

自治体水道部門の勤務表作成支援システムの開発と実運用評価

Development and Operational Evaluation of a Staff Rostering Support System for a Municipal Waterworks Department

○小林 敬明¹
株式会社 TransRecog

松村 諭²
吹田市

森口 聡子³
東京都立大学

¹ Takaaki KOBAYASHI
TransRecog Co., Ltd.
takobaya@transrecog.com

² Satoshi MATSUMURA
Suita City

³ Satoko MORIGUCHI
Tokyo Metropolitan University
satoko5@tmu.ac.jp

Abstract

This study develops and deploys a staff rostering support system for a municipal waterworks department. The model represents daily staffing, managerial coverage, branch-office assignment, weekly days-off requirements, fixed events, consecutive-work limits, and fairness criteria using binary and integer variables, and is solved with Google OR-Tools CP-SAT. To handle infeasible instances caused by strict weekly day-off requirements, a two-stage scheme first solves the strict model and, only when infeasible, relaxes the weekly-day-off constraints with shortage variables and reports the affected employees and weeks. Using operational data for 14 staff members over an eight-month planning horizon, the strict model reached optimality in approximately 177 seconds. Roster preparation time decreased from 12–18 hours to about 2–3 hours. A full-horizon formulation also avoided excess days off caused by month-wise decomposition at boundary weeks.

Keywords: staff rostering, integer optimization, CP-SAT, municipal waterworks, decision support

1 はじめに

自治体の水道部門では、サービス継続性の要請から土日祝日を含む毎日の人員確保が求められ、勤務表（ロスター）の作成は現場運営の基盤業務である。対象とした吹田市水道部浄水室では、従来、担当者が表計算ソフト上で手作業により勤務表を作成しており、1 か月分の作成に作業時間の積算で 12～18 時間を要し、職員による確認等を含め暦日として 2～3 日にわたっていた。また、複数条件の調整手順が特定の担当者の経験に依存し、担当者交代時の引継ぎも課題であった。

勤務表作成は看護師スケジューリングを中心に長く研究されており、ハード制約とソフト制約を組み合わせた数理最適化モデル、複数目的の扱い、現場での意思決定支援などが論じられている[1–3]。一方、自治体水道部門を対象とし、月境界をまたぐ週の公休管理、複数拠点への配置、固定予定、実働人数を同時に扱い、実運用への導入効果まで報告した事例は限られている。看護師スケジューリングでは夜勤やスキルミックスが主な論点であるのに対し、本研究の対象業務では週公休管理、拠点配置、固定予定の重なりなど異なる運用上の課題を有する。

本研究では、勤務表作成を 0-1 変数と整数補助変数を用いた整数制約モデルとして定式化し、Google OR-Tools の CP-SAT[4]を用いた作成支援システムを開発した。本研究の主な貢献は、(1) 水道部門固有の実務条件を、勤務扱いと実働計上の分離を含む整数制約モデルとして体系的に表現したこと、(2) 月境界週を含む計画期間全体を一括求解する方式を採用したこと、(3) 厳密条件で解が得られない場合に週公休のみを緩和し、不足箇所を提示する二段階求解を実装したこと、

(4) 実運用データによって計算性能と業務削減効果を評価したことである。水道事業は災害対応や設備異常への即応が求められるため、休日を含めた継続的な要員確保が必要である。一方で、人員は限られており、勤務条件や公平性を考慮した勤務表作成は担当者に大きな負担となる。

2 対象業務と勤務条件

対象は職員 14 名（うち係長以上 8 名、常勤 13 名、嘱託 1 名）の勤務表作成である。勤務表では、単に出勤・公休を決定するだけでなく、本庁舎側と分庁舎側の配置、研修・出張・休暇等の予定、日別の実働人数、週単位の公休数、連続勤務、休日勤務の公平性などを同時に考慮する必要がある（表 1）。

