程式設計(111-1) 期中專案

題目設計:孔令傑 國立臺灣大學資訊管理學系

1 題目敘述

你剛被派任到一家人壽保險公司的客服部(Customer Service Department)擔任部長,帶領著部門內所有的客戶服務專員(Customer Service Representative,CSR,以下簡稱員工)。最近因為人力的短缺,加上原本的部長使用傳統的手工方式排班表,導致班表效率不彰,員工們經常需要超時工作或連續上好幾天夜班,但尖峰時刻值班的同仁又忙不過來(電話接不完)。為了改善這個狀況,總經理將一切託付在你身上,希望具有資訊與管理雙專長的你,可以設計一個解決方案(你也可以稱之為演算法、軟體、程式),每個月都能為下個月產出能夠盡量滿足人力需求,又可以同時兼顧員工生活品質的班表¹。

在這一個部門裡,總共有 n_I 位員工,每個月月中要安排次月共 n_J 天的班表(例如次月如果是一月那 $n_J=31$,如果是四月那 $n_J=30$,依此類推)。客服部門營運的時間是每天的早上 9:00 到晚上 9:00 (21:00),從早上 9:00 開始,每半個小時定義為一個時段(例如 9:00-9:30、9:30-10:00 等等),直到 21:00 結束,一天共分成 24 個時段。每一天中有 n_K+1 種班型(英文叫「shift」),每一種班型代表著一位員工應該於 24 個時段之中的哪一些時段上班、哪一些時段休息。舉例來說,如圖 1 所示的這一個例子中,我們總共定義了 12 種班型,綠色格子代表上班時間,灰色格子代表休息時間。如果有一位員工當天被指派了第 1 號班型,那麼他就需要於早上 9:00 準時上班,在 12:00 到 13:00 之間休息,再從 13:00 開始工作直到 18:00 下班。此外,公司也進一步定義班型的種類(shift category),分別為「正常」、「含夜班」以及「休假」三種,正常班型代表著不會在 18:00 之後值班,換句話說也就是不含夜班;含夜班的班型就會在 18:00 之後至少一任意時段值班;休假的班型則是整天 24 個時段都不用上班。仍然以圖 1 為例,第 1 到 4 號班型即是正常班型,第 5 到 11 號班型則是含夜班的班型,最後,第 0 號班型固定為休假班型。題目給定的 n_K 代表除了第 0 號班型(也就是休假班型)之外的班型個數,若以圖 1 為例, n_K 就會是 11。

根據長期累積的經驗與數據,公司估計了每一天的每個時段的人力需求(理想上應該要有幾個員工在值班),並且希望排出來的班表能夠盡量滿足人力需求。圖 2 提供了一個例子(的前十天),在第 1 天的 9:00-9:30 我們需要 29 位員工、在第 5 天

¹這個期中專案的任務,是從孔令傑老師過往和真實企業合作的真實問題簡化而來的。

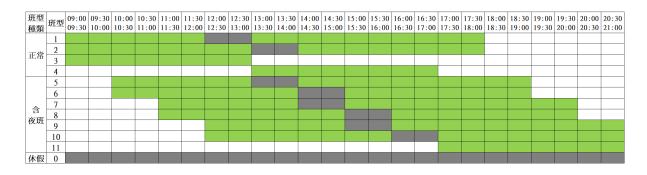


圖 1: 班型範例

的 17:30–18:00 我們需要 11 位員工,依此類推。我們定義一個時段的缺工人數(lack amount) 為

max{表定人力需求 - 實際值班人數,0},

舉例來說,若我們在第 1 天的 9:00-9:30 共有 25 位員工值班,則此時段的缺工人數即 為 29-25=4;若我們指派 30 位員工上班,則缺工人數為 0。排班的基本目標是最小 化下個月共 $24n_J$ 個時段的總缺工人數。

除了盡可能滿足人力需求以外,我們也希望同時達成人性化的排班,因此,每個月月中我們會調查每一位員工的休假需求,讓他們可以提出在特定某幾天休假的願望,等調查完畢後才開始排班。我們令 R 為蒐集到的休假需求個數(想當然爾, $0 \le R \le n_I n_J$),而一個休假需求則紀錄了某個員工希望在某一天休假。我們也想要盡量減少不被滿足的休假需求。此外,所有的員工都不喜歡晚上要工作,所以公司裡也希望每一位員工在每連續七天中只有最多一天被指派「含夜班」的班型。

