コールセンターのスケジューリングにおける必要人員のシミュレーションと整数計画問題

ーオペレーターの不足を背景にー

首都大学東京　都市教養学部　都市教養学科　経営学系　経済学コース　４年

蘭牟田　佳佑

はじめに

・カスタマーとベンダーを結ぶ役割としてコールセンターはとても重要な役割を担っている。しかし昨今、**オペレーターの不足/離職率の高さ**などが問題となっている

・今回は上記の問題を少しでも改善できるよう、オペレーターの希望を取り入れたシフト表を作成できないかと考え研究を行った

＊定式化は本資料末にまとめて記載する

＊実行結果であるシフト表は扱っているオペレーターが多く全体を添付すると文字がつぶれてしまい解読ができないため、省略し、1～10のオペレーターのシフト表のみ添付する

＊シミュレーションに必要な「入電数のデータ、オペレータの人数」は実際に働いている方々へのインタビューに基づいて作成された架空のデータである

コールセンターとは

＜定義＞

「コールセンターとは、企業などの部署や施設の一つで、電話による外部との連絡や応対を集中的に取り扱うところ」

※ IT用語辞典e-words
(http://e-words.jp/w/コールセンター.html)より引用

＜補足＞

・業務内容としては大きくインバウンドとアウトバウンドに分かれるが、今回は**インバウンド業務のみ**を扱う

・様々な企業がコールセンターを設置しているがその目的は「**顧客満足度**」の向上である

・入電の内容は商品/契約の注文/質問/意見・クレーム等が挙げられる

コールセンターの現状

＜問題点＞

・コールセンターの現場では大きな問題が二つあると言われている[2]

⇨一つ目は**オペレーターの不足/離職率の高さ**である

マニュアルが多く仕事のストレス(特にクレーム対応)が高いため、1年で3割が退職する[3]とも言われており、一年を通して求人を出している

⇨二つ目は**管理する人(SV)の仕事量の多さ**である

対応にミスがないかの確認作業、オペレーターのシフト作成、オペレーター教育用の資料作成、クレーマーによっては上席対応など接客をしながらの事務作業が大量にある

研究の背景

オペレーターの人数不足を改善できる研究はないのか？

⬇

現場では**マネジメントサイドの意見が重視**される事が多いが、

オペレーターの働きやすさも重視する事でオペレーターの離職率が減ると仮定

⬇

オペレーターは自分の希望が反映されたシフト表だと働きやすくなる

⬇

しかし、オペレーターの希望が多くなると

シフト作成に時間がかかったり、取り入れられない希望も出てきてしまう

⬇

オペレーターの希望を取り入れつつ、必要な人員も確保したシフト作成を支援する研究であれば、

オペレーター不足に悩んでる現場にとって価値のある研究になる(働き方改革の一環にもなる)

研究対象とする「現場」

・コールセンターは様々な業界・会社で運営されているため、すべての会社に適用できるような研究を行うことは現実的ではない。そのため、今回の研究で扱う現場を選定する

＜会社情報＞

・対象とする会社はインフラ(ガス・電気)会社である

・インフラ会社は多くの会社が引越しシーズンである**3月**に入電数のピークを迎える

・3月に向けて新人を雇わなければいけないが、雇う新人は最小限に抑えたい、という希望が責任者にはある

・仮定として**現在雇用しているオペレーターは3月までは辞めない**とする

研究対象とする「現場」

＜オペレーター情報＞

・シフトは早番(9:00-17:00)と遅番(11:00-19:00)の二つである

・現在いるベテランオペレーターは早番が26人、遅番が21人である

・応答率は**95%**を切らないことを目標にしている

＜休憩＞

昼休憩が**1回**と10分休憩が**3回**ある

(昼休憩)

早番:11:30～12:30 or 12:30～13:30

遅番:13:30～14:30 or 14:30～15:30

2つの時間帯の内どちらかで1時間休憩を取る

(10分休憩)

早番:10:00～11:00/14:30～15:30/15:30～16:30

遅番:12:30～13:30/15:30～16:30/17:00～18:00

それぞれ1時間の中で10分ずつローテーションで回す

本研究の概要

① 研究対象現場のシミュレーションの実施

② 研究対象月(3月)のシフト作成

③ オペレーターの希望の反映(オペレーターの希望に関する制約式の追加)

