

Optimization Models for Hybrid Work Scheduling

Vardges Melkonian

Department of Mathematics, Ohio University, Athens, USA

Email: melkonian@ohio.edu

How to cite this paper: Melkonian, V. (2023) Optimization Models for Hybrid Work Scheduling. *American Journal of Operations Research*, 13, 147-176. <https://doi.org/10.4236/ajor.2023.136009>

Received: September 29, 2023

Accepted: November 5, 2023

Published: November 8, 2023

Copyright © 2023 by author(s) and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Remote and Hybrid work has been a common practice for many organizations in recent years. It has many advantages such as offering a better work-life balance but it might also negatively affect productivity and teamwork. While an organization would like to satisfy the remote/hybrid preferences of its employees, it also must ensure that there are enough people working in the office to satisfy certain professional needs. Finding the right balance between in-office and remote work is not an easy task. We develop three optimization models to give solutions to the problem. The most comprehensive model allows employees to work remotely some days of the week and flexible hours for those weekdays when employees work in the office. Our computational results show that the models are very time-efficient in practice. The computational results also include a sensitivity analysis of the most comprehensive model.

Keywords

Integer Linear Programming, Work Scheduling, Remote Work, Hybrid Work

1. Introduction

Main idea of our work.

The COVID pandemic set a new reality for the workplace. During the pandemic, most people had to work remotely. That experience showed that many activities can be done effectively without being in the office. Remote work certainly has some advantages for many employees by saving them commute time and offering a better work-life balance. But working in the office has its own advantages, such as offering more connectedness to coworkers, higher motivation and productivity. It is widely accepted that for many organizations and employees, the future is in hybrid work [1], finding the right balance between in-office and remote work.

When planning for remote/hybrid work, an organization might take into account employee preferences but also should ensure that there are enough people working in the office to satisfy certain professional needs of the organization. Finding the right balance between in-office and remote work might not be an easy task, even for small companies. In this paper, we build optimization models to offer solutions for the problem.

Advantages and disadvantages of remote/hybrid work.

The main idea of our models is that there are conflicting factors for and against working remotely or in hybrid mode, and the models try to find the right balance for the company. Many structures of our models are based on those conflicting factors which are listed below.

Advantages of remote work.

- It is possible to have *smaller salaries* for those employees who want and are allowed to work remotely. Those employees themselves save money by working remotely: they save on commuting and transportation and might be able to save on rent/mortgage/taxes when having more flexibility to choose the location of their places of residence. In return for that flexibility, those employees might be willing to work for the company for less salary.
- In some situations, an employer could have two choices to keep a good employee: either increase the salary or let them work remotely. Thus, letting work remotely saves a salary expense for the company.
- If a company allows remote work then it might be able to hire professionals with high qualifications who live in places different from the company's location and are unwilling to relocate. Conversely, if a company forces employees to work in the office then some employees might leave the company [1].
- Employees working remotely will save on *commuting time* to have more time for work.
- Even if an employee must work in the office, allowing *flexible hours* will allow to avoid rush hours and thus save more time.
- Remote/hybrid offers a *better work-life balance*.
- Having some employees working remotely will allow a company to save money by *paying less for office space and maintenance*.

Disadvantages of remote work.

- Some company operations need to be done or are more effective when done in the office. Some examples are orientation and training of new employees.
- Offices might have more resources available to make the work more productive.
- Not being in a workplace might negatively affect productivity, teamwork and motivation [1]. Employees might feel more connected when having in-person interactions with other employees in the office.
- Remote projects might take longer.

The advantages and disadvantages listed above are mostly supported by the findings of the Gallup survey [1]. Since remote work has both advantages and disadvantages, it is important to have a balance between in-office and remote

work, rather than completely switching to remote work or not allowing remote work at all. Thus, hybrid work could be a good tradeoff between the two extreme options.

General description of our models.

Based on the arguments above, our models assume that the cost of those employees who prefer working remotely is smaller if they are allowed to work remotely. On the other hand, there are some professional activities that should be performed in the office. Thus, the main idea of our *first basic model* is to maximize salary-related savings subject to the constraint that there should be enough people working in the office to satisfy certain professional needs. The model decides whether each employee should be allowed to work remotely or not.

Our *second model* allows the possibility of working in hybrid mode. More specifically, an employee might be able to work some weekdays remotely while being in the office the rest of the week. The model assumes that the company's professional needs and employees' preferences about remote work are different for different weekdays.

Our *third model* allows more flexibility with hybrid work. If an employee must be in the office on a certain weekday, the model allows them to have flexible hours for that particular day. An employee can be in the office not for the whole workday but in a shorter time window. That flexibility saves time by helping to avoid rush hours and offers a better work-life balance.

Related literature.

Workforce optimization has been extensively studied before. [2] [3] [4] give surveys of personnel scheduling. [5] [6] give models for flexible staff scheduling during the Covid pandemic. [7] gives scheduling models that take into account employee preferences.

The current state and future of remote/hybrid work is a topic of active discussion and research. There are surveys that show what employees think about the issue. Particularly, a Gallup survey [1] finds that hybrid work is both the most preferred and anticipated work mode. Based on the survey, 32% of employees prefer working exclusively remotely and 59% prefer working in hybrid mode (while their anticipated locations are 24% and 53% respectively). Some datasets in our computational results are based on the findings of the Gallup survey.

The structure of the paper.

Our three models, the Basic Model, the Model with Hybrid Option, and the Model with Flexible Hours are presented in Sections 2, 3, and 4 respectively. Section 5 gives a computational analysis of the models. Some conclusions and future directions are discussed in Section 6. Full implementations of the models with some computational results are given in the **Appendix Section**.

2. Basic Model (Model 1)

The main idea of the model is the following. A company wants to maximize salary savings by letting some employees work remotely subject to constraints that

some company needs must be satisfied by having a certain number of employees working in the office.

Below we define the mathematical structures we need for the model. Sets and parameters, variables, objective function and constraints are given in Subsections 2.1 - 2.4 correspondingly. Sets and parameters are the input to the model. The output of the model is a combination of values for the variables (unknowns) that optimize the objective function subject to the constraints. Those structures are defined using the syntax of the mathematical modeling language AMPL [8]. The complete AMPL model is given in Subsection 2.5.

2.1. Sets and Parameters

The input data we need for the model are divided into three categories: Employee preference-related data, Company professional needs data, and Salary savings-related data. We give data definitions and descriptions for each category below.

1) *Employee preference related data*

The first set represents the set of company employees.

set employees; (S2.1)

In the basic model, there are two types of employees: those who prefer working remotely and those who do not have such a preference.

param remote_preference {employees} binary; (P2.1)

This parameter is binary. It is assigned a value 1 if the employee prefers to work remotely, and 0 if there is no such preference.