最後,為了讓員工們有良好的工作生活平衡(work-life balance),有三點限制需要絕對遵守:第一點是源自於勞動基準法的規定,我們需要讓每一位員工在接下來的 n_J 天排班中,至少有L天完整的休假(以圖1為例,若被排為班型3或4或11,都不算是完整休假;換言之,每位員工都需要有L天被排為班型0);第二點是規定每位員工每連續的七天中至少要有一天完整休假(被排為班型0);第三點則是每一位員工在一天內只能被指派恰好一種班型(以圖1為例,不能指派某人在某天又班型3又班型11,畢竟這雖然看起來是可行的,但對員工來說太不方便了)。

要安排出一個好的班表,聽起來要考慮的東西很多,我們試著將以上複雜的狀況, 寫成清楚的目標函數。根據以上的題目敘述,我們希望在滿足限制的前提下最小化總 缺工數、未滿足的休假需求總數,以及七天內的超額夜班總數。因此我們定義目標函 數如下:

- 總缺工人數:為接下來 n_I 天中共 $24n_I$ 個時段的缺工人數總和。
- 未滿足的休假需求總數:當某員工提出希望在某一天休假(亦即希望被指派第0

Time	1	2	3	4	5	6	7	8	9	10
09:00~09:30	29	20	21	29	20	21	17	29	20	21
09:30~10:00	24	20	18	24	20	18	18	24	20	18
10:00~10:30	25	21	20	25	21	20	17	25	21	20
10:30~11:00	24	22	20	24	22	20	18	24	22	20
11:00~11:30	24	19	22	24	19	22	20	24	19	22
11:30~12:00	24	18	19	24	18	19	17	24	18	19
12:00~12:30	17	12	13	17	12	13	12	17	12	13
12:30~13:00	13	14	11	13	14	11	12	13	14	11
13:00~13:30	17	14	14	17	14	14	11	17	14	14
13:30~14:00	23	18	17	23	18	17	15	23	18	17
14:00~14:30	21	20	20	21	20	20	18	21	20	20
14:30~15:00	23	20	21	23	20	21	17	23	20	21
15:00~15:30	21	18	18	21	18	18	16	21	18	18
15:30~16:00	18	20	18	18	20	18	17	18	20	18
16:00~16:30	20	17	17	20	17	17	14	20	17	17
16:30~17:00	20	17	15	20	17	15	15	20	17	15
17:00~17:30	16	11	15	16	11	15	13	16	11	15
17:30~18:00	11	11	13	11	11	13	11	11	11	13
18:00~18:30	10	8	8	10	8	8	9	10	8	8
18:30~19:00	8	8	6	8	8	6	7	8	8	6
19:00~19:30	8	9	6	8	9	6	8	8	9	6
19:30~20:00	8	8	8	8	8	8	7	8	8	8
20:00~20:30	9	8	7	9	8	7	7	9	8	7
20:30~21:00	8	7	6	8	7	6	6	8	7	6

圖 2: 人力需求範例(節錄第一天至第十天)

號班型),但仍被指派去上班,我們就說這個休假需求未被滿足,未滿足的休假需求總數就往上加1。換言之,未滿足的休假需求總數最多R個,最少0個。

 超額夜班總數:如前所述,我們希望每一位員工在連續七天中,最多只有一天被 指派「含夜班」的班型。因此,當一位員工在連續七天中被指派兩天含夜班的班 型,則他在這七天中的超額夜班數為2-1=1天。

需要注意的是,在「每七天要休一天」和「超額夜班」這兩件事上的「連續七天」的檢查是以「rolling window」的方式進行的,也就是這些「連續七天」是會重疊的,亦即第 1 天到第 7 天要檢查一次,第 2 天到第 8 天又要檢查一次。以「每七天要休一天」來說,如果想要安排某位員工在第 2、10、13、16、19、22、25、28 天休假,則他在第 3 到 9 天的這連續七天沒有休任何一天,這樣是不行的。又以「超額夜班」來說,若某一位員工在前十天的排班狀況為「正常、正常、休假、含夜班、含夜班、含夜班、正常、休假、正常、含夜班」,我們會先檢查第一到第七天,其中包含三次夜班,超額兩天,接著檢查第二到第八天,又超額兩天,接著檢查第三到第九天和第四