④ 緩和した制約の重要度に応じたシフト作成

⬇ 本研究はシフト作成の手法を確立させることが目的だが、ユーザーの使いやすさを考慮し、ユーザーが制約を選択する際の参考となるように以下の研究も行った

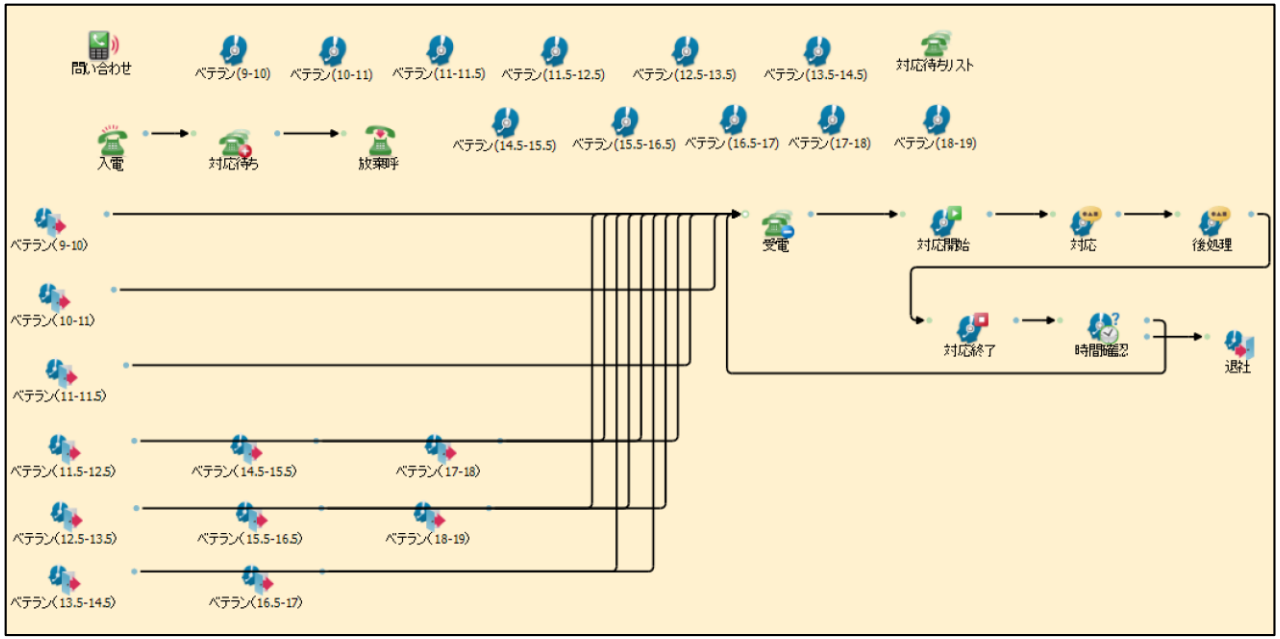
⑤ ソフト制約の影響度の比較

⑥ シフト表の比較

＊シミュレーションはS4 Simulation System、シフト作成はNumerical Optimizerを使用した(NTTデータ数理システム様より貸与)

①シミュレーション　ーモデル化ー

＜モデルの作成＞



①シミュレーション　ーモデル化ー

＜モデルの作成＞

↓入電数のデータ

	A	B	C	D
1	時間帯	平均(秒)	本数	
2	9	36	100	
3	10	40	90	
4	11	40	90	
5	12	36	100	
6	13	48	75	
7	14	48	75	
8	15	48	75	
9	16	48	75	
10	17	48	75	
11	18	72	50	
12		合計	805	
13				

＜補足＞

・休憩を反映させるためオペレーターの一回の勤務時間を11回に分割する。それぞれの勤務時間はアイテム名に記載

・対応時間は**平均10分**、後処理時間は**平均3分**の指数分布に従うとする

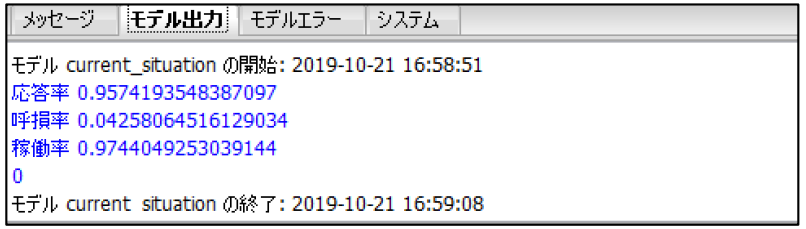
・入電数のデータは指数分布に従うように変換する

＜実行方法＞

入電数のデータを基にS4 Simulation Systemを用いて、応答率が95%を切らないようなオペレーター数をシミュレーションする
(応答率が95%になるまで**オペレーターの人数を変化させる**)

①シミュレーション　－結果－

＜結果＞



ベテランー遅番：　15人(シフト)/26人(全員)

ベテランー新人：　12人(シフト)/21人(全員)

応答率：95.74%

呼損率：4.26%

上記の結果を、現在の現場の状況とする

①シミュレーション　ーピーク期verー

＜ピーク期へのシミュレーション(**入電数のみを変化**させてシミュレーション実行)＞

⇨もし現在の人数のまま3月のピーク期を迎えたら、、、

↓入電数のデータ(一番入電数が多い場合を想定)　↓シミュレーションの結果

	A	B	C	D
1	時間帯	平均(秒)	本数	
2	9	10	360	
3	10	11	327.2727	
4	11	11	327.2727	
5	12	12	300	
6	13	12	300	
7	14	12	300	
8	15	12	300	
9	16	12	300	
10	17	12	300	
11	18	14	257.1429	
12		合計	3071.688	
13				
14				