2) *Company professional needs data*

The company has a set of professional needs, and some of them should be done in the office.

set company_needs; (S2.2)

For each professional need, we define a parameter `need_in_office` which indicates the minimum number of employees who can fulfill the need and should work in office.

param need_in_office {company_needs}; (P2.3)

On the other hand, each employee has a set of skills and can fulfill different needs of the company. We define a binary parameter `employee_need` which takes value 1 if the employee can fulfill the specific need, and 0 otherwise.

param employee_need{employees, company_needs}; (P2.4)

3) *Salary savings related data*

The main idea of our model is that there might be a cost-saving for those employees who prefer working remotely and are allowed to do so. Salary savings can be for any employee who works remotely, not just those who prefer working remotely. But the savings are significantly higher for those who prefer working remotely; for example, they might be more willing to take salary cuts in order to work remotely. On the other hand, if an employee wants to work in the office or is indifferent then it is hard to negotiate salary savings by letting them work remotely.

Thus, we define the following parameter indicating how much salary could be saved for employees who prefer working remotely and are allowed to do so.

param salary_saving{e in employees: remote_preference[e] = 1}; (P2.5)

The savings should be estimated by the company for each employee separately. It could be proportionate to the salary and dependent on the skillset of the employee.

2.2. Variables

The company needs to make decisions about allowing employees to work remotely. Thus, we define a binary decision variable for each employee; it takes a value 1 if the employee is allowed to work exclusively remotely and 0 otherwise.

var employee_remote{e in employees} binary; (V2.1)

2.3. Objective Function

The objective function is maximizing the total salary savings for those employees who prefer working remotely (that is, **remote_preference[e] = 1**) and are allowed to do so (that is, **employee_remote[e] = 1**).

maximize salary_savings: sum{e in employees: remote_preference[e] = 1} salary_saving[e]*employee_remote[e]; (O2.1)

2.4. Constraints

The basic model has only one set of constraints. It provides that for each company need c , the number of employees who work in the office and can fulfill need c is at least $need_in_office[c]$.

subject to company_professional_needs{c in company_needs}:

sum{e in employees}

employee_need[e,c]*(1-employee_remote[e]) >= need_in_office[c]; (C2.1)

The constraint works the following way. Note that the product **employee_need[e,c]*(1-employee_remote[e])** can take only values 0 or 1. It equals 1 if and only if **employee_need[e,c] = 1** (that is, employee e can fulfill need c) and **employee_remote[e] = 0** (that is, employee e works in the office). Thus, the summation in the left-hand side of (C2.1) equals the number of employees who work in the office and can fulfill need c ; while the right-hand side requires that the number of that kind of employees is at least $need_in_office[c]$.

2.5. Complete AMPL Model with a Sample Data Set

EMPLOYEE PREFERENCE DATA

set employees; (S2.1)

param remote_preference {employees}; (P2.1)

takes two possible values: 1 if prefers to work remotely, 0 otherwise

COMPANY PROFESSIONAL NEEDS DATA

set company_needs; (S2.2)

company has professional needs that should be done in office

```

param need_in_office{company_needs}; (P2.2)
# minimum number of employees with the need who should work in office
param employee_need{e in employees, c in company_needs}; (P2.3)
# is 1 if the employee e can fulfill need c
### SALARY SAVINGS RELATED DATA ###
param salary_saving{e in employees: remote_preference[e] = 1}; (P2.4)
# salary saving amount for the employees who prefer working remotely
##### VARIABLES #####
var employee_remote{e in employees} binary; (V2.1)
# is equal 1 if the employee works remotely
##### OBJECTIVE FUNCTION #####
maximize Total_Savings: sum{e in employees: remote_preference[e] = 1} salary_saving[e]*employee_remote[e]; (O2.1)
##### CONSTRAINTS #####
subject to company_professional_needs{c in company_needs}:
sum{e in employees}
employee_need[e,c]*(1-employee_remote[e]) >= need_in_office[c]; (C2.1)
# number of employees work in office and can fulfill need c is at least need_in_office[c]
data;
### EMPLOYEE PREFERENCE DATA ###
set employees:= 1 2 3 4 5 6 7 8 9 10;
param remote_preference :=
1    1
2    0
3    0
4    0
5    1
6    0
7    1
8    0
9    0
10   0;
### COMPANY NEEDS DATA ###
set company_needs := 1 2 3;
param need_in_office :=
1    3
2    2
3    3;
param employee_need:
      1    2    3 :=
1    1    0    1
2    0    1    1
3    0    1    0

```

```

4    1    1    0
5    0    0    1
6    0    1    0
7    1    1    1
8    1    0    1
9    0    0    1
10   1    0    0;
### SALARY SAVINGS RELATED DATA ###
param salary_saving :=
1    2
5    3
7    1;

```

3. Model with Hybrid Work Option (Model 2)

In this section, we develop the basic model further to have employees who prefer working in a hybrid mode. More specifically, an employee might want to work several days a week remotely and the other days in the office.

The new model needs new structures. Also, some of the structures defined in the basic model are updated.

3.1. Sets and Parameters

Input data for this model are divided into four categories: Time periods data, Employee preference related data, Company professional needs data, and Salary savings related data. We give definitions and descriptions for each category below.

1) *Time periods data*

For hybrid work, we need to define a set of weekdays. For convenience, we use 1, 2, 3, 4, 5 for Monday, Tuesday, Wednesday, Thursday, and Friday correspondingly.

set weekdays := 1 2 3 4 5; (S3.1)

2) *Employee preference related data*

We still have a set of employees for a company.

set employees; (S3.2)

Employees might have three different preferences in this new model: exclusively remote work, hybrid work, and no preference for remote work.

param remote_preference {employees}; (P3.1)

The parameter takes three possible values: 2 if the employee prefers to work remotely, 1 if the employee prefers to work in hybrid mode, 0 otherwise.

Employees with hybrid or remote work preferences specify the minimum and maximum number of weekdays they would like to work remotely. Note that an employee who prefers working only remotely might not be able to do so. But they will at least be granted to work remotely several days per week. Thus, we have the following two new parameters.

- Minimum number of days an employee with hybrid or remote preference

would like to work remotely:

param min_days_remotely
{e in employees: remote_preference[e] = 1 or remote_preference[e] = 2};
(P3.2)

- Maximum number of days an employee with hybrid or remote preference would like to work remotely:

param max_days_remotely
{e in employees: remote_preference[e] = 1 or remote_preference[e] = 2};
(P3.3)

This parameter should be set to 5 if an employee prefers working exclusively remotely.

