A Stochastic Programming Approach for Cyclic Personnel Scheduling with Double Shift Requirement

PATTARAPONG PAKPOOM
Kasetsart University
Department of Industrial Engineering
50 Ngamwongwan Rd.
Chatuchak, Bangkok, 10900
THAILAND
pattarapong.p@ku.ac.th

PEERAYUTH CHARNSETHIKUL
Kasetsart University
Department of Industrial Engineering
50 Ngamwongwan Rd.
Chatuchak, Bangkok, 10900
THAILAND
fengprc@ku.ac.th

Abstract: We study cyclic personnel scheduling with double shift requirement under demand uncertainty. The problem is formulated as a two-stage stochastic integer program with integer recourse. Solving it with commercial software CPLEX takes extended period of time. We explore an exact approach based on Benders decomposition technique and compare its performance to a heuristics approach based on genetic algorithm and a mip approach , solving the original problem by a commercial mip solver. A special solution method allow us to obtain optimal solutions to subproblems efficiently. This method are applicable to both exact and heuristic approaches which significantly accelerate overall solution process. Numerical results illustrate that the proposed approach, exact approach based on Benders decomposition technique, outperform the heuristics approach based on genetic algorithm and the CPLEX mip solver in all 16 instances.

Key–Words: Stochastic programming; Benders decomposition; Personnel scheduling; Genetic algorithm.

1 Introduction

Labor cost constitutes a major cost especially in service sectors such as health care, call centers, and toll booths. Businesses rely heavily on workforce management which plays a crucial role in determining success and survivability. Personnel scheduling is used to fill work periods with agents according to the demand. Other constraints or requirements could make the problems more complicate and cause difficulty to the solution process. Effective personnel scheduling methods can lead to significant reduction of labor cost.

Each time period requires certain amount of working agents. This demand can be predicted ahead of time. However, it is known without absolute certainty in many cases. The demand is confirmed just in a short time ahead of each period. To maintain quality of service after the actual demand is realized, the manager will need to hire extra workers in case of a demand shortage or cancel some assigned workers if they are oversupplied.

We divide up a time horizon in consideration into several time periods. This time horizon can be a day, a week, a month, or any other period of time. In many cases, the time horizon can be viewed as a cycle, meaning that the period after the last period can be considered as the first period. In such situations, actions during the last few periods may influence or have an effect on the first few periods. For example, if working a double shift is required and a worker starts working on the last period, he or she must work on the first period. Since our time horizon consists of one cycle, first period of the next cycle is considered the same as first period of the current cycle. Much research has been done in this context or similar situations.

Reference [5] provides a unified formulation for both deterministic and stochastic labor scheduling problems and solves them by a distributed genetic algorithm. This work influences us to compare our proposed algorithm to a genetic algorithm, for which research and examples can be found in [1] and [2]. Reference [3] investigates deterministic cyclic personnel scheduling of nurses using a Lagrangian-based heuristic. In [4], a real world scheduling problem of workforce planning at USPS processing and distribution centers is modeled and solved as stochas-

tic integer program yielding significant saving. Nurse scheduling from recorded data can be found in [6], [8], and [9]. In [11], personnel scheduling is modeled as MIP and solved by branch-and-price and heuristic algorithms. In [10], employee scheduling system is modeled for retail outlets and solve the problems using MIP. In [7], mixed-integer rounding inequalities are used to enhanced Benders decomposition technique to solve multiclass service system staffing and scheduling with arrival rate uncertainty. In [12] multi-cut L-shaped method is applied as their solution approach to solve multi-activity tour scheduling problem under demand uncertainty. All of the work mentioned above have similar stochastic personnel scheduling structure as our work in this paper. However, cyclic nature of the planning horizon and double shift requirements are not both considered together.

In this paper, we work on a problem of personnel scheduling of a system with uncertain demand where time horizon is considered as a cycle and double shift requirements are imposed. The manager may choose to let go some workers if a shift is overstaffed or must hire more workers to cover the shift in case of demand shortage. We formulate the problem as two-stage stochastic mixed integer programming and solve to minimize total operating cost. To our best knowledge, this problem structure and assumptions have never been considered in the literature. We test our proposed algorithm and compare it with genetic algorithm and CPLEX mip solver Our computational results show that our approach can speed up optimization process, in many instances, by more than 30 times comparing to solving the problems by the CPLEX MIP solver. When compare our proposed algorithm to genetic algorithm, our proposed algorithm not only have an ability to provide optimality gap and optimality proof but also find solutions of good quality faster than genetic algorithm in all cases.

2 Model Description and Formulation

Our cyclic scheduling horizon consists of N time periods. Quantity of service demanded for each period is a random variable with discrete probability distribution. Assuming all the workers' skill are comparable and employees must work two shifts back to back. This double shift requirement applies to initial scheduling as well as extra worker hiring and scheduling cancellation. The double shift requirement can be considered as a job condition or an agreement between employer and employee union. The objective is to minimize expected operating cost consisting of laboring cost plus expected hiring cost of extra workers when labor is short less expected shift canceling cost of surplus workers. In [5], a general model for stochastic personnel scheduling problems in extensive form is given. The followings are parameters for the model.