表 1: 主な勤務条件とモデル上の扱い

区分	内容	扱い
必要人数	平日・休日別の実働人数を確保	ハード
管理職配置	係長以上を毎日配置し、本庁舎側にも 1 名以上残す。2 名体制を推奨	混合
分庁舎勤務	毎日 2 名、連続割当を禁止し、回数を平準化	混合
週公休	常勤 2 日、嘱託 3 日、非常勤 4 日	二段階
固定予定	確定済みの休暇、研修、出張、グループ会議等を反映	ハード
連続勤務	6 日間の実働を 5 日以下等	ハード
公平性	祝日休、土日公休、分庁舎勤務回数を平準化	ソフト

表中の「混合」は、必須条件をハード制約、望ましい条件をソフト制約として扱うことを示す。

週公休は週起算日に基づく週単位で評価されるため、月境界をまたぐ週では前後の月の勤務が連動する。月ごとに独立して勤務表を作成すると、同じ週が二つの部分週に分断され、公休

数の過不足や不要な公休が生じ得る（図 1）．そこで、本システムでは指定された計画期間全体を一つのモデルとして求解する．

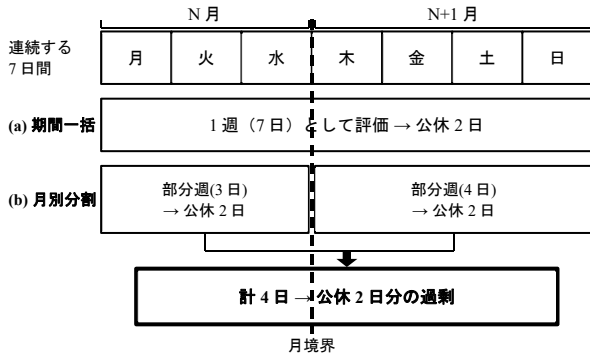
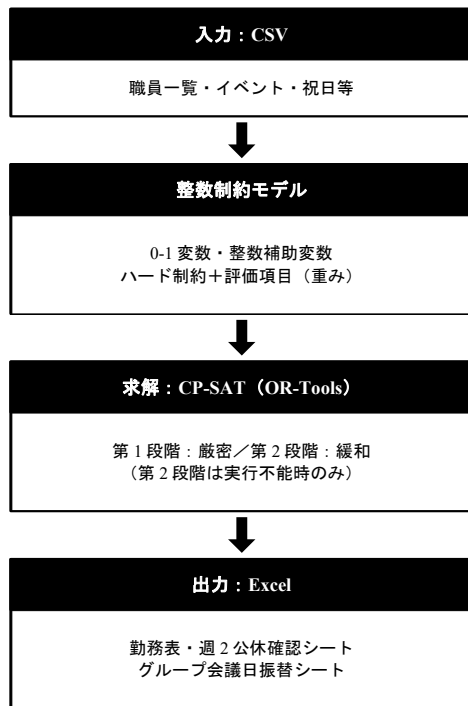


図 1: 月境界をまたぐ週の扱い

3 作成支援システム

システムは C# で実装し、入力には職員一覧、イベント一覧、祝日・閉庁日一覧等の CSV ファイルとした．求解結果は既存業務で利用している形式に近い Excel ファイルとして出力する．この構成により、新たなクラウドサービスや専用データベースを前提とせず、既存の表計算中心の運用へ後付けできる（図 2）．



C# で実装、クラウドや専用 DB を前提としない後付け構成。

図 2: システムの入出力と処理の流れ

出力は勤務表だけではない．週公休の充足状況と不足が生じた場合の内訳を示す「週 2 公休確認」シート，グループ会議日が祝日・閉庁日等

と重なった場合の振替候補を示すシートを併せて作成する．これにより、最適化結果をそのまま採用するのではなく、担当者が要確認箇所を把握して最終判断できる構成とした．

4 整数制約モデル

4.1 集合と変数

職員集合を I ，日集合を D ，係長以上の職員集合を $L \subseteq I$ ，週集合を T とする． T は、運用上定められた週起算日を基準として計画期間を 7 日単位に区切ったものである．各職員 $i \in I$ ，各日 $d \in D$ に対し、勤務扱いを表す 0-1 変数 $w_{i,d}$ ，分庁舎勤務を表す 0-1 変数 $k_{i,d}$ ，実働計上を表す 0-1 変数 $e_{i,d}$ を導入する．勤務扱いと実働を分けることで、研修・出張・休暇等の「勤務日として扱うが必要人数には算入しない予定」を表現できる．