3) *Company professional needs data*

set company_needs; **(S3.3)**

This set is the same as in the basic model.

The professional needs data are modified as follows. For each professional need, the parameter *need_in_office* indicates the minimum number of employees who can fulfill the need and should work in office *on a given weekday*. (It was a general needs requirement in the basic model.)

param need_in_office{company_needs, weekdays}; **(P3.4)**

param employee_need{employees, company_needs}; **(P3.5)**

This parameter is the same as in the basic model.

4) *Salary savings related data*

Salary savings data are also modified for the hybrid work. The company has savings for each weekday an employee works remotely.

param salary_saving_daily
{e in employees: remote_preference[e] = 1 or remote_preference[e] = 2};
(P3.6)

The parameter for the total salary saving amount for the employees who prefer working exclusively remotely remains the same.

param salary_saving_remote{e in employees: remote_preference[e] = 2};
(P3.7)

The amount of salary savings is proportional to the number of days an employee is allowed to work remotely. But as discussed in the Introduction, the company will get extra savings from those employees who prefer working only remotely and are granted to do so.

3.2. Variables

The variables are modified to reflect the fact that an employee might work in office some days and remotely other days of week.

For each employee *e* and each weekday *d*, we define a binary variable which takes value 1 if employee *e* works remotely in weekday *d*.

var employee_remote_weekday{e in employees, d in weekdays} binary;
(V3.1)

Since some employees might prefer working only remotely and might be allowed to do so then we also have the binary variable defined in the basic model.

$$\text{var employee_remote}\{e \text{ in employees: remote_preference}[e] = 2\} \text{ binary;} \quad (\text{V3.2})$$

Note that this variable is defined only for employees who prefer working remotely.

The two sets of variables defined above are clearly related. Thus, they are connected by constraints in Subsection 3.4.

3.3. Objective Function

The objective function of the basic model is updated to include the total daily salary savings for those employees who work some days of week remotely. The company also gets extra savings from those employees who prefer working only remotely and are granted to do so.

$$\begin{aligned} & \text{maximize Total_Savings:} \\ & \text{sum}\{d \text{ in weekdays, } e \text{ in employees: remote_preference}[e] = 1 \text{ or remote_} \\ & \text{preference}[e] = 2\} \\ & \text{salary_saving_daily}[e]*\text{employee_remote_weekday}[e,d] \\ & + \text{sum}\{e \text{ in employees: remote_preference}[e] = 2\} \text{salary_saving_extra}[e]* \\ & \text{employee_remote}[e]; \end{aligned} \quad (\text{O3.1})$$

3.4. Constraints

The constraint on company needs is updated to ensure that company in-office needs are satisfied on *any given weekday*. The constraint provides that the number of employees who work in office in day d and can fulfill need c is at least $\text{need_in_office}[c,d]$.

$$\begin{aligned} & \text{subject to company_professional_needs} \\ & \{c \text{ in company_needs, } d \text{ in weekdays}\}: \text{sum}\{e \text{ in employees}\} \\ & \text{employee_need}[e,c]*(1-\text{employee_remote_weekday}[e,d]) \geq \text{need_in_off-} \\ & \text{ice}[c, d]; \end{aligned} \quad (\text{C3.1})$$

This constraint works the same way as constraint (C2.1) of the basic model.

Two new sets of constraints are added to ensure that for employees with hybrid or remote preference the minimum and maximum days that they are allowed to work remotely are correspondingly $\text{min_days_remotely}[e]$ and $\text{max_days_remotely}[e]$.

$$\begin{aligned} & \text{subject to Min_days_remotely} \\ & \{e \text{ in employees: remote_preference}[e] = 1 \text{ or remote_preference}[e] = 2\}: \\ & \text{sum}\{d \text{ in weekdays}\} \text{employee_remote_weekday}[e,d] \geq \text{min_days_rem-} \\ & \text{otely}[e]; \end{aligned} \quad (\text{C3.2})$$

$$\begin{aligned} & \text{subject to Max_days_remotely} \\ & \{e \text{ in employees: remote_preference}[e] = 1 \text{ or remote_preference}[e] = 2\}: \\ & \text{sum}\{d \text{ in weekdays}\} \text{employee_remote_weekday}[e,d] \leq \text{max_days_rem-} \\ & \text{otely}[e]; \end{aligned} \quad (\text{C3.3})$$

Note that in both constraints (C3.2) and (C3.3), the left-hand side of the inequality represents the number of employees who are allowed to work remotely on weekday d .

A new constraint is needed to connect the two sets of variables defined in Subsection 3.2. It provides that if an employee who prefers working exclusively remotely and is allowed to do so (that is, $\text{employee_remote}[e] = 1$) then the employee will be working remotely on any weekday d (that is, $\text{employee_remote_weekday}[e,d] = 1$).

**subject to Connecting_variables{ d in weekdays, e in employees: remote_preference[e] = 2}: $\text{employee_remote}[e] \leq \text{employee_remote_weekday}[e,d]$;
(C3.4)**

4. Model with Hybrid Mode Option and Flexible Hours (Model 3)

In this section, we further develop the hybrid work model of the previous section. As in Section 3, an employee still might work several days a week remotely and the other days in the office. But we have additional arrangements and requirements for hybrid work as discussed next. Even if an employee is scheduled to work in the office on a certain day it does not have to be all day. An employee can work in the office in shorter time windows. That flexibility might also help to avoid longer commute times in rush hours.

Based on the above arguments, we could define time periods in every workday. 4-hour periods are the most reasonable and we will use it in our dataset. But it could be shorter time periods too, depending on the specific needs of the company.

In our dataset, employee preferences are modified to be 4-hour time periods they would like to be in the office. An employee can have preferences for specific time periods in each day: morning (8-12), late morning-early afternoon (10-2), afternoon (12-4). We assume that there should be no more than one 4-hour time period per day for employees with hybrid or remote preferences.

Company needs requirements also should be modified. A given professional need of a company might stay the same throughout a workday. Alternatively, that number might be different at different times of the day. The former option is easier to model. The latter gives more flexibility and is more realistic, and thus we adopt that approach.

4.1. Sets and Parameters

We need new structures for the new model. Also, some of the parameters defined in the previous model are updated. Input data for this model are again divided into four categories: Time periods data; Employee preference related data; Company professional needs data, and Salary savings related data.