²換言之,如果有一位員工連續數天都有含夜班,可能會造成相當大的超額夜班總數。

到第十天,依序分別超額兩天和三天。因此,這一位員工前十天的超額夜班總數為 2+2+2+3=9。最後,理論上我們也應該要跨月檢查(例如檢查下個月的前六天和 本月的最後一天是否有員工都沒休假),但為了簡單起見,在這個期中專案中讓我們假 設這不是個問題。

結合以上三項指標,我們可以將目標函數寫成:

min
$$\sum_{j=1}^{n_J} \sum_{t=1}^{24}$$
 缺工數 $+ w_1 \times$ 未滿足的休假需求總數 $+ w_2 \times$ 超額夜班總數 ,

其中 w_1 為題目給定的未滿足的休假需求總數之權重, w_2 為七天內的超額夜班總數之權重。

你將被給定兩個權重、班別、需求人力、休假需求等資訊,請發揮你的專長,試 著設計一個演算法,能夠一方面滿足限制,一方面最小化目標函數,在照顧員工的同 時也提高服務品質吧!

2 輸入輸出格式

系統會提供許多筆測試資料,每筆測試資料裝在一個檔案裡。在每個檔案中,第一列 存放七個整數 $n_I \cdot n_J \cdot n_K \cdot L \cdot w_1 \cdot w_2$ 和 R。接下來的資訊分為三大部分,依序分 別是 n_K 種班型的各時段排班狀況、 n_J 天中各時段的人力需求以及 R 項休假需求。

- 第一部分從第二列開始至第 n_K+2 列結束,每列各存放 24 個整數,其中第 k+1 列依序存著 $s_{k,1}$ 、 $s_{k,2}$ 直到 $s_{k,24}$,表示第 k 號班型需要值班的時段,若為 1 表示要值班,若為 0 則否。第 n_K+2 列會固定存放第 0 號班型(休假)的排班狀態,也就是 $\{0,0,0,0,...,0\}$ 。
- 第二部分從第 $n_K + 3$ 列開始至第 $n_K + n_J + 2$ 列,每列各存放 24 個整數,其中 第 $n_K + 2 + j$ 列依序存著 $d_{j,1} \cdot d_{j,2}$ 直到 $d_{j,24}$,表示第 j 天各時段的人力需求。 請注意在資料被輸入時,其格式是圖 2 的轉置(transpose),亦即在圖 2 中一列 是一個時段、一欄是一天,而在輸入資料中一列是一天,一欄則是一個時段。
- 第三部分從第 $n_K + n_J + 3$ 列開始至第 $n_K + n_J + 4$ 列結束,也就是最後的兩列資料。每列各存放 R 個整數,其中第 $n_K + n_J + 3$ 列依序存著 $i_1 \cdot i_2$ 直到 i_R ,依序表示提出請假需求的員工編號;第 $n_K + n_J + 4$ 列依序存著 $j_1 \cdot j_2$ 直到 j_R ,依序表示第 $i_1 \cdot i_2$ 直到 i_R 號員工希望請假的日期,舉例來說,若 $i_1 = 3$ 且 $j_1 = 5$,代表第一項請假需求為第三號員工希望在第五天請假。

每一列中的任兩個數字之間都用一個空白鍵隔開。已知 $10 \le n_I \le 100 \cdot n_J \in \{28, 29, 30, 31\} \cdot 5 \le n_K \le 30 \cdot L \in \{8, 9, 10\} \cdot 1 \le w_1 \le 100 \cdot 1 \le w_2 \le 100 \cdot 0 \le R \le n_i n_j \cdot s_{k,t} \in \{0, 1\} \cdot 0 \le d_{j,t} \le n_I \cdot 1 \le i_r \le n_I \cdot 1 \le j_r \le n_J$ 。

舉例來說,假設班別和人力需求資訊如圖 1 和圖 2 所示、共有 $n_I = 40$ 位員工、計畫進行次月共 $n_J = 31$ 天的排班、給定的班型(不含休假)有 $n_K = 11$ 種、勞基法規定次月每一位員工的休假不得少於 L = 8 天。此外,我們總共蒐集了 20 份員工的請假需求,且目標函數的權重 w_1 、 w_2 分別為 3 與 4。則一個合理的輸入格式會是

40 31 11 8 3 4 20

...班型安排...

...人力需求...