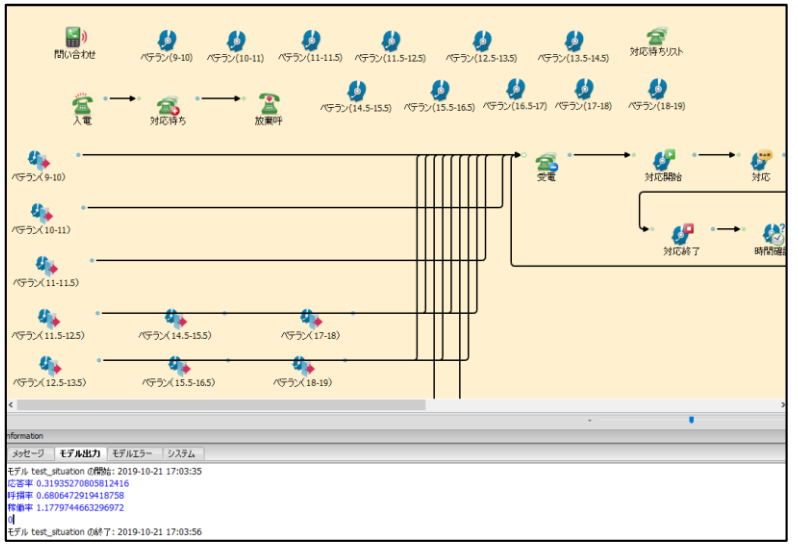
応答率が**31.94%**になる

⬇

今のままでは人員が足りない事が想定される

⬇

新人を何人雇用するか検討をしなければいけない



②ピーク期に向けたシフト作成

Ⅰ)想定される入電数を基に応答率が**90%**を超えるために必要な新人の人数をシミュレーションする ※ピーク期なので応答率は90%を目標とする

Ⅱ)シミュレーションの結果を基に必要な人員を確保しつつもハード制約を満たしたシフト表を作成する

※ハード制約：業務遂行上絶対に守らなければならない制約

　ソフト制約：制約を違反しても業務遂行はできるが、できるだけ満たして欲しい制約

②ーⅠ　シミュレーション　ー概要ー

＜前提＞

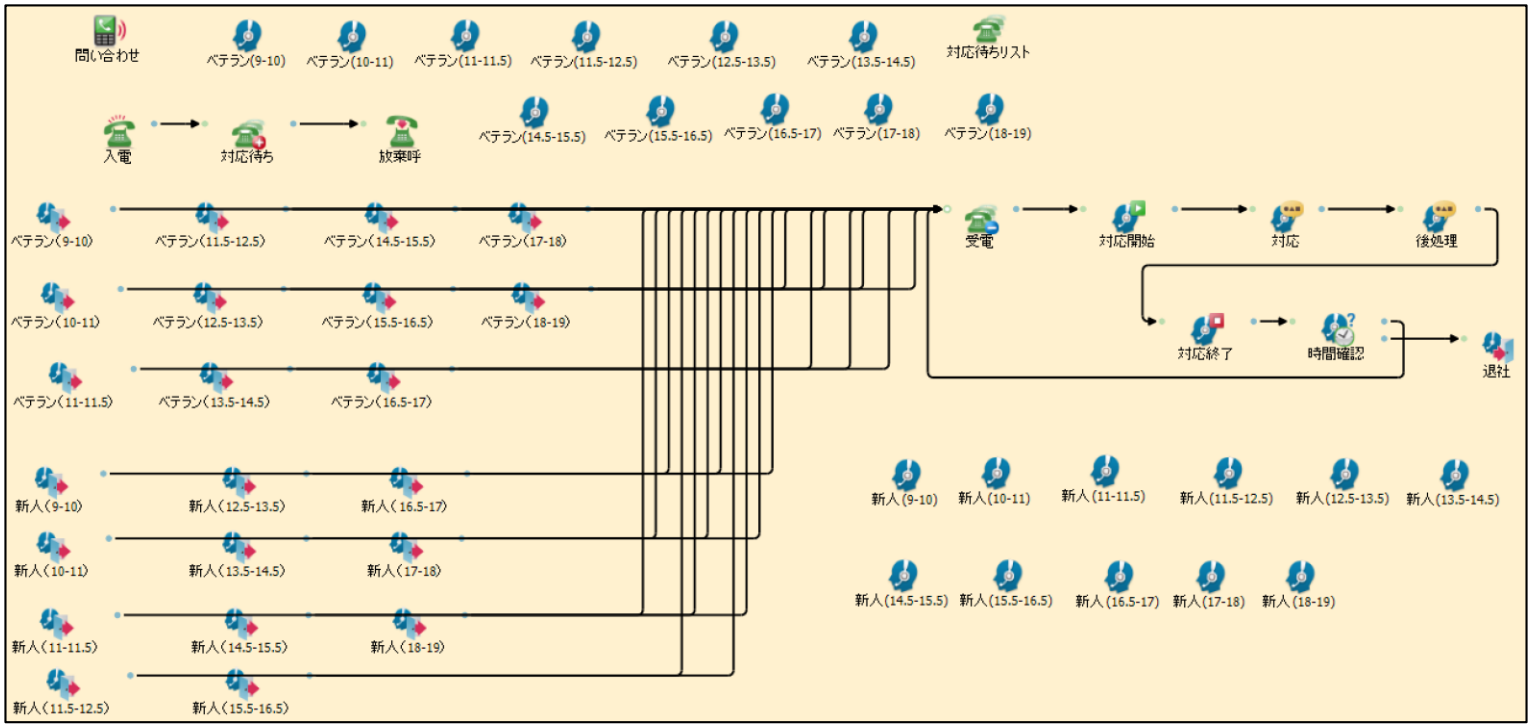
・ベテランー早番は17人、ベテランー遅番は14人出勤とし、ピーク期の必要人数はこの人数で固定する