1) *Time periods data*

We still have the same set of weekdays, with 1, 2, 3, 4, 5 corresponding to

Monday, Tuesday, Wednesday, Thursday, Friday.

set weekdays:= 1 2 3 4 5; (S4.1)

But to allow more time flexibility, the new model assumes that employees can work in office in 4-hour time periods. More specifically, we define three such time periods: period 1 is for morning, 8 am - 12 pm; period 2 is for late morning-early afternoon, 10 am - 2 pm; and period 3 for is afternoon, 12 pm - 4 pm.

set employee_time_periods := 1 2 3; (S4.2)

The company needs at different time periods also can be different. In our model, we define four such periods.

set needs_time_periods := 1 2 3 4; (S4.3)

Here time periods 1, 2, 3, 4 correspond to 8 am - 10 am (early morning), 10 am - 12 pm (late morning), 12 pm - 2 pm (early afternoon), 2 pm - 4 pm (late afternoon).

The next set gives a correspondence between the above two sets which is necessary for writing our constraints. It specifies those employee time periods that cover a given needs time period. For example, needs period 2 (corresponding to 10 am - 12 pm) is covered by two employee time periods: 1 (8 am - 12 pm) and 2 (10 am - 2 pm).

set covering_employee_time_periods{t in needs_time_periods} within employee_time_periods; (S4.4)

All those subsets are specified in the data as follows.

set covering_employee_time_periods[1] := 1;

set covering_employee_time_periods[2] := 1 2;

set covering_employee_time_periods[3] := 2 3;

set covering_employee_time_periods[4] := 3;

Of course, companies can define the employee time periods, needs time periods, and their correspondence differently, based on their working hours, company needs and employee preferences.

2) *Employee preference related data*

Employee preference data are mostly the same as in the previous model.

set employees; (S4.5)

param remote_preference {employees}; (P4.1)

This parameter takes three possible values: 2 if the employee prefers to work remotely, 1 if the employee prefers to work in hybrid mode, 0 otherwise.

- Minimum number of days an employee with hybrid or remote preference would like to work remotely:

param min_days_remotely

{e in employees: remote_preference[e] = 1 or remote_preference[e] = 2}; (P4.2)

- Maximum number of days an employee with hybrid or remote preference would like to work remotely:

param max_days_remotely

{e in employees: remote_preference[e] = 1 or remote_preference[e] = 2};

(P4.3)

This parameter should be set to 5 if an employee prefers working exclusively remotely.

A new parameter in this model is about preferred time periods to work in office. Employees specify the time periods in each weekday they prefer to be in office if they have to work in office that workday. Note that an employee who prefers to be in office every day ($\text{remote_preference}[e] = 0$) still might not be in office all the time during a day and thus has preferred time periods to be in office.

param time_period_preference{e in employees, d in weekdays, t in employee_time_periods} binary; (P4.4)

This parameter is equal 1 if employee e prefers period t in weekday d for working in office.

3) *Company professional needs data*

set company_needs; (S4.6)

This set is the same as in the previous model.

As we discussed above, company professional needs are different at different time periods. Thus, parameter need_in_office of the previous model is modified as follows.

param need_in_office{c in company_needs, d in weekdays, t in needs_time_periods}; (P4.5)

It specifies minimum number of employees who can fulfill need c that should work in office in period t of weekday d .

But binary parameter employee_need is the same as in the previous model.

param employee_need{employees, company_needs}; (P4.6)

It equals 1 if the employee can fulfill the need.

4) *Salary savings related data*

Parameters in this part are the same as in the previous model.

param salary_saving_daily{e in employees: remote_preference[e] = 1 or remote_preference[e] = 2}; (P4.7)

It is the daily salary saving amount for the employees who prefer working in hybrid mode.

param salary_saving_remote{e in employees: remote_preference[e] = 2}; (P4.8)

It is the total salary saving amount for the employees who prefer working exclusively remotely.

4.2. Variables

We need a new set of variables for specific periods of time. For each employee e and each weekday d , we define the following binary variable which takes value 1 if employee e works in office in period t of weekday d .

var employee_office_period {e in employees, d in weekdays, t in employee_time_periods} binary; (V4.1)

We also keep the two sets of variables defined in the previous section.

var employee_remote_weekday{e in employees, d in weekdays} binary;
(V4.2)

This variable is equal 1 if employee e is scheduled to work remotely in weekday d.

var employee_remote{e in employees: remote_preference[e] = 2} binary;
(V4.3)

This variable is equal 1 if employee e (remote preference) works exclusively remotely.

The three sets of variables are clearly interconnected. For example, the values of variables employee_office_period uniquely determine the values of variables employee_remote_weekday. But we need all those variables for properly expressing the objective function and the constraints. The connection of different sets of variables is given by constraints in Subsection 4.4.

4.3. Objective Function

The objective function is the same as in the previous model.

maximize Total_Savings:

$$\sum\{d \text{ in weekdays, } e \text{ in employees: remote_preference}[e] = 1 \text{ or remote_preference}[e] = 2\} \text{ salary_saving_daily}[e]*\text{employee_remote_weekday}[e,d]$$

$$+ \sum\{e \text{ in employees: remote_preference}[e] = 2\} \text{ salary_saving_extra}[e]*\text{employee_remote}[e];$$
(O4.1)

4.4. Constraints

The constraints are divided into three groups as discussed below.

4.4.1. Company Professional Need Constraints

The constraint on company professional needs is modified to ensure that in-office needs are satisfied at any period of any weekday.

subject to Company_professional_needs
{c in company_needs, d in weekdays, t1 in needs_time_periods}:

$$\sum\{e \text{ in employees, } t2 \text{ in covering_employee_time_periods}[t1]\} \text{employee_need}[e,c]*\text{employee_office_period}[e,d,t2] \geq \text{need_in_office}[c,d,t1];$$
(C4.1)

The constraint works the following way. Note that the product employee_need[e,c]*employee_office_period[e,d,t2] can take only values 0 or 1. It equals 1 if and only if employee_need[e,c] = 1 (that is, employee e can fulfill need c) and employee_office_period[e,d,t2] = 1 (that is, employee e works in office in period t2 of day d). But since t2 belongs to covering_employee_time_periods[t1], then employee_office_period[e,d,t2] = 1 also implies that employee e works in office in period t1 of day d. Thus, the summation in the left-hand side of (C4.1) equals the number of employees who work in office in period t1 of day d and can fulfill need c; while the right-hand side requires that the number of those kind of employees is at least need_in_office[c,d,t1].