4.2 主なハード制約

分庁舎勤務は勤務扱いかつ実働である必要があるため、変数間の関係を式(1)で表す．

$$k_{i,d} \leq e_{i,d} \leq w_{i,d} \quad (i \in I, d \in D) \quad (1)$$

日 d の必要実働人数を n_d とすると、日別の必要人数は式(2)で課す．本事例では、平日は実働人数を 6 名以上とし、土日祝日および閉庁日は別途等式制約を課して 6 名ちょうどとした．

$$\sum_{i \in I} e_{i,d} \geq n_d \quad (d \in D) \quad (2)$$

係長以上については、毎日少なくとも 1 名を実働として配置するとともに、分庁舎勤務者を除いた本庁舎側にも少なくとも 1 名を残す．

$$\begin{aligned} \sum_{i \in L} e_{i,d} &\geq 1, \\ \sum_{i \in L} (e_{i,d} - k_{i,d}) &\geq 1 \end{aligned} \quad (d \in D) \quad (3)$$

分庁舎勤務は毎日 2 名とする．

$$\sum_{i \in I} k_{i,d} = 2 \quad (d \in D) \quad (4)$$

週 t における職員 i の公休カウント対象日集合を $A_{i,t}$ ，雇用区分に応じた目標公休数を $r_{i,t}$ とする．祝日・閉庁日，グループ会議日，研修・出張等の固定予定は通常の公休候補から除外し，第 1 段階では式(5)を等式として課す．なお，式(5)は完全な 7 日からなる週にのみ課し，計画期間の両端に生じる 7 日未満の部分週は対象から除外する．

本モデルにおいて最も重要な制約は週公休日数制約である．

$$\sum_{d \in A_{i,t}} (1 - w_{i,d}) = r_{i,t} \quad (i \in I, t \in T) \quad (5)$$

このほか、連続する 6 日間の実働を 5 日以下とする制約，7 日連続の勤務表示を禁止する制約，分庁舎勤務の連続割当を禁止する制約，稼働期間外の割当禁止，固定予定の反映等を課す．

4.3 ソフト制約と目的関数

望ましい条件として、係長級2名体制の確保、勤務負担（週あたり契約勤務日数からの偏差）の平準化、公休数（祝日休、土曜・日曜公休）の平準化、分庁舎勤務回数の平準化を扱う．各評価項目 $j \in J$ の未充足量を非負整数補助変数 u_j 、重みを ω_j とし、式(6)を最小化する．

$$\min \sum_{j \in J} \omega_j u_j \quad (6)$$

実装では、最大回数と最小回数の差や目標値からの絶対偏差を CP-SAT の最大値・最小値・絶対値制約で記述した．これらは補助変数と線形制約を用いることで混合整数線形計画へも置換できるため、本モデルは CP-SAT に固有の表現だけに依存するものではない．

5 二段階求解と不足箇所の提示

固定予定が特定の週に集中すると、必要人数や連続勤務制限を守りながら全職員の週公休条件を同時に満たす解が存在しない場合がある．すべての制約を一度にソフト化すると、「不足が0となった緩和解」と「厳密モデル自体が実行可能であった解」の区別が分かりにくい．そこで、第1段階では週公休を含む全ハード制約を厳密に課し、実行可能性を判定する（図3）．