・ベテランと新人の間で受付の種類や入電数の違いは無いとする

・新人の休憩回数はベテランと同じ回数とする

・新人の対応時間は平均15分、後処理時間は平均6分の指数分布に従うとする

②ーⅠ　シミュレーション　ーモデル化ー



②ーⅠ　シミュレーション　ーモデル化ー

＜シミュレーションの方法＞

日付：1日～31日、**新人ー早番、新人ー遅番**：5人～40人

上記の情報を一括実行機能を用いて自動的に変動させ、新人の必要人数をシミュレーションする

↓入電数のデータ(シミュレーションの際は指数分布に従うように変換する)

入電数(本)	3月1日	3月2日	3月3日	3月4日	3月5日	3月6日	3月7日	3月8日	3月9日	3月10日	3月11日	3月12日	3月13日	3月14日	3月15日	3月16日	3月17日	3月18日	3月19日	3月20日	3月21日	3月22日	3月23日	3月24日	3月25日	3月26日	3月27日	3月28日	3月29日	3月30日	3月31日
9時	200	180	180	210	190	190	190	200	180	180	250	240	240	240	250	230	230	300	290	290	230	300	280	280	350	340	340	340	330	300	300
10時	180	160	160	190	170	170	170	180	160	160	230	220	220	220	230	210	210	280	270	270	210	280	260	260	330	320	320	320	310	280	280
11時	180	160	160	210	170	170	170	180	160	160	230	220	220	220	230	210	210	280	270	270	210	280	260	260	330	320	320	320	310	280	280
12時	200	180	180	160	190	190	190	200	180	180	200	240	240	240	200	230	230	250	290	290	230	250	280	280	300	340	340	340	280	250	250
13時	150	130	130	160	140	140	140	150	130	130	200	190	190	190	200	180	180	250	240	240	180	250	230	230	300	290	290	290	280	250	250
14時	150	130	130	160	140	140	140	150	130	130	200	190	190	190	200	180	180	250	240	240	180	250	230	230	300	290	290	290	280	250	250
15時	150	130	130	160	140	140	140	150	130	130	200	190	190	190	200	180	180	250	240	240	180	250	230	230	300	290	290	290	280	250	250
16時	150	130	130	160	140	140	140	150	130	130	200	190	190	190	200	180	180	250	240	240	180	250	230	230	300	290	290	290	280	250	250
17時	150	130		160	140	140	140	150	130		200	190	190	190	200	180		250	240	240		250	230		300	290	290	290	280	250	
18時	100	80		110	90	90	90	100	80		150	140	140	140	150	130		200	190	190		200	180		250	240	240	240	230	200	
合計	1610	1410	1200	1680	1510	1510	1510	1610	1410	1200	2060	2010	2010	2010	2060	1910	1600	2560	2510	2510	1600	2560	2410	2000	3060	3010	3010	3010	2860	2560	2110