4.4.2. Employee Remote and In-Office Work Preference Constraints

The constraints with minimum and maximum days of remote work stay the same.

$$\begin{aligned} &\text{subject to Min_days_remotely} \\ &\{e \text{ in employees: remote_preference}[e] = 1 \text{ or } \text{remote_preference}[e] = 2\}: \\ &\text{sum}\{d \text{ in weekdays}\} \text{ employee_remote_weekday}[e,d] \geq \text{min_days_remotely}[e]; \end{aligned} \quad (\text{C4.2})$$

$$\begin{aligned} &\text{subject to Max_days_remotely} \\ &\{e \text{ in employees: remote_preference}[e] = 1 \text{ or } \text{remote_preference}[e] = 2\}: \\ &\text{sum}\{d \text{ in weekdays}\} \text{ employee_remote_weekday}[e,d] \leq \text{max_days_remotely}[e]; \end{aligned} \quad (\text{C4.3})$$

A new constraint provides that each employee with hybrid or remote preference is required to work in office at most one 4-hour period in a weekday.

$$\begin{aligned} &\text{subject to At_most_one_work_period_per_day} \\ &\{e \text{ in employees, } d \text{ in weekdays: remote_preference}[e] = 1 \text{ or } \text{remote_preference}[e] = 2\}: \\ &\text{sum}\{t \text{ in employee_time_periods}\} \text{ employee_office_period}[e,d,t] \leq 1; \end{aligned} \quad (\text{C4.4})$$

Similarly, a new constraint provides that each employee with no remote preference is scheduled to work in office at least one 4-hour period in a weekday. (But the employee might work in office more than 4 hours if they prefer to do so.)

$$\begin{aligned} &\text{subject to At_least_one_work_period_per_day} \\ &\{e \text{ in employees, } d \text{ in weekdays: remote_preference}[e] = 0\}: \\ &\text{sum}\{t \text{ in employee_time_periods}\} \text{ employee_office_period}[e,d,t] \geq 1; \end{aligned} \quad (\text{C4.5})$$

The next constraint provides that an employee with no remote preference cannot be scheduled in two periods that have a time overlap; in our model, those are periods 1, 2 and periods 2, 3.

$$\begin{aligned} &\text{subject to No_overlap_of_time_periods} \\ &\{e \text{ in employees, } d \text{ in weekdays, } t1 \text{ in employee_time_periods, } t2 \text{ in employee_time_periods: remote_preference}[e] = 0 \text{ and } t2 = t1+1\}: \\ &\text{employee_office_period}[e,d,t1] + \text{employee_office_period}[e,d,t2] \leq 1; \end{aligned} \quad (\text{C4.6})$$

Our model also requires that an employee is not scheduled to work in office in those time periods when they prefer working remotely.

$$\begin{aligned} &\text{subject to Time_period_preference} \\ &\{e \text{ in employees, } d \text{ in weekdays, } t \text{ in employee_time_periods}\}: \\ &\text{if time_period_preference}[e,d,t] = 0 \text{ then } \text{employee_office_period}[e,d,t] = 0; \end{aligned} \quad (\text{C4.7})$$

4.4.3. Constraints Connecting Variables

We need two sets of constraints connecting the three sets of variables.

A new constraint provides that if an employee (hybrid or remote preference) works in office in any time period t of weekday d (that is, $\text{employee_office_period}[e,d,t] = 1$) then that employee cannot work remotely the whole day (that is,

$employee_remote_weekday[e,d] = 0$).

subject to **Connecting_variables1**{ e in employees, d in weekdays, t in employee_time_periods: $remote_preference[e] = 1$ or $remote_preference[e] = 2$ }:
 $employee_remote_weekday[e,d] \leq 1 - employee_office_period[e,d,t]$;
 (C4.7)

The constraint connecting variables $employee_remote[e]$ and $employee_remote_weekday[e,d]$ stays the same.

subject to **Connecting_variables2**{ d in weekdays, e in employees: $remote_preference[e] = 2$ }:
 $employee_remote[e] \leq employee_remote_weekday[e,d]$;
 (C4.8)

The full AMPL model for Model 3 is given in **Appendix A.1**.

5. Computational Results

We have tested all three models. Below we present some computational results for the Model with Hybrid Mode and Flexible Hours, the most comprehensive of our models since it mostly includes the elements of other models too.

5.1. Input Data and Model Outputs

We have tested the model for different data sets. The problem size is determined by the number of employees which is 20, 40, and 80 in our data sets.

In our data, employee preferences for remote and hybrid work are consistent with the findings of the Gallup survey in [1]. While getting data from individual companies also can be helpful, survey results like Gallup are the best available source for generating data for our models. For 20 employees, the data are created the following way: 5 employees (25%) prefer exclusively remote work; 10 (50%) prefer hybrid work and 5 (25%) prefer work in the office. Those who prefer hybrid work specify how many days they want to work in the office. In our data set, 2 employees prefer 1 - 2 days, 2 employees prefer 2 - 3 days, 2 employees prefer 3 - 4 days, 2 employees prefer 1 - 3 days, and 2 employees prefer 2 - 4 days.

The data sets for 40 and 80 employees are created using the same percentages discussed above for 20 employees.

A full data set for 20 employees is given in **Appendix A.2**. We ran the models on the NEOS server using the optimization software package CPLEX [9]. The CPLEX output for the data set with 20 employees is given in **Appendix A.3**.

Some characteristics and outputs of the model for different problem sizes (20, 40, 80 employees) are summarized in **Table 1**.

Table 1. Summary of computational results.

Number of employees	Number of variables	Number of constraints	Running time (seconds)	Optimal value	Number of employees allowed/want to work exclusively remotely
20	205	790	0.018	129	3/5
40	810	1520	0.026	304	9/10
80	1620	2980	0.079	584	18/20

5.2. Sensitivity Analysis

Sensitivity analysis is done by changing one of the parameters to see how the optimal solution changes in the result. The most reasonable analysis for our model is when changing the parameter *company_need* because it could depend on the company's policies. Reducing some values of parameter *company_need* is equivalent to relaxing the requirement of doing certain activities in the office. It will allow more people to work remotely or in hybrid mode and thus might increase associated salary savings.

We have done a sensitivity analysis for the data set with 20 employees. Parameter *company_need* has 3 different office needs in our data set.

For our default data set (called **Data1**, given in **Appendix A.2**) the model has an optimal value 129; 3 out of 5 employees who prefer working exclusively online are allowed to do so.

In **Data2**, all data for office need #3 are made 0. It is equivalent to relaxing the requirement that some activities related to need #3 must be done in the office. All other data are the same as in data1.txt. As a result, the optimal value goes up to 134. But still the same 3 out of 5 employees who prefer working only online are allowed to do so. Thus, the savings are the result of having some employees work more days remotely.

Another option is reducing all three office needs uniformly to see how it affects the solution. In **Data3**, all office needs are decreased by 1 (unless it was already 0). As a result, everyone who prefers to work exclusively remotely is allowed to do so, and the new optimal value is 173 which implies more significant salary savings compared to Data2.

