.csv**
  - *First line:* Number of tasks.
  - *Second line:* Number of sites. It includes the office, which ID equals 0 and it does not contain any tasks.
  - *Third line:* Column titles, i.e. Task ID ($j$), Required Skill, Site ID, c_j ($c_j$), a_j ($a_j$), b_j ($b_j$), C^tard ($C_j^{tard}$), C^task ($C_j^{task}$)
  - *Other lines:* For each task in each line there are data which corresponds to column titles.
- **traveltime.csv** contains a matrix of travel time between sites, i.e. $d_{uv}$ $\forall u \in V$ and $\forall v \in V$. each matrix row is in one line. Remember that it contains the office, i.e. starts from site ID 0.
- **preferences.csv** contains a matrix of preferences, i.e. $C_{fv}^{pref}$ $\forall f \in F$ and $\forall v \in V$.

A solution have to be submitted in a file in the following format:

- **solution.csv**
  - *First line:* Number of completed tasks. Tasks, which are not listed in this file, are supposed to be not completed.
  - *Second line:* Column titles, i.e. Task ID ($j$), Shift ID ($s$), Eingineer's ID ($f$), order (the order in which the task is completed, the first is 1, second – 2, etc.), begin time ($t_j$), end time ($t_j + c_j$).
  - *Other lines:* For each task in each line there are data which corresponds to column titles.

A solution file is checked with respect to the conditions 1)-9) and evaluated by the sum of objectives (i)-(viii).

We provide competitors with C++ code and Python script, which read the instance data and solution and then check and evaluate the solution. The both of them are quite easy and self-explanatory.