②ーⅠ　シミュレーション　ー結果ー

＜結果＞

一括実行の結果から応答率を90%を超える組み合わせを見つける　例)↓

反復回数	レプリケーション番号	operators3	operators4	date	1. 応答率 ▼	稼働率
85	2	13.000	10.000	1.000	0.907	1.081
59	1	10.000	14.000	1.000	0.907	1.088
47	1	9.000	12.000	1.000	0.906	1.136
86	1	13.000	11.000	1.000	0.906	1.100
94	1	14.000	9.000	1.000	0.906	1.091
58	2	10.000	13.000	1.000	0.906	1.089
84	2	13.000	9.000	1.000	0.905	1.094
48	0	9.000	13.000	1.000	0.905	1.109
39	2	8.000	14.000	1.000	0.905	1.090
85	1	13.000	10.000	1.000	0.904	1.101
29	1	7.000	14.000	1.000	0.904	1.100
67	0	11.000	12.000	1.000	0.904	1.091
67	2	11.000	12.000	1.000	0.904	1.107
66	2	11.000	11.000	1.000	0.904	1.086
64	0	11.000	9.000	1.000	0.903	1.139
83	1	13.000	8.000	1.000	0.903	1.119
55	1	10.000	10.000	1.000	0.903	1.140
38	1	8.000	13.000	1.000	0.903	1.126
57	0	10.000	12.000	1.000	0.902	1.121
19	2	6.000	14.000	1.000	0.902	1.097
86	0	13.000	11.000	1.000	0.902	1.086
47	2	9.000	12.000	1.000	0.902	1.095
56	1	10.000	11.000	1.000	0.902	1.133
56	0	10.000	11.000	1.000	0.902	1.115
66	0	11.000	11.000	1.000	0.901	1.094
66	1	11.000	11.000	1.000	0.901	1.112
38	2	8.000	13.000	1.000	0.901	1.108
74	2	12.000	9.000	1.000	0.900	1.107
65	2	11.000	10.000	1.000	0.900	1.116
19	1	6.000	14.000	1.000	0.899	1.110
55	2	10.000	10.000	1.000	0.899	1.122
54	1	10.000	9.000	1.000	0.899	1.160

ここでシミュレーションされた人数を基に次項以降のシフト作成を行う

初めて90%を超えるライン
人数の組み合わせが2パターン以上出た場合は早番と遅番の人数の差が少ない方を選択する
←

シミュレーション結果をまとめた必要な新人の人数↓

	オペ3(シミュ)	オペ4(シミュ)	合計(シミュ)
1日	11	10	21
2日	10	9	19
3日	11	11	22
4日	11	11	22
5日	9	10	19
6日	9	10	19
7日	9	10	19
8日	11	11	22
9日	10	9	19
10日	11	11	22
11日	21	13	34
12日	26	10	36
13日	26	10	36
14日	26	10	36
15日	21	13	34
16日	15	18	33
17日	20	11	31
18日	27	26	53
19日	25	27	52
20日	25	27	52
21日	20	11	31
22日	27	26	53
23日	26	25	51
24日	27	25	52
25日	36	35	71
26日	35	35	70
27日	35	35	70
28日	35	35	70
29日	32	33	65
30日	26	28	54
31日	27	25	52
小計	660	580	
合計	1240		

②ーII　シフト作成　－概要－

＜シフト作成の方法＞

Numerical Optimizerを用いて②ーIIで算出した**オペレーター数を下回らないように**、また、ハード制約を満たすようにシフト作成を行う

＜目的関数＞

制約の違反を変数に代入するようにし、その変数の値の最小化(現在では必要オペレータ数を上回った分のみ)

Minimize:$\sum_{d\in D} X_1 + \sum_{d\in D} X_2 + \sum_{d\in D} X_3 + \sum_{d\in D} X_4$

②－Ⅱ　シフト作成　－制約－

＜ハード制約＞

① 一日の中で早番と遅番は**一回**しか担当できない

② 週の勤務日数が**5日**を超えてはならない

③ 希望休は守るようにする(ベテランは5日、新人は3日)(希望休は**乱数**で発生させた)

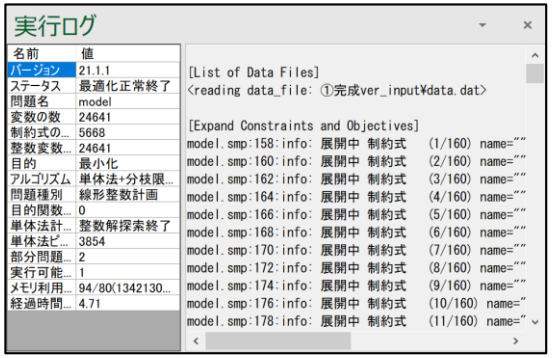
④ 合計出勤日が**22日**を下回るようにする

⑤ 必要オペレーター数を下回らないようにする(上回った分は目的関数値に加算)