Our results imply that reducing the office needs uniformly (even by a small number) might lead to more savings than replacing one of the office needs with remote work altogether. It is also a more balanced approach of switching to hybrid work. But in real life, the best option might be company specific. A company might try our model for different combinations of office need reductions to see which option brings the best possible savings to the company while also keeping a good balance between in-office and remote work.

5.3. Computational Complexity and Running Time

For computational complexity of the model, we estimate the number of binary variables. Let N be the number of employees. Assuming 5 weekdays, 3 employee time periods, and a quarter of employees with exclusively remote work preference, we have the following number of variables.

The number of *employee_remote* variables is $0.25 * N$.

The number of *employee_remote_weekday* variables is $5 * N$.

The number of *employee_office_period* variables is $5 * 3 * N = 15 * N$.

Thus, the total number of variables is $0.25 * N + 5 * N + 15 * N = 20.25 * N$.

For example, when $N = 40$ the number of variables is 810, and for $N = 80$ the number of variables is 1620.

For all our data sets, including the one with 80 employees, the output was received in less than a second (see **Table 1**).

5.4. Relaxing the Integrality of Variables

We defined several sets of related binary variables in Model 3. When we ran the model for the data set with 20 employees by relaxing the integrality of all variables, the LP relaxation returned a fractional solution. But when the integrality of only variables `employee_office_period` were relaxed to $\geq 0, \leq 1$, the LP relaxation returned an integer solution. Our conjecture is that it is generally true, relaxing the integrality of variables `employee_office_period` will always return an integer solution. If the conjecture is proved to be true, then it might significantly improve the computational efficiency for data sets with large numbers of employees since `employee_office_period` are the largest number of variables in the model. At this point, it is an open question and is listed as one of the future directions below.

6. Conclusions and Future Directions

6.1. Conclusions

We develop mathematical models that might help companies to organize remote and hybrid work. Some conclusions are given below.

- *Choosing the right model.* Model 3 is the most comprehensive in the sense that it includes all the hybrid work features we introduced in this paper. But a company might decide which model works best in their case. If a company allows its employees to work some days from home but no flexible hours on those days when they work in the office, then Model 2 would be best suited for creating their work schedule. But if flexible work hours during a workday are allowed then Model 3 should be used.
- *Increasing savings through remote and hybrid work.* Companies might increase their savings by allowing some employees to work in hybrid mode or exclusively remotely. More savings can be achieved by relaxing the requirement on numbers of employees who should work in the office to satisfy certain company needs. Based on our sensitivity analysis, the savings are higher when those number requirements are reduced uniformly for all office needs. But companies can use the models to perform their own sensitivity analysis for their company-specific data to determine the best way to reduce salary costs.
- *Efficiency of models.* Our models are computationally very efficient. For the data set with 80 employees, the output was received in less than a second.
- *Use of models by organizations.* Linear/Integer Programming software is very accessible (for example, through AMPL and NEOS websites [8] [9]) and relatively easy to use for people with Applied Math and/or CS background. Organizations can conduct surveys of their employees about hybrid work preferences to create their own data sets for the models.

6.2. Future Directions

The models of this paper can be further developed in different ways. Two possible directions are described below.

- *Modeling Teamwork.* Some employees might work in teams. Some of those teams might be highly interdependent and might have meetings (or other types of work) that should be in person [1]. Thus, one way to develop our models is to assign certain time periods, not necessarily the whole workday, when all team members are scheduled to be in the office.
- *Model with different options for office space.* Renting and maintaining office space might be expensive for companies. If a significant number of employees are given the option of working remotely or in hybrid mode then a company might be able to save money by reducing its office space. Thus, another extension of our models could be companies choosing among several possible locations for renting or building their offices. The objective of the new model would be minimizing the total cost (by having less expensive office space and more salary savings by having employees working remotely) but at the same time satisfying the office needs requirements.
- *Models with a large number of employees.* Our models make decisions about remote/hybrid work for each individual employee separately. For companies with large numbers of employees, the number of decision variables might be big which might negatively affect the efficiency of running the models. In that case, an alternative approach would be having decisions about categories of employees who have similar professional duties. Then for each category of employees, the decision could be about the percentage of employees who are allowed to work remotely.
- *Relaxing the integrality of variables.* Our conjecture is that after relaxing the integrality of the largest number of variables in Model 3 (called `employee_office_period`) the model will still return an integer solution. It is supported by our computational results. If the conjecture is proved to be true, then it will significantly improve the computational efficiency of the model.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

- [1] Wigert, B. (2022) The Future of Hybrid Work: 5 Key Questions Answered with Data. Gallup: Workplace. <https://www.gallup.com/workplace/390632/future-hybrid-work-key-questions-answered-data.aspx>
- [2] Castillo-Salazar, J., Landa-Silva, D. and Qu, R. (2016) Workforce Scheduling and Routing Problems: Literature Survey and Computational Study. *Annals of Operations Research*, **239**, 39-67. <https://doi.org/10.1007/s10479-014-1687-2>
- [3] Ernst, A.T., Jiang, H., Krishnamoorthy, M. and Sier, D. (2004) Staff Scheduling and

-
- Rostering: A Review of Applications, Methods and Models. *European Journal of Operations Research*, **153**, 3-27. [https://doi.org/10.1016/S0377-2217\(03\)00095-X](https://doi.org/10.1016/S0377-2217(03)00095-X)
- [4] Van den Bergh, J., Beliën, J., Bruecker, P.D., Demeulemeester, E. and Boeck, L.D. (2013) Personnel Scheduling: A Literature Review. *European Journal of Operations Research*, **226**, 367-385. <https://doi.org/10.1016/j.ejor.2012.11.029>
- [5] Guerriero, F. and Guido, R. (2022) Modeling a Flexible Staff Scheduling Problem in the Era of Covid-19. *Optimization Letters*, **16**, 1259-1279. <https://doi.org/10.1007/s11590-021-01776-3>
- [6] Zucchi, G., Iori, M. and Subramanian, A. (2021) Personnel Scheduling during Covid-19 Pandemic. *Optimization Letters*, **15**, 1385-1396. <https://doi.org/10.1007/s11590-020-01648-2>
- [7] Mohan, S. (2008) Scheduling Part-Time Personnel with Availability Restrictions and Preferences to Maximize Employee Satisfaction. *Mathematical and Computer Modelling*, **48**, 1806-1813. <https://doi.org/10.1016/j.mcm.2007.12.027>
- [8] The AMPL Website. <http://www.ampl.com/>
- [9] The CPLEX Website. <https://neos-server.org/neos/solvers/lp:CPLEX/AMPL.html>