The model is given as the following.

$$\min \quad \sum_{e=1}^{E}\sum_{j=1}^{N_e} C_j X_j + \sum_{t=1}^{T}\sum_{k=1}^{W_t} \rho_{tk}(S_{tk}d_{tk}^{-} - O_{tk}d_{tk}^{+}) \qquad (1a)$$

s.t.

$$\sum_{e=1}^{E}\sum_{j=1}^{N_e} a_{ij}X_j + d_{tk}^{+} - d_{tk}^{-} = R_{tk}; \quad k = 1, \cdots, W_t, \forall t, \qquad (1b)$$

$$d_{t1}^{-} = 0, \forall t, \qquad (1c)$$

$$\sum_{j\in E_e} X_j \leq \sum_{e=1}^{E} \pi_e \sum_{j=1}^{N_e} X_j \forall e, \qquad (1d)$$

$$X_j \in \{0, 1, 2, \dots\}, \qquad (1e)$$

$$d_{tk}^{-}, d_{tk}^{+} \geq 0; pk = 2, \cdots, W_t; \forall t. \qquad (1f)$$

The objective (1a) is to minimize total operating cost, which includes hiring costs plus expected shortage costs level as less the expected benefit of oversupplied staff. Constraints (1b) ensures quality of service by maintaining the staff level at service demand level k. Constraints (1c) restricts the minimum number of workers. Constraints (1d) forces the number of employees of a certain type to be within some fraction of the total workers.

Mathematical model representing our problem can be derived from above model. In our model, since we assume that all workers are comparable, constraints (1d) are dropped. Constraints (1c) are also dropped since we will explicitly state the required number of workers for each scenario. On the decision variables, the number of employees assigned to schedule j, X_j, can be rewritten as x_i representing the number of employees assigned to start working on period i. Similarly, d_{tk}^{+} can be rewritten as u_i^{ω} representing the number of extra employees who are

Table 1: Parameters for a general model for stochastic personnel scheduling problems

Parameter	Description
M	The number of difference employee types, indexed $e = 1, \ldots, M$
E_e	The set of all possible schedules for employee type e for which $E_i \cap E_j = \varnothing$ for $i \neq j$
N_e	The total number of possible scheduling patterns for employee type e
p_e	The average productivity of type e employees
π_e	Type e employee maximum proportion
C_j	The wages of one employee assigned to schedule j when $j \in E_e$ for some e
T	The number of periods in the planning horizon, indexed t
a_{tj}	p_e if period t is in schedule j or 0 otherwise
W_t	The range of possible labor requirements needed for period t
R_{tk}	The quantity of labor required for satisfactory service during period t for service demand at level k, which is a strictly increasing function of k for each t
ρ_{tk}	The probability that the demand for labor will be at level k during period t
O_{tk}	Economic benefit per unit for excess staff during period t when service demand is at level k
S_{tk}	Unit cost of labor shortage during period t when service demand is at level k
Decision variables	
X_j	The number of employees assigned to schedule j
d_{tk}^+	The labor surplus for period t if demand is at level k
d_{tk}^-	The labor shortage for period t if demand id at level k2

assigned to start working on period i to cover demand shortage and d_{tk}^- can be rewritten as v_i^ω representing the number of employees assigned to start working on period i whose shift is canceled due to demand oversupplied. This results in the following model.

$$\min_{x,u,v} \quad \sum_{i\in I} c_i x_i + \sum_{i\in I}\sum_{\omega\in\Omega} (g_i^\omega u_i^\omega - h_i^\omega v_i^\omega) \quad (2a)$$

$$\text{s.t.} \quad x_i + x_{i+1} + u_i^\omega + u_{i+1}^\omega - v_i^\omega - v_{i+1}^\omega = d_{i+1}^\omega, \omega \in \Omega, \text{ for } i \leq N-1 \quad (2b)$$

$$x_N + x_1 + u_N^\omega + u_1^\omega - v_N^\omega - v_1^\omega = d_1^\omega, \omega \in \Omega \quad (2c)$$

$$\forall i \in I, \forall \omega \in \Omega, x_i, u_i^\omega, v_i^\omega \in \mathbb{N}. \quad (2d)$$

We will show efficient algorithm to solve subproblems of (2) when some additional requirements are imposed. This includes the number of total time period being odd and all possible demand being even. The model can be shown here.

$$\min_{x,u,v} \quad \sum_{i\in I} c_i x_i + \sum_{i\in I}\sum_{\omega\in\Omega} (g_i^\omega u_i^\omega - h_i^\omega v_i^\omega) \quad (3a)$$

$$\text{s.t.} \quad x_i + x_{i+1} + u_i^\omega + u_{i+1}^\omega - v_i^\omega - v_{i+1}^\omega = d_{i+1}^\omega, \omega \in \Omega, \text{ for } i \leq N-1 \quad (3b)$$

$$x_N + x_1 + u_N^\omega + u_1^\omega - v_N^\omega - v_1^\omega = d_1^\omega, \omega \in \Omega \quad (3c)$$

$$N \text{ is odd}, \forall i \in I, \forall \omega \in \Omega, x_i, u_i^\omega, v_i^\omega \in \mathbb{N}, d_i^\omega \in 2\mathbb{N}. \quad (3d)$$

For the rest of this work, we assume that the total number of time period N is odd. In the next section, we will show how to efficiently solve the problem.