②－Ⅱ　シフト作成　－結果－

＜結果＞

shift(ベテラン)	1	2	3	4	5	6	7	8	9	10
3/1	0	1	1	1	0	0	1	1	1	0
3/2	0	1	1	1	0	1	1	1	1	0
3/3	0	0	1	1	1	1	1	1	1	1
3/4	1	1	1	0	1	1	1	1	1	1
3/5	1	1	1	1	1	0	1	1	1	1
3/6	1	1	0	1	1	1	0	0	0	1
3/7	1	0	0	0	1	1	0	0	0	1
3/8	0	1	0	0	0	1	0	0	0	1
3/9	0	1	1	1	1	0	1	1	1	1
3/10	1	1	1	1	1	0	1	1	1	1
3/11	1	0	1	1	1	1	1	1	1	1
3/12	1	0	1	1	0	1	1	1	1	1
3/13	1	1	0	0	1	1	0	0	1	0
3/14	1	1	1	1	1	1	1	1	0	0
3/15	1	0	0	1	1	1	0	1	1	1
3/16	1	1	1	1	1	1	1	1	1	0
3/17	0	1	1	1	1	1	1	1	1	0
3/18	1	1	1	1	1	1	1	1	1	0
3/19	1	1	1	1	1	0	1	1	0	1
3/20	1	0	0	0	0	1	1	0	1	1
3/21	0	1	1	0	0	0	0	0	0	1
3/22	1	1	0	1	0	1	1	0	1	1
3/23	1	1	0	1	1	1	1	1	0	1
3/24	1	1	1	1	1	0	1	0	1	0
3/25	0	1	1	0	1	1	0	1	1	1
3/26	1	0	1	1	1	1	1	1	1	1
3/27	1	1	1	1	1	1	1	1	1	1
3/28	0	0	1	0	0	0	0	0	0	0
3/29	1	1	1	1	1	1	1	1	0	1
3/30	1	1	1	1	0	1	1	1	1	1
3/31	1	0	0	0	0	0	0	1	1	1



業務に必要なシフト表は完成

⬇

オペレーターによって出勤日数や時間帯に偏りがある

⬇

オペレーターの希望を取り入れたシフト作成が必要

③ソフト制約式の追加　－制約－

＜追加するソフト制約＞

① 合計出勤日の**下限**を指定
② 日曜日の出勤回数は**3日**まで
③ 連続勤務日数が**5日**を下回るようにする
④ 早番・遅番の**最低出勤日**を指定
⑤ 遅番の次の日に早番は入れないようにする
⑥ 禁止ペアが一緒にならないようにする
⑦ シフトの間隔が**3日以上**にならないようにする

＜目的関数＞

制約の違反を変数に代入するようにし、その変数の値の最小化
(現段階ではソフト制約には変数を与えないため、目的関数値は必要オペレーター数を上回った分のみ)

Minimize:$\sum_{d\in D} X_1 + \sum_{d\in D} X_2 + \sum_{d\in D} X_3 + \sum_{d\in D} X_4$

③制約式の追加　－結果－

＜結果＞

実行不可能になる

＜対応策＞

Ⅰ)どこまでの制約なら解を導出できるか検証

Ⅱ)ソフト制約式に変数を与え、制約を超えた分はペナルティとして**変数に値を代入**させることで制約式を緩和する

③－Ⅰ　解のでる制約式の検証　－目的関数－

＜概要＞

ソフト制約を**オペレーターから見た**重要度の高さで並べ替え、

実行可能解が出るまで、制約式を減らしていく。

(19ページに記載されている①～⑦の番号は重要度の順である)

※本研究で扱うオペレーターから見た重要度の高さの並び替えは実際に働いている方へのインタビューをもとに並び替えを行った

＜目的関数＞

制約の違反を変数に代入するようにし、その変数の値の最小化

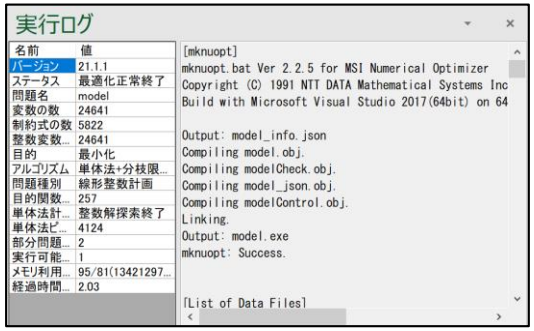
(現段階ではソフト制約には変数を与えないため、目的関数値は必要オペレーター数を上回った分のみ)

Minimize:$\sum_{d\in D} X_1 + \sum_{d\in D} X_2 + \sum_{d\in D} X_3 + \sum_{d\in D} X_4$

