SCHEDULING EMPLOYEES IN QUEBEC’S LIQUOR STORES WITH INTEGER PROGRAMMING

Bernard Gendron
Département d’informatique et de recherche opérationnelle
Université de Montréal

February 2004
• SAQ (Société des alcools du Québec):
  – public corporation of the Province of Quebec
  – distributes and sells alcohol-based products
  – more than 400 stores + warehouses (divided into geographical areas)
• Problem: generate the weekly schedules of more than 3000 employees
• Process handled manually until 2002
• Very costly: salaries + complaints = 1,300,000$/year
• Objective: develop a Web-based database system to automate this process
CONTRIBUTION OF OR

• Problem: generating optimal schedules cannot be achieved with simple heuristics!
• Optimal schedule: strictly adhere to all union agreement rules
• Methodology of choice: Integer Programming (IP)
DECOMPOSITION BY EMPLOYEE

- The union agreement imposes a sequential assignment!
- The most senior employee gets the best schedule
- The best remaining schedule is assigned to the next most senior employee
- And so on, until all shifts are assigned
- Guaranteed to produce a feasible schedule: enough employees to fill the requirements of all stores
DECOMPOSITION BY DAY

• Schedules are planned every week
• Any employee cannot work more than 10 hours daily and 38 hours over the whole week
• **Backward Assignment Rule:** the schedule must be generated on a daily basis, starting from the end of the week (Saturday) and going backward until the beginning (Sunday)
• Rationale: drive the days off towards the beginning of the week
• Of course, we could do better (assign more work hours) by planning over the whole week, but we are not allowed to do so: we must respect all union agreement rules!
SOME BASIC DEFINITIONS

• Each day is divided into 15-minute periods

• Interval: set of consecutive time periods

• Work interval: interval entirely worked by an employee

• Categories of work intervals:
  – Daily: 8:00am-11:00pm
  – Overnight: 9:00pm-6:00am
  – Mixed

• When planning a schedule for a given day, we must also consider the last three hours of the day before!

• Discontinuity: two disjoint work intervals on the same day
SOME RULES

• *Rest Rule:* at least 8 hours of rest, after and before an overnight work interval

• Easy to enforce using the Backward Assignment Rule, but we must know when the employee stopped working on the Saturday of the week before!

• *Substitution Rule:* a *guaranteed shift* is assigned to the employee, unless there is another schedule that produces more work hours of the same work interval category and in the same store

• *Travel Time Rule:* when assigned to two different stores, the employee must have enough time to travel between them

• No more than two discontinuities allowed

• Favor schedules with no discontinuity: to be chosen, a schedule with at least one discontinuity must have more than one work hour more than any other schedule with no discontinuity
SOME DIFFICULT RULES

- **Lunch Break Rule:** the employee has a one-hour *unpaid* lunch break when the work interval is entirely contained in the interval 10:30am-3:30pm.

- Similar rules for dinner (3:30pm-8:30pm) and overnight breaks.

- **Split Shift Rule:** a *splittable shift* can be split into two parts, the *piece*, assigned to the employee, and the *residual*, subsequently assigned to another employee.

- Each work interval must contain at least three hours: a splittable shift has at least 6 work hours, and both the piece and the residual have at least three work hours.

- Each rule would be easy to manage (and model) independently, but their interaction creates problematic situations!
SOME PROBLEMATIC SITUATIONS

• The employer must not pay more work hours than required: if the employee is assigned a work interval of \( p \) work hours, obtained by splitting some shifts whose total number of work hours is \( n \), the residuals should not contain more than \( n - p \) work hours.

• Adjust the residual:
  – Remove one hour
  – Remove two hours
  – Add one hour

• Forbid *opportunistic split*, a split that allows to avoid the lunch break.
OBJECTIVE FUNCTION

- Maximize number of work hours
- Task preference
- Minimize discontinuities
- Minimize number of stores
- Minimize number of split shifts
- Store preference
- Favor earliest periods (starting at 8:00am)
- Maximize number of shifts
SOME CONSTRAINTS

• Assignment Constraints: \( y_i = 1 \): if period \( i \) is assigned
  \( z_j = 1 \): if shift \( j \) is assigned
  \( x_{ij} = 1 \): if splittable shift \( j \) is assigned at period \( i \)

  \[ \sum_{j \in J_i^U} z_j + \sum_{j \in J_i^D} x_{ij} = y_i, \quad i \in I \]

• Break Constraints: \( r_k = 1 \): if the employee takes break \( k \)

  \[ r_k \geq \sum_{i \in I_k} y_i - |I_k| + 1, \quad k \in K \]