3 Research Methodology

3.1 Benders Decomposition Algorithm

We would like to apply Benders decomposition to solve the problem. To fully take advantage of Benders decomposition methodology, we examine a special case of the above model. The following proposition considers (2) when all possible demands are even numbers.

Proposition 1. *Suppose that problem (3) is feasible then its RMIP has an integer optimal solution.*

Proof. The coefficient matrix of (3b)-(3c) has the following pattern.

$$\begin{bmatrix} A & A & -A & 0 & 0 & \dots & 0 & 0 \\ A & 0 & 0 & A & -A & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ A & 0 & 0 & 0 & 0 & \dots & A & -A \end{bmatrix}$$

where A has the following form.

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 & 1 \\ 1 & 0 & 0 & 0 & \dots & 0 & 1. \end{bmatrix}$$

When consider matrices as column vectors, they have a property called miltilinearity [16]. The determinant, as a function of the column vectors, is linear. That is,

$$det \begin{pmatrix} a & b+b' \\ c & d+d' \end{pmatrix} = det \begin{pmatrix} a & b \\ c & d \end{pmatrix} + det \begin{pmatrix} a & b' \\ c & d' \end{pmatrix} \tag{4}$$

Let A_1 and A_2 be matrices such that all of their columns but the last one are the same as those of A and last column of A_1 plus last column of A_2 equals last column of A. Then, by multilinearity property (4) we have the following.

$$det\left(A\right) = det\left(A_1\right) + det\left(A_2\right)$$

The following can be A_1 and A_2.

$$A_1 = \begin{bmatrix} 1 & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 & 1 \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 1 & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 & 1 \\ 1 & 0 & 0 & 0 & \dots & 0 & 0 \end{bmatrix}$$

The remaining of the proof is to show that A_1 and A_2 are TU matrices. We achieve that by dividing even rows of A_1 into one set and the rest into the other set. Similarly, we divide all even rows of A_2 plus the last row into one set and the rest into the other set. This shows that A_1 and A_2 are TU matrices.
Proof of the claim: let B be the optimal basis of (2). From Crammer's rule, $B^{-1} = B^*/det(B)$, where B^* is adjoint of B. Since B is integral, RHS is assumed to be even, and $det(B)$ is at most two, $(B^*/det(B)) * RHS$ is integral.

□

Proposition 1 allows us to solve MIP (3) by solving its integral relaxation as the following.

$$\begin{aligned} \min_{x,u,v} \quad & \sum_{i\in I} c_i x_i + \sum_{i\in I}\sum_{\omega\in\Omega} (g_i^\omega u_i^\omega - h_i^\omega v_i^\omega) && \text{(5a)} \\ \text{s.t.} \quad & x_i + x_{i+1} + u_i^\omega + u_{i+1}^\omega \\ & -v_i^\omega - v_{i+1}^\omega = d_{i+1}^\omega, i \le N-1, && \text{(5b)} \\ & x_N + x_1 + u_N^\omega + u_1^\omega \\ & -v_N^\omega - v_1^\omega = d_1^\omega, && \text{(5c)} \\ & x, u, v \ge 0, \forall\omega \in \Omega, d_i^\omega \in 2\mathbb{N}. && \text{(5d)} \end{aligned}$$

Hence, we can solve our original integer programming problem by linear programming. This is essentially what commercial software CPLEX does. Its optimization process only provides one integral solution which is the optimal solution to the problem.
Given $x^* \ge 0$, (5) becomes

$$\begin{aligned} \min_{u,v} \quad & \sum_{i\in I} c_i x_i^* + \sum_{i\in I}\sum_{\omega\in\Omega} (g_i^\omega u_i^\omega - h_i^\omega v_i^\omega) && \text{(6a)} \\ \text{s.t.} \quad & x_i^* + x_{i+1}^* + u_i^\omega + u_{i+1}^\omega \\ & -v_i^\omega - v_{i+1}^\omega = d_{i+1}^\omega, i \le N-1, && \text{(6b)} \\ & x_N^* + x_1^* + u_N^\omega + u_1^\omega \\ & -v_N^\omega - v_1^\omega = d_1^\omega, && \text{(6c)} \\ & u, v \ge 0, \forall\omega \in \Omega. && \text{(6d)} \end{aligned}$$

To construct a linear programming duality of (6), we associate each of its constraint with a dual variable y_i^ω and obtain the following.