③－Ⅰ　解の出る制約式の検証　－結果－

＜結果＞

shift(ベテラン)	1	2	3	4	5	6	7	8	9	10
3/1	0	0	1	0	0	1	0	1	1	1
3/2	1	1	1	1	1	1	1	1	1	0
3/3	1	0	1	1	1	1	1	1	1	1
3/4	1	1	0	0	0	1	1	1	1	1
3/5	0	1	1	1	1	0	1	1	1	1
3/6	0	1	1	1	1	1	1	0	0	1
3/7	1	1	0	1	1	0	0	0	0	0
3/8	1	1	0	1	1	1	0	0	0	1
3/9	0	1	1	1	0	0	1	1	1	1
3/10	1	1	1	1	1	0	1	1	1	1
3/11	1	0	1	1	1	1	1	1	1	1
3/12	0	0	1	0	0	1	1	1	1	1
3/13	1	1	0	0	1	1	0	0	1	0
3/14	1	1	1	1	1	1	1	1	0	0
3/15	1	0	0	1	1	1	0	1	1	1
3/16	1	1	1	1	1	1	1	1	1	0
3/17	0	0	1	1	1	1	1	1	1	0
3/18	1	1	1	1	1	1	1	1	1	0
3/19	1	1	1	1	1	0	1	1	0	1
3/20	1	1	0	0	0	0	1	0	1	1
3/21	0	1	1	0	0	1	0	0	0	0
3/22	0	1	1	1	0	1	1	0	1	1
3/23	0	1	0	1	0	1	1	1	0	1
3/24	1	0	1	1	1	0	1	0	1	0
3/25	1	1	0	0	1	1	0	0	1	1
3/26	1	0	1	1	1	0	1	1	0	1
3/27	1	1	1	1	1	1	1	1	0	0
3/28	1	1	1	0	1	1	0	0	0	0
3/29	1	1	1	0	1	1	1	0	0	1
3/30	1	1	1	1	0	0	1	0	0	0
3/31	1	0	0	1	0	1	0	0	0	0



①の制約を含ませると解が出ることが判明
(②～⑦を含ませると実行不可能)

⬇

制約式の緩和をしないとオペレーターの希望が反映されていないシフト表になってしまう

⬇

Ⅱ)制約式を緩和させる を実行

③ー II　制約式の緩和　ー目的関数ー

＜概要＞

全てのソフト制約式に変数を追加し、制約を違反した分だけその変数に値が代入されるようにする。そうすることで全ての制約式を含んだ上で最適なシフト表が導出できる

＜目的関数＞

制約の違反を変数に代入するようにし、その変数の値の最小化

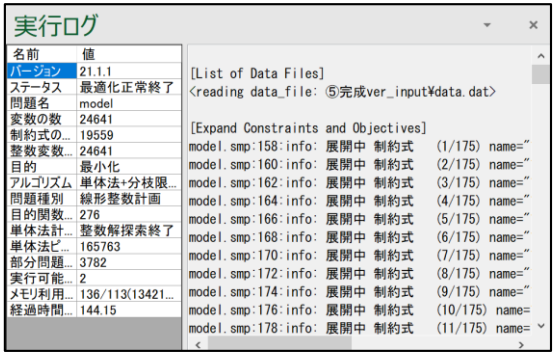
Minimize:

$$\sum_{d\in D} X_1+\sum_{d\in D} X_2+\sum_{d\in D} X_3+\sum_{d\in D} X_4+\sum_{s\_1\in S\_1} X_5+\sum_{s\_2\in S\_2} X_6+\sum_{s\_1\in S\_1} X_7+\sum_{s\_2\in S\_2} X_8$$
$$+\sum_{d\in D}\times\sum_{s\_1\in S\_1} X_9+\sum_{d\in D}\times\sum_{s\_2\in S\_2} X_{10}+\sum_{s\_1\in S\_1} X_{11}+\sum_{s\_1\in S\_1} X_{12}+\sum_{s\_2\in S\_2} X_{13}+\sum_{s\_2\in S\_2} X_{14}+$$
$$\sum_{d\in D}\times\sum_{s\_1\in S\_1} X_{15}+\sum_{d\in D}\times\sum_{s\_2\in S\_2} X_{16}+\sum_{d\in D} X_{17}+\sum_{d\in D}\times\sum_{s\_1\in S\_1} X_{18}+\sum_{d\in D}\times\sum_{s\_2\in S\_2} X_{19}$$

③－II　制約式の緩和　－結果－

＜結果＞

shift(ベテラン)	1	2	3	4	5	6	7	8	9	10
3/1	1	1	1	1	0	0	0	1	1	1
3/2	0	1	0	1	1	1	1	0	1	0
3/3	1	1	1	1	1	1	1	1	0	1
3/4	1	0	1	0	0	1	1	1	0	0
3/5	1	1	1	1	1	0	0	1	1	1
3/6	0	1	1	0	1	1	1	1	1	1
3/7	1	0	0	1	1	1	1	0	1	1
3/8	1	0	0	1	1	1	1	1	0	0
3/9	0	1	1	0	0	0	1	1	1	1
3/10	1	1	1	1	1	0	0	0	1	1
3/11	1	0	1	1	1	1	1	1	1	1
3/12	1	0	1	1	0	1	1	1	0	0
3/13	0	1	0	0	0	0	0	0	1	1
3/14	1	0	1	1	1	0	1	1	1	1
3/15	1	1	1	1	1	1	0	0	0	1
3/16	1	1	0	1	1	0	1	1	1	0
3/17	0	1	1	0	1	1	1	1	0	0
3/18	1	1	1	1	0	1	1	1	1	0
3/19	1	0	1	0	1	0	1	1	0	1
3/20	0	0	1	1	1	1	0	0	1	0
3/21	0	1	0	1	0	1	1	1	1	1
3/22	1	1	1	1	0	0	1	0	1	1
3/23	1	0	1	1	1	1	0	1	0	1
3/24	1	1	0	0	0	0	1	0	1	0
3/25	1	1	1	0	0	1	0	1	1	1
3/26	0	0	0	1	1	1	1	1	0	1
3/27	0	1	1	1	1	1	1	1	1	0
3/28	1	1	1	0	1	0	0	0	1	1
3/29	1	1	1	0	1	1	1	0	0	1
3/30	1	1	1	1	0	1	1	1	1	1
3/31	0	0	0	1	0	1	0	1	1	1