• Work Hour Constraints: \( y_i^W = 1 \): if period \( i \) is worked

  \[ \sum_{i \in I_k} y_i^W = \sum_{i \in I_k} y_i - \rho r_k, \quad k \in K \]

  \[ y_i^W = y_i, \quad k \in K, i \notin I_k \]
DISCONTINUITY CONSTRAINTS

• $u_i = 1$: if the employee is assigned to period $i$, but not period $i - 1$

\[ u_i \geq y_i - y_{i-1}, \quad i \in I, i \neq f(I) \]

\[ u_i \leq y_i, \quad i \in I \]

\[ u_i \leq 1 - y_{i-1}, \quad i \in I, i \neq f(I) \]

• Each work interval must contain at least three hours:

\[ u_i \leq y_{i'}, \quad i \in I, i < i' \leq \min(i + \tau - 1, l(I)) \]

• $u_i^D = 1$, if there is a discontinuity at period $i$

\[ u_i^D \geq u_i + u_{i'} - 1, \quad i \in I, f(I) < i' < i \]

• No more than two discontinuities:

\[ \sum_{i \in I} u_i^D \leq \phi \]

• $u_0^D = 1$, if there is at least one discontinuity

\[ u_0^D \geq u_i^D, \quad i \in I \]
**SPLIT SHIFT CONSTRAINTS**

- \( v_{ij} = 1 \): if splittable shift \( j \) is split “forward” at period \( i \), i.e., \( x_{ij} = 1 \) and \( x_{(i-1)j} = 0 \)

- \( w_{ij} = 1 \): if splittable shift \( j \) is split “backward” at period \( i-1 \), i.e., \( x_{ij} = 0 \) and \( x_{(i-1)j} = 1 \)

\[
x_{ij} - v_{ij} - x_{(i-1)j} + w_{ij} = 0, \quad i \in I, j \in J_i^D \cap J_{i-1}^D, i \neq f(I)
\]

- Each splittable shift can be split only once:

\[
\sum_{i \in I_j} (v_{ij} + w_{ij}) \leq 1, \quad j \in J^D
\]

- The residual must contain at least three hours (simplified version):

\[
\sum_{i \in I_j} x_{ij} + \tau \sum_{i \in I_j} (v_{ij} + w_{ij}) \leq |I_j|, \quad j \in J^D, |K_j| = 0
\]
OBJECTIVE

- Maximize Number of Work Hours:
  \[
  \sum_{i \in I} y^W_i
  \]

- Penalize Discontinuity:
  \[-u^D_0\]

- Task Preference: \( o_{it} = 1 \): if period \( i \) is assigned to task \( t \)
  \[
  \sum_{j \in J^I_i \cap J_t} z_j + \sum_{j \in J^D_i \cap J_t} x_{ij} = o_{it}, \quad i \in I, t \in T_i
  \]

- \( o^W_{it} = 1 \): if period \( i \) is worked and assigned to task \( t \)
  \[
  o^W_{it} \leq y^W_i, \quad i \in I, t \in T_i
  \]
  \[
  o^W_{it} \leq o_{it}, \quad i \in I, t \in T_i
  \]
  \[
  \sum_{i \in I} \sum_{t \in T_i} \theta_t o^W_{it}
  \]

- Minimize discontinuities:
  \[-\sum_{i \in I} u^D_i\]
IMPLEMENTATION

• C++ code
  – interacts with the Web-based database system developed by the SAQ to acquire the data for each employee
  – implements the IP model using ILOG Concert Technology
  – solves the IP model with ILOG CPLEX

• Fine tuning CPLEX parameters is important (example: from 20 minutes to 20 seconds!)

• Most instances are solved within seconds, except a few that can take more than one hour!

• Two CPLEX licenses + queueing system (the different geographical areas can be processed independently)
CASE HISTORY

- March 2000: beginning of the project (focus on modeling split shifts)
- May 2000: first release of the C++ code
- December 2000: version 1.0 (after 13 releases involving multiple bug fixes + definition of new constraints)
- July 2001: version 3.0 (includes several new constraints: Substitution Rule, Task Preference, overnight shifts)
- August 2002: version 5.0 (interacts with the queueing system)
- Summer 2002: implementation in all stores
- July 2004: version 5.9
IMPACT ON THE ORGANIZATION

• Savings: about 1,000,000$/year (75% salaries, 25% complaints)
• Cost of developing the system: 1,300,000$
• Simplifies the work of store managers and union representatives: eliminates paperwork, simplifies data management, reduces the time dedicated to scheduling task
• Union agreement rules now interpreted in a uniform way in all stores across the Province: eliminates many complaints
• Positive impact on the working relations all across the organization (employees and store managers, union and human resources department)