$$\begin{aligned} \max_{y} \quad & \sum_{i\in I}\sum_{\omega\in\Omega} (d_{i+1}^\omega - x_i^* - x_{i+1}^*) y_i^\omega && \text{(7a)} \\ \text{s.t.} \quad & y_i^\omega + y_{i+1}^\omega \le g_{i+1}^\omega, i \le N-1, && \text{(7b)} \\ & -y_i^\omega - y_{i+1}^\omega \le -h_{i+1}^\omega, \\ & \qquad i \le N-1, && \text{(7c)} \\ & y_N^\omega + y_1^\omega \le g_1^\omega, && \text{(7d)} \\ & -y_N^\omega - y_1^\omega \le -h_1^\omega, && \text{(7e)} \\ & y_i^\omega \text{ urs}, \forall\omega \in \Omega. && \text{(7f)} \end{aligned}$$

Problem (7) can be rearranged as

$$\begin{aligned}
\max_{y} \quad & \sum_{i\in I}\sum_{\omega\in\Omega}(d_{i+1}^{\omega}-x_i^*-x_{i+1}^*)y_i^{\omega} && \text{(8a)}\\
\text{s.t.} \quad & h_{i+1}^{\omega}\le y_i^{\omega}+y_{i+1}^{\omega}\le g_{i+1}^{\omega}, \\
& \qquad i\le N-1, && \text{(8b)}\\
& h_1^{\omega}\le y_N^{\omega}+y_1^{\omega}\le g_1^{\omega}, && \text{(8c)}\\
& y_i^{\omega}\ \text{urs}, \forall\omega\in\Omega. && \text{(8d)}
\end{aligned}$$

To simplify the problem, coefficients of y_i^{ω} in (8) can be renamed as $\tilde{d_i^{\omega}}$, i.e.,

$$\tilde{d_i^{\omega}} := d_{i+1}^{\omega}-x_i^*-x_{i+1}^*. \tag{9}$$

Therefore, a more compact form of (8) is

$$\begin{aligned}
\max_{y} \quad & \sum_{i\in I}\sum_{\omega\in\Omega}\tilde{d_i^{\omega}}y_i^{\omega} && \text{(10a)}\\
\text{s.t.} \quad & h_{i+1}^{\omega}\le y_i^{\omega}+y_{i+1}^{\omega}\le g_{i+1}^{\omega}, \\
& \qquad i\le N-1, && \text{(10b)}\\
& h_1^{\omega}\le y_N^{\omega}+y_1^{\omega}\le g_1^{\omega}, && \text{(10c)}\\
& y_i^{\omega}\ \text{urs}, \forall\omega\in\Omega. && \text{(10d)}
\end{aligned}$$

To apply Benders decomposition technique to solve the original problem (2), a quick algorithm to find an optimal solution to (10) would be beneficial. To develop such an algorithm, we reformulate (10) using new variables.

$$\begin{aligned}
\alpha_1^{\omega} &:= y_N^{\omega}+y_1^{\omega}, && \text{(11a)}\\
\alpha_{i+1}^{\omega} &:= y_i^{\omega}+y_{i+1}^{\omega}, \\
& \qquad i\le N-1. && \text{(11b)}
\end{aligned}$$

Also, let

$$\begin{aligned}
k_i^{\omega} \quad := \quad & (\tilde{d_i^{\omega}}+\sum_{j\ge i+1}(-1)^{j+i+1}\tilde{d_j^{\omega}}+ \\
& \sum_{j\le i-1}(-1)^{j+i}\tilde{d_j^{\omega}})/2. && \text{(12)}
\end{aligned}$$

Putting this together, we have the following problem equivalent to (8)

$$\begin{aligned}
\max_{\alpha} \quad & \sum_{i\in I}\sum_{\omega\in\Omega}k_i^{\omega}\alpha_i^{\omega} && \text{(13a)}\\
\text{s.t.} \quad & h_i^{\omega}\le\alpha_i^{\omega}\le g_i^{\omega}, \\
& \qquad \forall i\in I,\omega\in\Omega && \text{(13b)}
\end{aligned}$$

Solution to (13) can be obtained from the following proposition.

Proposition 2.

$$(\alpha_i^{\omega})^* = \begin{cases} g_i^{\omega}, \textit{ if} \quad k_i^{\omega}>0 \\ h_i^{\omega}, \textit{ otherwise.} \end{cases} \tag{14}$$

is an optimal solution to (13).

