Shift design in a Client Contact Center

Cyril Canon[1,2], Nicolas Monmarché[2], Jean-Charles Billaut[2], Jean-Louis Bouquard[2], and Mathieu Salvant[2]

[1] VITALICOM, 643 avenue du grain d'or,
41350 Vineuil, France
ccanon@fr.snt.com
[2] Université de Tours, Laboratoire d'Informatique, 64 avenue Jean Portalis,
37200 Tours, France
{billaut,monmarche,bouquard}@univ-tours.fr

We deal in this paper with the Shift Design Problem: the problem is to create shifts and assign them to employees over the week, according to the labour code and respecting a given load curve. The constraints are of two types:

- Constraints per agent: the sum of duration of work over the week; each agent has exactly two days-off per week; each agent has at least 11 hours of rest between two consecutive days of work.
- Constraints per period: the load curve has to be respected.

The objective is to smooth the amount of workload each period. This problem can be encountered in any type of industry, for which the determination of the employees shifts constitutes a critical problem.

In order to solve this problem, we propose and compare three heuristic algorithms. The first one is a truncated Branch-and-Bound procedure based on Constraint Programming. The second one is a Tabu Search algorithm and the last one is an Ant Colony Search method. We present the detailed algorithm for each heuristic and the main results.

The first heuristic is a truncated branch-and-bound procedure, based on Constraint Programming (CP). For each day of the week and for each agent, the decisional variables are the starting times of work and of break and the duration of work and of break. It is easy to set the constraint that some operations have a cummulated activity less or equal to a given limit. In our problem the total activity has to be greater or equal to the load curve. Thus our reasoning is reversed: the inactivity periods are enforced to be less or equal to a given curve: the slight transformation - described below - is used.

We consider a cumulative resource. We associate to this resource a profile equal at each period to the total number of agents minus the number of agents required this period. Each agent is considered as a job to schedule composed by three tasks: the first task starts at the beginning of one day and its duration has to be determined, for the second task the starting time and the duration have to be determined and the third task completes at the end of the day and its starting time has to be determined. In fact, the tasks correspond to the non activity of work of each agent. Solving this problem can be done using the classical cumulative constraint in CP.

The second heuristic is a Tabu Search algorithm, inspired from the method proposed by [Musliu et al.]. We call *legal solution* a solution that respect constraints per agent, but not necessary constraints per period. At the beginning, an initial legal solution is produced by a simple algorithm, for which the main idea is to start a shift each time the workload increases. The length of the shift, length of the break and duration between start of the shift and start of the break are generated randomly in their definition domain. Then, we explore the neighbourhood by using a Tabu Search algorithm. The authorised moves are:

1. the swap between two equivalent days (equal amount of work) for two different agents,
2. the swap of the shifts between a day-off and a day of work for an agent,
3. start a shift earlier or later,
4. increase or decrease the length of a shift,
5. start a break earlier or later,
6. increase or decrease the length of a break.

In order to restrict the neighbourhood, a move is not authorised if it leads to an illegal solution. Each solution can be reduced into a *reduced solution*, which is a matrix with one row for each type of agent (same number of work hours in the week and same min-max number of work hours each day) and one column for each period, representing how many agents of each type are assigned to each period. Two solutions are equivalent if they have the same reduced solution. The Tabu list is composed by the reduced solutions of the n last visited solutions. Two different algorithms based on this method are tested:

– the first one only uses swap moves (1 and 2) for the neighbourhood definition and at each step, a steepest descent based on the other moves (3 to 6) is called to locally improve the best neighbour.
– the second one uses all the moves for the neighbourhood definition.

These methods are also compared to a simple steepest descent algorithm.

The third heuristic is an Ant Colony Search algorithm, inspired from the method proposed by [Solnon, 2002]. The principle of this search is to assign a value to each variable (beginning and end of each shift, beginning and duration of each break and days-off) according to pheromone values stored in a graph. The objective is to minimize the number of weighted violated constraints.

At each step, each ant constructs iteratively a solution and then improves it. We compute for each ant the number of violated constraints of its solution and pheromones are updated according to the best solution found. We repeat these two phases (solutions building and pheromones updating) until we find a solution that does not violate any constraint or if a given number of iterations is reached.

To construct a solution, the ant chooses a variable not yet assigned to a value. Then, for this variable, a value is selected according to the pheromone intensities and a random parameter is used to perform either exploration or exploitation. The pheromone trails are initialized according to the value found by the same algorithm used to determine the initial solution in the second heuristic.

We present in details the three heuristic algorithms and their computational results, both in terms of quality and computation time. For the definition of benchmark instances we use a realistic data generator and some industrial instances.

References

[Azmat and Widmer, 2004] C.S. Azmat and M. Widmer: A case study of single shift planning and scheduling under annualized hours: A simple three-step approach. European Journal of Operational Research. **153**:1 (2004) 148-175

[Fukunaga et al., 2002] A. Fukunaga and E. Hamilton and J. Fama and D. Andre and O. Matan and I. Nourbakhsh: Staff scheduling for inbound call centers and customer contact centers. AI Magazine. **23**:4 (2002) 30-40

[Musliu et al.] Musliu N., Schaerf A. and Slany W.: Local search for shift design. European Journal of Operational Research. **153**:1 (2004) 51-64

[Partouche, 1998] Partouche A.: Planification d'horaires de travail : Méthodologie, modélisation et résolution à l'aide de la Programmation Linéaire en nombres entiers et de la Programmation par Contraintes. PhD thesis (1998) Université Paris-Dauphine, Paris.

[Solnon, 2002] Solnon, C.: Ants can solve constraint satisfaction problems. EEE Transactions on Evolutionary Computation. **6**:4 (2002) 347-357