Appendix A. Complete AMPL Model, Data Set and Its Output for Model 3

A.1. Full AMPL Model

```

set employees := 1..20;
set weekdays;
set employee_time_periods;
set needs_time_periods;
set covering_employee_time_periods{t in needs_time_periods} within em-
ployee_time_periods;
# those employee time periods that cover a needs time period t
### WORK MODE RELATED ###
param remote_preference {employees};
# takes three possible values:
# 2 if prefers to work remotely
# 1 if prefers to work in hybrid mode
# 0 otherwise
param min_days_remotely{e in employees: remote_preference[e] = 1 or re-
mote_preference[e] = 2};
param max_days_remotely{e in employees: remote_preference[e] = 1 or re-
mote_preference[e] = 2};
param time_period_preference{e in employees, d in weekdays, t in em-
ployee_time_periods} binary;
# is equal 1 if employee e prefers that time period t in weekday d for working
in office
### COMPANY PROFESSIONAL NEEDS DATA ###
set company_needs;
# company has professional needs that should be done in office
param need_in_office{company_needs, d in weekdays, t in needs_time_per-
iods};
# minimum number of employees with the need who should work in office in
a given time period of a given weekday
param employee_need{e in employees, c in company_needs};
# is 1 if the employee e can fulfill need c
### SALARY SAVINGS RELATED ###
param salary_saving_daily{e in employees: remote_preference[e] = 1 or re-
mote_preference[e] = 2};
# daily salary saving amount for the employees who prefer working in hybrid
mode
param salary_saving_extra{e in employees: remote_preference[e] = 2};
# extra salary saving amount for the employees who prefer working remotely
and are granted to do so
##### VARIABLES #####
var employee_office_period{e in employees, d in weekdays, t in employee_

```

```

time_periods} binary;
# is equal 1 if employee e works in office in time period t of day d
var employee_remote_weekday{e in employees, d in weekdays} binary;
# is equal 1 if employee e works remotely in day d
var employee_remote{e in employees: remote_preference[e] = 2} binary;
# is equal 1 if the employee e works remotely all the time
##### OBJECTIVE FUNCTION #####
maximize Total_Savings:
sum{d in weekdays, e in employees: remote_preference[e] = 1 or remote_
preference[e] = 2} salary_saving_daily[e]*employee_remote_weekday[e,d]
+ sum{e in employees: remote_preference[e] = 2} salary_saving_extra[e]*
employee_remote[e];
##### CONSTRAINTS #####
### Company need constraints ###
subject to Company_professional_needs{c in company_needs, d in weekdays,
t1 in needs_time_periods}:
sum{e in employees, t2 in covering_employee_time_periods[t1]}employee_
need[e,c]*employee_office_period[e,d,t2] >= need_in_office[c,d,t1];
# number of employees work in office in period t1 of day d and can fulfill
need c is at least need_in_office[c,d,t]
### Employee remote work preference constraints ###
subject to Min_days_remotely{e in employees: remote_preference[e] = 1 or
remote_preference[e] = 2}:
sum{d in weekdays} employee_remote_weekday[e,d] >= min_days_remotely[e];
# number of days an employee (hybrid or remote preference) works remotely
is at least min_days_remotely[e]
subject to Max_days_remotely{e in employees: remote_preference[e] = 1 or
remote_preference[e] = 2}:
sum{d in weekdays} employee_remote_weekday[e,d] <= max_days_remotely[e];
# number of days an employee (hybrid or remote preference) works remotely
is at most max_days_remotely[e]
subject to At_most_one_work_period_per_day{e in employees, d in week-
days: remote_preference[e] = 1 or remote_preference[e] = 2}:
sum{t in employee_time_periods} employee_office_period[e,d,t] <= 1;
# each employee (remote or hybrid preference) can work at most one time pe-
riod per day
subject to At_least_one_work_period_per_day{e in employees, d in weekdays:
remote_preference[e] = 0}:
sum{t in employee_time_periods} employee_office_period[e,d,t] >= 1;
# each employee who prefers office work should work in office at least one
time period per day
subject to No_overlap_of_time_periods
{e in employees, d in weekdays, t1 in employee_time_periods, t2 in em-
ployee_time_periods:

```

```

remote_preference[e] = 0 and t2 = t1+1};
employee_office_period[e,d,t1] + employee_office_period[e,d,t2] <= 1;
# no overlap: cannot be scheduled for periods 1,2 or periods 2,3
subject to Time_period_preference{e in employees, d in weekdays, t in employee_time_periods}:
if time_period_preference[e,d,t] = 0 then employee_office_period[e,d,t] = 0;
# an employee is not scheduled to work in office at the time period he doesn't
prefer
### Constraints connecting variables ###
subject to Connecting_variables1{e in employees, d in weekdays, t in employee_time_periods: remote_preference[e] = 1 or remote_preference[e] = 2}:
employee_remote_weekday[e,d] <= 1 - employee_office_period[e,d,t];
# if an employee (remote or hybrid preference) works in office in period t of
day d then the employee can't work remotely that day
subject to Connecting_variables2{d in weekdays, e in employees: remote_preference[e] = 2}:
employee_remote[e] <= employee_remote_weekday[e,d];
# if an employee (remote preference) works remotely all the time then he
works remotely any given day

```

A.2. Default Data Set for 20 Employees

```

data;
set weekdays:= 1 2 3 4 5;
# M Tu W Th F
set employee_time_periods := 1 2 3;
# time periods that an employee might work in office are M (morning 8-12),
MA (late morning-early afternoon 10-2), A (afternoon 12-4)
set needs_time_periods := 1 2 3 4;
# company needs are different at different time periods 8-10 10-12 12-2 2-4
set covering_employee_time_periods[1] := 1;
set covering_employee_time_periods[2] := 1 2;
set covering_employee_time_periods[3] := 2 3;
set covering_employee_time_periods[4] := 3;
param remote_preference :=
1    0
2    0
3    0
4    0
5    0
6    1
7    1
8    1
9    1

```

```

10  1
11  1
12  1
13  1
14  1
15  1
16  2
17  2
18  2
19  2
20  2
;
param:
    min_days_remotely  max_days_remotely :=
6      1              2
7      1              2
8      2              3
9      2              3
10     3              4
11     3              4
12     1              3
13     1              3
14     2              4
15     2              4
16     3              5
17     3              5
18     4              5
19     4              5
20     4              5
;
param time_period_preference :=
# employee 1 preferences
[1, *, *]:  1      2      3  :=
1      1      1      0
2      0      1      1
3      1      1      1
4      0      1      1
5      1      1      0
# employee 2 preferences
[2, *, *]:  1      2      3  :=
1      1      1      1
2      1      1      0
3      0      1      1