Proof. From (13), we associate constraints $\alpha_i^{\omega}\le g_i^{\omega}$ with β_i^{ω} and constraints $h_i^{\omega}\le\alpha_i^{\omega}$ with $\bar{\beta}_i^{\omega}$. The duality is as follow.

$$\begin{aligned}
\min_{\beta,\bar{\beta}} \quad & \sum_{i\in I}\sum_{\omega\in\Omega}-h_i^{\omega}\bar{\beta}_i^{\omega}+g_i^{\omega}\beta_i^{\omega} && \text{(15a)}\\
\text{s.t.} \quad & \beta_i^{\omega}-\bar{\beta}_i^{\omega}=k_i^{\omega}, \forall i\in I,\omega\in\Omega && \text{(15b)}\\
& \bar{\beta}_i^{\omega},\beta_i^{\omega}\ge 0, \quad \forall i\in I,\omega\in\Omega. && \text{(15c)}
\end{aligned}$$

We can check that the following is an optimal solution of (15).

$$(\beta_i^{\omega})^* = \begin{cases} k_i^{\omega} & \text{, if} \quad k_i^{\omega}>0 \\ 0 & \text{, otherwise} \end{cases} \tag{16}$$

and

$$(\bar{\beta}_i^{\omega})^* = \begin{cases} -k_i^{\omega} & \text{, if} \quad k_i^{\omega}>0 \\ 0 & \text{, otherwise} \end{cases} \tag{17}$$

This completes the proof as we check that all conditions of strong duality theorem are satisfied. □

Proposition 3. *(8) and (13) are equivalent. Also, an optimal solution to (8) is*

$$\begin{aligned}
(y_i^{\omega})^* = [(\alpha_i^{\omega})^* & +\sum_{j\ge i+1}(-1)^{j+i}(\alpha_j^{\omega})^* \\
& +\sum_{j\le i-1}(-1)^{j+i+1}(\alpha_j^{\omega})^*]/2. && \text{(18)}
\end{aligned}$$

Proof. To show that (8) and (13) are equivalent, we only need to show that (12) is a correct transformation of (8) to (13). We can show (18) by mathematical manipulation. □

Given Q as the set of all extreme points $\tilde{y}$ of (7), we can reformulate problem (5) as:

$$\begin{aligned}
\min \quad & z+\sum_{i\in I}c_ix_i && \text{(19a)}\\
\text{s.t.} \quad & z\ge(d-Ax)^{\intercal}\tilde{y} && \text{(19b)}\\
& \tilde{y}\in Q, x\ge 0 && \text{(19c)}
\end{aligned}$$

, where $d - Ax = [d_{i+1}^{\omega} - x_i - x_{i+1}]$ and $\tilde{y} = [y_i^{\omega}]$. The constraint

$$z \geq (d - Ax)^{\intercal} y^* \tag{20}$$

for a given y^* is referred to as a Benders' Cut [17].

Let $Q' \subset Q$, then the following is a relaxation of (19):

$$\begin{aligned} \min \quad & z + \sum_{i \in I} c_i x_i && \text{(21a)} \\ \text{s.t.} \quad & z \geq (d - Ax)^{\intercal} \tilde{y} && \text{(21b)} \\ & \tilde{y} \in Q', x \geq 0. && \text{(21c)} \end{aligned}$$

Using above transformation of original problem (5), we apply Bendersdecomposition algorithm to our problem as follows.

BendersDecomposition Iterations

Initialization:

1. Let x^* in (5) be some positive number.
2. Set UB = ∞ and LB = $-\infty$.
3. Let $Q' = \varnothing$.

Step 1:

1. Obtain optimal solution y^* to (8) from proposition 3.
2. Update the UB yielded by (8).
3. Add y^* to Q' (hence, add a Benders cut, to (21)).

Step 2:

1. Solve (21) for x^*.
2. Update the LB.
   - If $|UB - LB| < \epsilon$ then exit and declare x^* as an optimal solution to (5).
   - If the number of iteration exceeds iteration limit then exit and declare x^* as a feasible solution to (5), otherwise, go to step 1.

3.2 Genetic Algorithm (GA)

Performance of our exact method, Bendersdecomposition approach, is tested in comparison to a heuristic method. A genetic algorithm chosen as an inexact method to solve our stochastic labor scheduling problems was successfully demonstrated in [5]. In that work, the author apply genetic algorithm to solve stochastic labor scheduling problems on parallel computer network. The process is called distributed genetic algorithm (DGA). We make a small change to their procedure to solve our problem on a single computer.

One of the most crucial parts of genetic algorithm is the design of fitness function which distinguishes good solutions from bad ones and ranks all solutions by their quality. Since, the objective of our personnel scheduling problem is to minimize total labor cost, the best fitness function for our genetic algorithm is inverse of this objective function. A fitter candidate, which has a higher fitness function value, corresponds to a personnel schedule with lower total labor cost. However, given feasible first stage solution x_i, finding its complement, the second stage solution, is normally computationally expensive. This can be done by solving the MIP to optimality. Fortunately, our work of Benders decomposition method suggests a viable alternative. Given x_i, we can find the second stage solution and and its objective function from (18) efficiently. Our genetic algorithm makes up of the following components.