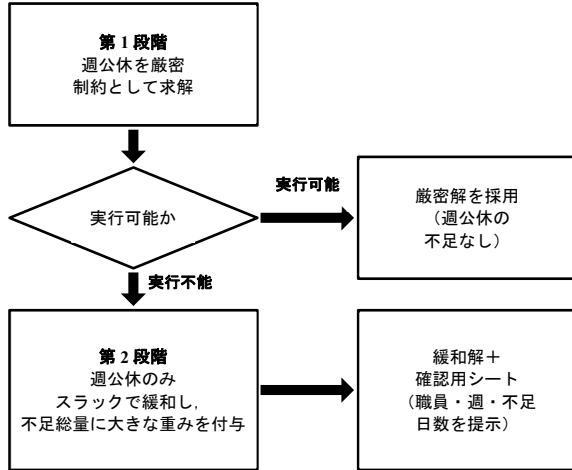


図3: 二段階求解の流れ

第1段階が実行不能の場合のみ第2段階へ移行し、どの職員のどの週で不足が生じるかを把握できるよう、職員 i 、週 t の公休不足を表す非負整数スラック $s_{i,t}$ を導入して式(5)を式(7)へ緩和する．

$$\begin{aligned} r_{i,t} - s_{i,t} &\leq \sum_{d \in A_{i,t}} (1 - w_{i,d}) \leq r_{i,t} \\ s_{i,t} &\geq 0 \end{aligned} \quad (7)$$

公休不足を強く抑制するため、大きな重み M を用いて式(8)を最小化する．本実装では $M = 10^9$ とした．これにより、公休不足に大きな罰則

を与えつつ、その他の評価項目も考慮した重み付き和最小化を行う．

$$\min \{ M \sum_{i,t} s_{i,t} + \sum_{j \in J} \omega_j u_j \} \quad (8)$$

第2段階で得られた解については、不足が生じた職員、対象週、必要公休数、実際の公休数、不足日数を確認用シートへ出力する．担当者は、固定予定の変更、必要人数の調整、公休の振替等を検討できる．

6 実運用データによる評価

6.1 評価条件

計画対象期間 2026 年 5 月～12 月（245 日）の実運用データを用いた．職員数は 14 名であり、モデル規模は変数約 2.1 万個、線形制約約 5.2 万本であった．求解には Google OR-Tools v9.14.6206 の CP-SAT を使い、各段階の探索時間上限を 600 秒、並列探索を 8 ワーカーに設定した．計算環境は Intel Core i7-1255U、メモリ 64 GB の Windows PC である．評価では、(1) 実運用設定での求解、(2) 二段階求解の動作確認、(3) 期間一括方式と月別分割方式の比較、(4) 勤務表作成時間の比較を行った．

6.2 求解結果

実運用設定では、計画期間 245 日、職員 14 名を対象とした約 2.1 万変数、約 5.2 万制約からなるモデルに対し、第1段階が 177 秒（3 分程度）で最適解を返し、週公休不足は生じなかった．二段階求解の動作を確認するため、同一データについてグループ会議日を実運用の木曜から月曜へ変更した．対象期間では月曜日に祝日が 5 日あり、会議日の振替によって固定予定が集中しやすい比較条件である．この条件では第1段階が約 3.7 秒で実行不能と判定され、第2段階が約 244 秒で最適解を返した．公休不足は延べ 5 日で、1 週に集中しており、職員・週単位で提示できた（表2）．

表2: グループ会議日設定による二段階求解の動作

項目	木曜（実運用）	月曜（比較用）
第1段階	最適（約 177 秒）	実行不能（約 3.7 秒）
第2段階	不要	最適（約 244 秒）
週公休不足	0 日	延べ 5 日（1 週）

6.3 月境界週の比較

同じ計画期間を月ごとに分割し、各月を独立に求解する方式と比較した．比較対象の月別分割方式では、月境界で分割された各部分週について、それぞれの月内で公休数を判定する従来方式を再現した．月別分割方式では、月境界を