3 5 7 2 9 16 39 36 24 35 40 1 8 12 37 13 31 22 21 20

2 5 8 3 5 2 8 5 9 9 1 3 6 7 4 2 2 4 6 3

因為版面有限,完整的範例輸入請參見附件的 inputExample.txt。

讀入資料後,請根據題目的描述,安排接下來 n_J 天的排班計畫。安排好班表後,請依照如下格式輸出你的計畫。你總共需要輸出 n_I 列,每一列依序輸出 $x_{i,1}$ 、 $x_{i,2}$ 直到 x_{i,n_J} ,代表第 i 號員工在接下來 n_J 天中,每一天被指派的班型為何,因此你的每個 $x_{i,j}$ 都應該要是 $0 \cdot 1 \cdot 2$ 直到 n_K 的其中一個整數。輸出時,請在每一列都用一個逗號隔開相鄰的兩個整數。

舉例來說,這邊安排了一套愚蠢但可行的班表,因為版面有限,輸出如附件outputExample.txt 所示。根據這個計畫,總缺工人數為 3456,未滿足的休假需求總數為 9,七天內的超額夜班總數為 2583,因此目標函數值為

 $3456 + 3 \times 9 + 4 \times 2583 = 13815 \circ$

3 評分原則

這一題的其中 75 分會根據程式運算的結果給分。你的程式不需要找出真的能最小化目標函數的最佳方案(optimal solution)。只要你的程式在時間和記憶體限制內跑完、輸出符合格式規定,且確實是一組可行方案(feasible solution,意即符合上文提及的三項規定,每一位員工都有休息至少 L 天、每連續七天至少休假一天、一天中恰被指派一種班型),就會得到分數。對於每一組輸入,PDOGS 會檢查你的輸出,如果輸出格式不合乎要求或方案不可行,則在該筆測試資料會得到零分;如果合乎要求,且目標式值非負,則對每筆測資,我們依下一段的方式計分。

首先,助教會寫兩個程式,在 PDOGS 上分別取名為「TA-smart」和「TA-naive」,前者的演算法會稍微有點聰明,後者的則是相當陽春(但不愚蠢)。針對某一筆測試資料,假設 z 是你這一組的班表得到的目標函數值、 z_0 是「TA-naive」得到的目標函數值、 z_1 是所有組(包含「TA-smart」)得到的目標函數值中最低的,則你的組在這筆測資的得分就是

 $1.5 + 1.5 \left(\frac{\max\{z_0 - z, 0\}}{z_0 - z_1} \right) \circ$

换言之,你的班表(相對於全班)愈好,就愈能得到好成績。好消息是,如公式所示,如果 $z_0 \le z$,亦即你的班表比「TA-naive」的還要糟,那你的組會得到的是 0 分而不會被倒扣 3 。

寫程式之外,每組還需要合力用中文或英文寫一份書面報告(所謂「寫」,就是用電腦打的意思),以組為單位上傳 PDF 檔至 NTU COOL。在報告裡請用文字描述你的演算法(可以用 pseudocode 但不能直接貼 code)、系統的設計(哪個函數做什麼、程式執行的流程等等)、分工方式(誰寫哪個函數、誰負責指揮、誰負責寫書面報告、誰負責買便當等等;當然一個人可以又買便當又寫程式),以及每個人的簡單心得感想。報告不可以超過八面 A4 紙。書面報告佔 25 分。這份專案截止後,書面報告才會被批改。

此外,授課團隊會考慮 PDOGS 上的得分以及書面報告上的解法,邀請若干組在 合適的上課時間,每組用 10 分鐘跟全班同學介紹自己的解法。被邀請的組可以婉拒上 台報告,但上台報告的組可以獲得專案總成績 5 分的額外加分。

4 繳交方式

請修課的同學們自行組成**每組三至四人**的小組,以組為單位繳交你們的程式和報告。

有兩件事需要注意。首先,系統會以該組內任意一位同學的最後一次上傳得到的分數,做為該組所有人的分數,所以愈傳愈低分是有可能的。其次,原則上 PDOGS 不限制兩次上傳間的時間間隔,但如果有許多組在同個時間大量地上傳執行時間很長的程式,導致 PDOGS 大塞車,屆時我們會對兩次上傳的時間間隔做出限制。

繳交組員資訊的截止時間是 10 月 25 日早上八點,繳交程式的截止時間是 11 月 15 日早上八點,繳交書面報告的截止時間是 11 月 17 日早上八點。錯過分組期限扣 10 分,錯過書面報告期限一天內扣 10 分,超過一天扣 25 分。

³在附件的 TA-naive.pdf 有敘述 TA-naive 的演算法概要與 pseudocode 供大家參考。如大家所見,確實是相當陽春的演算法,大家的程式要贏過 TA-naive 應該不是太困難。