1. Population
   Initial small subset called deme of population are generated with values nearby the demand. Each individual represent first stage solution, a vector of positive integers x_i, to the problem. This solution has a ring configuration as our time horizon is considered cyclic. We create left and right demes from this initial solution as $x_i^L = x_{i+1}$ and $x_i^R = x_{i-1}$, respectively. Each iteration of the genetic algorithm maintains these three demes, or a neighborhood, with individuals evolve over time. One iteration of GA provides a generation of the population.
2. Fitness function and hill climbing
   Fitness function is the inverse of the objective value of each individual. The objective value is obtained from (8) for each solution. Hill climbing process, adapted from [14] and [15] by local search technique, is applied to

40% of new solutions at the beginning of each iteration to refine solution quality. The change to each x_i of ± 1 is applied if the new solution is fitter.

3. Reproduction selection
This process involves chosen individuals only within each deme called a mating pool. An application of elitist [13] and biased sample will be applied to form a mating pool by ranking individuals in the ascending manner with equally spaced score of .8 to 1.2 according to their fitness score, selecting the ones with scores at least one to put in the mating pool then reduce their scores by one, and then normalized remaining scores to use them as probability to fill in the remaining spots in the mating pool.

4. Crossover and mutation
Randomly paired up two individuals in each mating pool proceed crossover process. With 0.7 coefficient for crossover, 70% of all chosen pairs undergo crossover operation while the other 30% are clones , remaining untouched during this process. Crossover process randomly picks starting and ending points for gene to switch. This corresponds to X_i, X_{i+1}, $\cdots X_{i+k}$ for a length k crossover genes which can possibly rollover to X_1 as our solution is cyclic. In this length k genes, one gene is randomly pick and mutated with mutation coefficient of 0.2, or 20% chance, to ensure population diversity. If the new best fit solution is less fit than the former best fit, replace the new least fit individual with the former best fit individual.

5. Interdeme migration
At the end of each iteration of evolving by this genetic algorithm, the least fit individual within each deme is replaced by the best fit individual from the other two demes within the neighborhood, if the fitness value is better.

We apply the genetic algorithm described above to initial population to evolve for ten generations.

4 Numerical Results

We generate test instances by generating parameters g_i^ω, h_i^ω, and d_i^ω. Employee wages per shift are assumed to be equal for everyone. Instance size suitable for our proposed algorithm and the computational power is $N = 11$. Each instance differs by demands and demand variations. The demand variations define the number of possible events $|\Omega|$ for our stochastic model. We implemented all algorithms in GAMS [19] using CPLEX solver [18] on a Windows 10 Pro personal computer with 2.20GHz Intel Core i5 CPUs and 12 GB memory.

Table 2 shows size of our test instances in term of number of variables and number of constraints. Table 3 shows number of Benders iterations needed to achieve optimal solution.

Table 2: Instance size as number of columns (# Variables)and rows (# Constraints)

Instance	# Constraints	# Variables
002	1,368,576	2,737,163
003	1,520,640	3,041,291
004	1,710,720	3,421,451
005	684,288	1,368,587
006	1,026,432	2,052,875
007	4,105,728	8,211,467
008	2,737,152	5,474,315
009	1,425,600	2,851,211
010a	2,138,400	4,276,811
010b	2,138,400	4,276,811
011	2,027,520	4,055,051
012	2,376,000	4,752,011
013	1,013,760	2,027,531
014	2,433,024	4,866,059
015	912,384	1,824,779
016	2,566,080	5,132,171

Tables 4 and 5 show comparison of solution time between our proposed algorithm, genetic algorithm, and CPLEX MIP solver. It is clear that on all instances, solving the problem by MIP take significantly longer than solving the problem by our proposed algorithm. Our genetic algorithm, which consists of evolving for ten generations, takes about three time longer than the proposed algorithm. Optimal solutions are found on 10 out of 16 tested problems. The other 6 problems have optimality gap of less than 0.2%. In all 16 cases as shown in table 4, the proposed algorithm performs better than genetic algorithm.

These four instances represent similar trends of the performance of all of our test instances in figures 1, 2, 3, and 4. These figures show solution quality of genetic algorithm of each generation for a total of ten generations.

We can see that the genetic algorithm per-

Table 3: Problem size as number of demand scenario (# events) comparing to number of Bendersiterations needed to achieve an optimal solution

Instance	# events ($\|\Omega\|$)	# BendersIterations
002	124,416	96
003	138,240	111
004	155,520	115
005	62,208	94
006	93,312	110
007	373,248	96
008	248,832	109
009	129,600	104
010a	194,400	90
010b	194,400	92
011	184,320	101
012	216,000	104
013	92,160	91
014	221,184	106
015	82,944	107
016	233,280	104

Table 4: Comparison of solution time between our proposed algorithm and Genetic Algorithm

Instance	Time on Proposed Algorithm (hh:mm:ss)	Time on GA (hh:mm:ss)
002	1:03:51	3:37:55
003	1:31:41	3:51:57*
004	1:51:22	4:44:40
005	0:30:54	1:55:25
006	1:00:34	2:21:22
007	5:54:18	14:29:22
008	2:42:13	8:23:21*
009	1:16:48	4:14:04
010a	1:27:29	6:49:55*
010b	1:30:18	6:50:27*
011	1:42:35	5:20:29
012	2:06:39	7:26:53*
013	0:42:01	2:48:05
014	2:13:23	3:59:36
015	0:50:38	2:15:22*
016	2:17:01	6:25:54

Symbol * indicates that solution is not optimal but is within 0.2% optimality gap.