```

4	0	1	0	
5	0	0	1	
# employee 3 preferences				
[3, *, *] :	1	2	3	:=
1	0	1	1	
2	0	1	0	
3	1	1	1	
4	1	1	0	
5	0	1	1	
# employee 4 preferences				
[4, *, *] :	1	2	3	:=
1	1	1	1	
2	1	1	0	
3	0	1	1	
4	1	0	0	
5	1	1	1	
# employee 5 preferences				
[5, *, *] :	1	2	3	:=
1	1	1	0	
2	1	1	0	
3	1	1	0	
4	1	1	0	
5	1	1	0	
# employee 6 preferences				
[6, *, *] :	1	2	3	:=
1	0	1	1	
2	0	1	1	
3	0	1	1	
4	0	1	1	
5	0	1	1	
# employee 7 preferences				
[7, *, *] :	1	2	3	:=
1	0	1	1	
2	0	1	1	
3	1	1	1	
4	0	1	1	
5	1	1	0	
# employee 8 preferences				
[8, *, *] :	1	2	3	:=
1	1	1	1	
2	1	1	0	
3	0	1	1	
4	0	1	0	
5	0	0	1	

employee 9 preferences

[9, *, *]:	1	2	3	:=
1	0	1	1	
2	0	1	0	
3	1	1	1	
4	1	1	0	
5	0	1	1	

employee 10 preferences

[10, *, *]:	1	2	3	:=
1	0	1	1	
2	0	1	1	
3	1	1	1	
4	0	1	1	
5	1	1	0	

employee 11 preferences

[11, *, *]:	1	2	3	:=
1	1	1	1	
2	1	1	0	
3	0	1	1	
4	0	1	0	
5	0	0	1	

employee 12 preferences

[12, *, *]:	1	2	3	:=
1	0	1	1	
2	0	1	0	
3	1	1	1	
4	1	1	0	
5	0	1	1	

employee 13 preferences

[13, *, *]:	1	2	3	:=
1	0	1	1	
2	0	1	1	
3	1	1	1	
4	0	1	1	
5	1	1	0	

employee 14 preferences

[14, *, *]:	1	2	3	:=
1	1	1	1	
2	1	1	0	
3	0	1	1	
4	0	1	0	
5	0	0	1	

employee 15 preferences

```

[15, *, *]: 1      2      3      :=
1          0      1      1
2          0      1      0
3          1      1      1
4          1      1      0
5          0      1      1
# employee 16 preferences
[16, *, *]: 1      2      3      :=
1          0      1      1
2          0      1      1
3          1      1      1
4          0      1      1
5          1      1      0
# employee 17 preferences
[17, *, *]: 1      2      3      :=
1          1      1      1
2          1      1      0
3          0      1      1
4          0      1      0
5          0      0      1
# employee 18 preferences
[18, *, *]: 1      2      3      :=
1          0      1      1
2          0      1      0
3          1      1      1
4          1      1      0
5          0      1      1
# employee 19 preferences
[19, *, *]: 1      2      3      :=
1          0      1      1
2          1      1      0
3          1      1      0
4          1      1      0
5          1      1      0
# employee 20 preferences
[20, *, *]: 1      2      3      :=
1          0      1      1
2          0      1      1
3          0      1      1
4          0      1      1
5          0      1      1 ;
### COMPANY NEEDS DATA ###
set company_needs := 1 2 3;

```

```

param need_in_office :=
# need 1 data
[1, *, *]: 1 2 3 4 :=
1 2 6 5 3
2 4 5 3 0
3 0 3 6 3
4 3 4 2 3
5 2 3 0 3
# need 2 data
[2, *, *]: 1 2 3 4 :=
1 5 3 4 0
2 3 2 5 3
3 3 3 2 4
4 0 4 5 2
5 1 6 3 0
# need 3 data
[3, *, *]: 1 2 3 4 :=
1 4 3 4 2
2 0 4 3 3
3 3 5 4 2
4 2 4 6 3
5 2 4 3 0;
param employee_need:
1 2 3 :=
1 1 0 1
2 1 1 1
3 0 1 1
4 1 1 0
5 0 0 1
6 1 1 0
7 1 0 1
8 1 1 1
9 0 1 1
10 1 1 0
11 0 0 1
12 1 1 0
13 1 0 1
14 1 1 1
15 0 1 1
16 1 1 0
17 0 0 1
18 1 1 0
19 1 0 1
20 1 0 0

```

```

;
### SALARY SAVINGS RELATED ###
param salary_saving_daily :=
6      3
7      2
8      1
9      3
10     2
11     3
12     2
13     1
14     3
15     2
16     3
17     2
18     1
19     3
20     2
;
param salary_saving_extra :=
16     8
17    10
18     8
19    12
20    10
;

```

A.3. CPLEX Output for 20 Employees

```

CPLEX 20.1.0.0: optimal integer solution; objective 129
34 MIP simplex iterations
0 branch-and-bound nodes
employee_remote [*] :=
16     0
17     1
18     0
19     1
20     1
;
employee_remote_weekday [*, *]
:      1      2      3      4      5      :=
6      1      0      0      0      0
7      1      0      0      0      1
8      0      0      1      1      0

```

9	1	1	0	0	0
10	0	0	1	1	1
11	0	1	1	1	1
12	0	0	1	0	0
13	0	0	0	0	1
14	0	0	1	1	1
15	1	1	0	0	0
16	1	0	1	0	1
17	1	1	1	1	1
18	1	1	1	0	1
19	1	1	1	1	1
20	1	1	1	1	1

;

employee_office_period [* , *, 1]

:	1	2	3	4	5	:=
1	0	0	1	0	1	
2	1	1	0	0	0	
3	0	0	1	0	0	
4	1	1	0	1	1	
5	1	0	1	1	1	
8	1	1	0	0	0	
9	0	0	1	0	0	
10	1	0	0	0	0	
11	1	0	0	0	0	
12	0	0	0	1	0	
14	1	1	0	0	0	
15	0	0	1	1	0	
18	0	0	0	1	0	

[* , *, 2]

:	1	2	3	4	5	:=
1	1	0	0	0	0	
2	0	0	0	1	0	
3	0	1	0	1	1	
5	0	1	0	0	0	
6	0	0	0	0	1	
7	0	0	1	0	0	
9	0	0	0	1	1	
12	0	1	0	0	1	
13	1	0	1	0	0	
15	0	0	0	0	1	

[* , *, 3]

:	1	2	3	4	5	:=
1	0	1	1	1	0	

2	1	0	1	0	1
3	1	0	1	0	0
4	1	0	1	0	1
6	0	1	1	1	0
7	0	1	0	1	0
8	0	0	0	0	1
10	0	1	0	0	0
12	1	0	0	0	0
13	0	1	0	1	0
16	0	1	0	1	0
;					