またぐ6週のうち2週で、職員別の公休数を合計して延べ16日分の過剰な公休が生じた。期間一括方式では、境界週を一続きの7日間として評価するため、この過剰公休は生じなかった。この結果は、月別帳票を作成する業務であっても、最適化の計画期間は月境界を越えて設定する必要があることを示している。

6.4 業務時間の削減

1か月分の勤務表作成時間は、従来の12～18時間から約2～3時間へ減少し、およそ8割削減された(図4)。作成時間の短縮に加え、担当者の経験への依存が減少し、引継ぎの容易化が期待できる。残る時間は、入力データの確認、結果の妥当性確認、例外条件への対応等である。

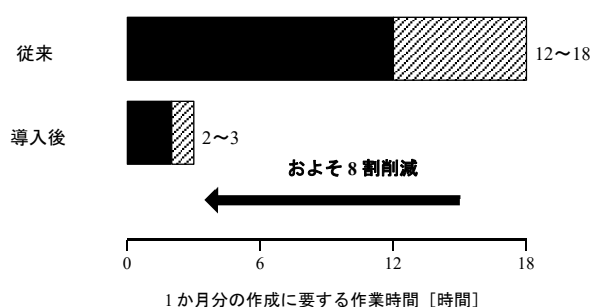


図4: 1か月分の勤務表作成時間の比較

7 考察

評価結果から、期間一括方式は月境界週における過剰公休を回避できることが確認された。また、二段階求解により、厳密条件下で実行可能解が存在しない場合にも、不足箇所を明示した代替案を提示できることを確認した。これらの結果は、勤務表の作成を自動化するだけでなく、担当者による調整作業を支援する仕組みとして本システムが有効であることを示している。

本研究の特徴は、自治体水道部門の業務条件を再現可能な整数制約モデルとして整理し、実運用に適用可能な形でシステム化した点にある。特に、勤務扱いと実働計上の分離により、研修・出張等を勤務日として表示しながら必要人数計算から除外する実務上の要件を表現できた。また、実行不能時にも不足箇所を具体的に提示することで、「週公休を含むハード制約を満たす通常解」と「公休不足を含み調整を要する代替案」を明確に区別し、人による最終調整を支援できる。確認用シートは、計算結果を単に提示するのではなく、人が最終判断を行う意思決定支援として機能する。

一方、本評価は一つの水道事業体を対象とした事例研究であり、職員規模や勤務制度が異な

る組織への一般化には追加検証が必要である。また、CP-SAT以外の混合整数計画ソルバーとの計算性能比較、重み設定の感度分析、入力変更時の安定性、担当者による手修正量の定量評価は今後の課題である。

8 おわりに

自治体水道部門の勤務表作成を対象に、期間一括最適化と二段階求解を備えた作成支援システムを開発し、吹田市水道部へ導入した。実運用データでは、245日・14名のモデルを数分で求解し、勤務表作成時間を約8割削減した。また、月境界週の一括評価により月別分割で生じる過剰公休を回避し、厳密モデルが実行不能となる場合には公休不足に大きな罰則を与えた緩和解と不足箇所を提示できた。本システムは勤務表を自動生成するだけでなく、実行不能時にも担当者による調整作業を支援する意思決定支援ツールとして機能することを示した。今後は、他組織への適用、ソルバー比較、運用後データを用いた継続的な評価を進める。

参考文献

- [1] Burke, E. K. et al.: The state of the art of nurse rostering, J. Sched., 7(6), 441–499 (2004).
- [2] Van den Bergh, J. et al.: Personnel scheduling: A literature review, Eur. J. Oper. Res., 226(3), 367–385 (2013).
- [3] Böðvardsdóttir, E. B. and Stidsen, T.: A review of multi-objective optimization methods for personnel rostering problems, J. Sched., 29(1), 1–19 (2026).
- [4] Google: OR-Tools CP-SAT Solver, https://developers.google.com/optimization/cp/cp_solver.