Table 5: Comparison of solution time between our proposed algorithm and CPLEX MIP solver

Instance	Time on Proposed Algorithm (hh:mm:ss)	Time on MIP (hh:mm:ss)
002	1:03:51	36:28:45
003	1:31:41	48:00:36
004	1:51:22	61:35:45
005	0:30:54	7:05:49
006	1:00:34	23:48:23
007	5:54:18	-
008	2:42:13	-
009	1:16:48	38:05:49
010a	1:27:29	-
010b	1:30:18	-
011	1:42:35	-
012	2:06:39	-
013	0:42:01	18:20:27
014	2:13:23	-
015	0:50:38	14:17:14
016	2:17:01	-

Symbol - indicates that solution time for the method is more than 72 hours.

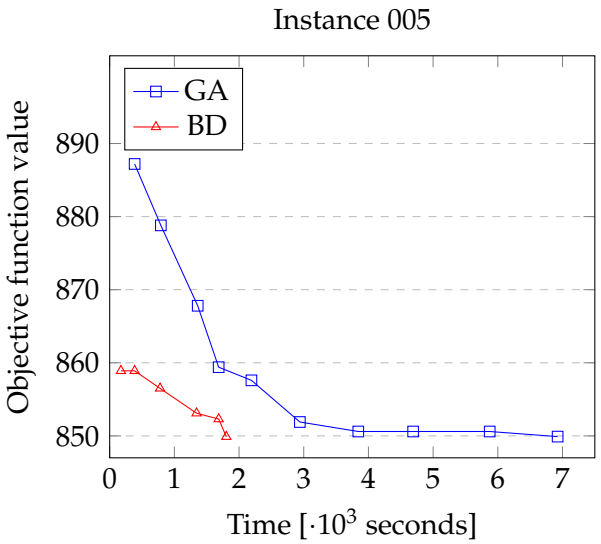

Figure 1: Performance comparison between genetic algorithm (GA) and proposed algorithm, Benders decomposition based algorithm (BD), on instance 005

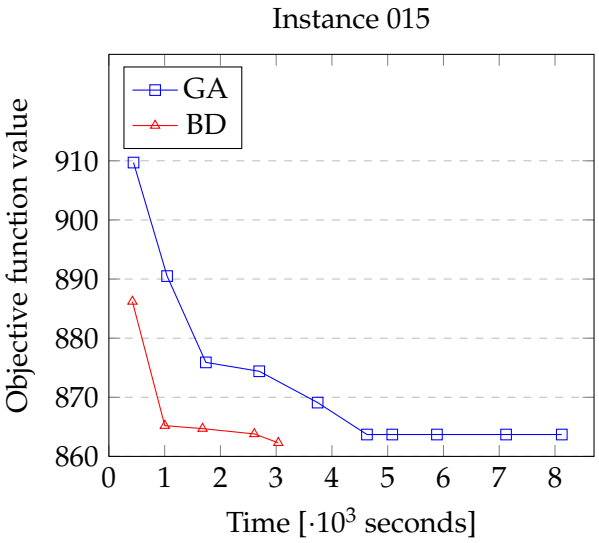

Figure 2: Performance comparison between GA and proposed algorithm on instance 015

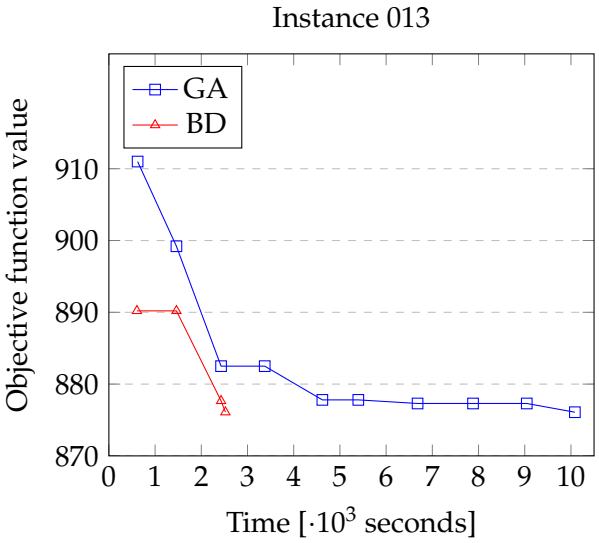

Figure 3: Performance comparison between GA and proposed algorithm on instance 013

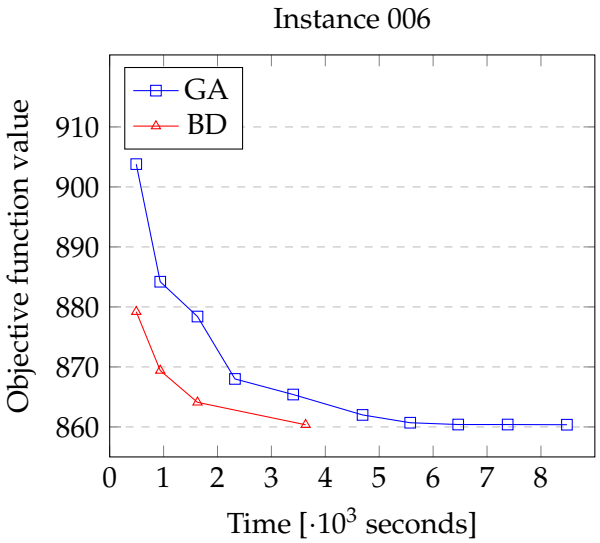

Figure 4: Performance comparison between GA and proposed algorithm on instance 006

forms worse than our proposed Bendersdecomposition algorithm. However, all problems can achieve optimal solutions or solutions within 0.2% optimality gap tolerance in ten generations.

5 Conclusions And Discussions

Our methods of solving cyclic personnel scheduling with uncertain demand and double shift requirements combine special solution method with Benders decomposition. The result is a quick and exact solution algorithm for our test instances which performs much better in all tested instances when we compare this to the CPLEX MIP solver. The proposed algorithm also outperforms genetic algorithm. However, the genetic algorithm has potential in parallel optimization application which should be examined in the near future especially for other cases or problems with less restrictions. Since our test instances only cover cases with certain assumptions, all other cases could be investigated in upcoming research.

References:

[1] N.E. Mastorakis, Unstable Ordinary differential equations: Solution via genetic algorithms and the method of Nelder-Mead, *Proc. WSEAS Int. Conf. Sys. Thorem. Sci. Comp*, 2006, pp. 1–6.

[2] N.E. Mastorakis, Solving non-linear equations via genetic algorithms,

Proc. WSEAS Int. Conf. Evol. Comp., 2005, pp. 24–28.

[3] J.F. Bard and H.W. Purnomo, Cyclic preference scheduling of nurses using a Lagrangian-based heuristic, *J. Sche.* 10(1), 2007, pp. 5–23.

[4] J.F. Bard, D.P. Morton and Y.M. Wang, Workforce planning at USPS mail processing and distribution centers using stochastic optimization, *Anal. Oper. Res.* 155(1), 2007, pp. 51–78.

[5] F.F. Easton and N. Mansour, A distributed genetic algorithm for deterministic and stochastic labor scheduling problems, *Eu. J. Oper. Res.* 118(3), 1999, pp. 505–523.

[6] P. Punnakitikashem, J.M. Rosenberber, and D.F. Buckley-Behan, A stochastic programming approach for integrated nurse staffing and assignment, *IIE Tran.* 45(10), 2013, pp. 1059–1076.

[7] M. Bodur and J.R. Luedtke, Mixed-integer rounding enhanced benders decomposition for multiclass service-system staffing and scheduling with arrival rate uncertainty, *Man. Sci.* 63(7), 2016, pp. 2073–2091.

[8] H. Huang, W. Lin, Z. Lin, Z. Hao and A. Lim, An evolutionary algorithm based on constraint set partitioning for nurse rostering problems, *Neur. Comp. App.* 25(3-4)2014, pp. 703–715.

[9] K. Kim and S. Mehrotra, A two-stage stochastic integer programming approach to integrated staffing and scheduling with application to nurse management, *Oper. Res.* 63(6), 2015, pp. 1431–1451.

[10] A. Parisio and C.N Jones, A two-stage stochastic programming approach to employee scheduling in retail outlets with uncertain demand, *Omeg,* 53, 2015, pp. 97–103.

[11] M.I. Restrepo, L. Lozano and A.L. Medaglia, Constrained network-based column generation for the multi-activity shift scheduling problem, *Int. J. Pro. Econ* 140(1), 2012, pp. 466–472.

[12] M.I. Restrepo, B. Gendron and L.M. Rousseau, A two-stage stochastic programming approach for multi-activity tour scheduling, *Eu. J. Oper. Res.* 262(2), 2017, pp. 620–635.

[13] J.E. Baker, Adaptive selection methods for genetic algorithms, *Proc. Int. Con. Gen. Alg.*, Hillsdale, New–Jersey, 1985, pp. 101–111.

[14] L. Davis, Handbook of genetic algorithms, CUMINCAD, 1991.

[15] C.L. Huntley and D.E. Brown, A parallel heuristic for quadratic assignment problems, *Comp. Oper. Res.* Elsevier, 18(3), 1991, pp. 275–289.

[16] P.B. Garrett, Abstract algebra, CRC Press,2007.

[17] J.F. Benders, Partitioning procedures for solving mixed-variables programming problems, *Num Math.* Springer, 4(1), 1962, pp. 238–252.

[18] IBM ILOG, CPLEX Release 12.6.3.0, 2016.

[19] GAMS Development Corporation, General Algebraic Modeling System (GAMS) Release 24.7.4, Washington, DC, 2016.