③ーII　制約式の緩和　－違反した制約－

①合計出勤日の下限を指定(新人)⇒1日下回った人数:12人、2日下回った人数:2人

②日曜日の出勤は3回まで(ベテラン)⇒4回出勤した人数:10人、5回出勤した人数:2人

⑦シフトの間隔が3日以上にならないようにする(ベテラン)⇒3日空いた人数:1人
⑦シフトの間隔が3日以上にならないようにする(新人)⇒3日空いた人数:1人

・新人ー早番の超過出勤した合計人数：142人

・新人ー遅番の超過出勤した合計人数：100人

③－Ⅱ　制約式の緩和　－考察－

＜新人の雇用の難しさ＞

ピーク期の必要人数は一ヶ月の中でも日によってかなりの差があるため

制約式①の違反と早番・遅番の出勤超過人数の違反が多くなるのは必然である

⬇

早番・遅番の出勤超過人数の違反の重みをつけることで改善できないのか？

⬇

次項「**④重要な制約に重みをつける**」で実験をする

④重要な制約式に重みをつける　－概要－

＜重みをつける制約式＞

オペレーターからすると、休日は休みたいという観点から日曜日の出勤は3日までという制約を重要だと考えており、

会社側からすると、人件費の観点から必要人員は上回らない事が重要だと考えていると仮定する

よって、上記の制約に重み**1.5**をかけてシフト作成を行う

④重要な制約式に重みをつける　－目的関数－

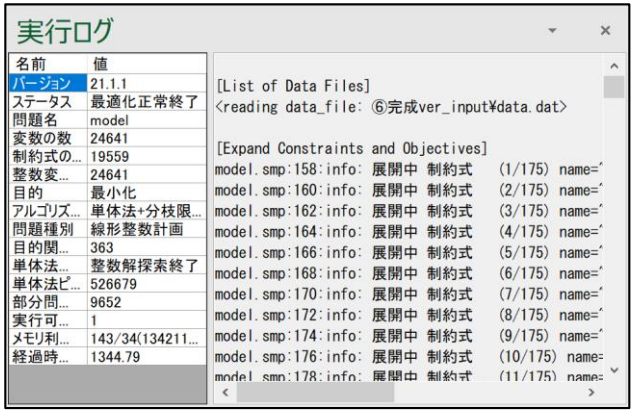
Minimize:

$$\sum_{d\in D} P * X_1 + \sum_{d\in D} P * X_2 + \sum_{d\in D} P * X_3 + \sum_{d\in D} P * X_4 + \sum_{s\_1\in S\_1} X_5$$
$$+ \sum_{s\_2\in S\_2} X_6 + \sum_{s\_1\in S\_1} P * X_7 + \sum_{s\_2\in S\_2} P * X_8 + \sum_{d\in D}\times\sum_{s\_1\in S\_1} X_9 + \sum_{d\in D}\times\sum_{s\_2\in S\_2} X_{10}$$
$$+ \sum_{s\_1\in S\_1} X_{11} + \sum_{s\_1\in S\_1} X_{12} + \sum_{s\_2\in S\_2} X_{13} + \sum_{s\_2\in S\_2} X_{14} + \sum_{d\in D}\times\sum_{s\_1\in S\_1} X_{15}$$
$$+ \sum_{d\in D}\times\sum_{s\_2\in S\_2} X_{16} + \sum_{d\in D} X_{17} + \sum_{d\in D}\times\sum_{s\_1\in S\_1} X_{18} + \sum_{d\in D}\times\sum_{s\_2\in S\_2} X_{19}$$

④重要な制約式に重みをつける　－結果－

＜結果＞

shift(ベテラン)	1	2	3	4	5	6	7	8	9	10
3/1	1	0	0	1	0	0	0	0	1	1
3/2	1	1	1	1	1	1	1	1	1	0
3/3	1	1	1	0	1	1	0	1	0	1
3/4	1	1	0	0	1	1	1	1	0	1
3/5	0	1	1	1	1	0	1	1	1	1
3/6	1	0	1	1	0	1	1	0	1	1
3/7	0	1	0	1	1	1	1	1	1	0
3/8	1	1	0	1	0	1	0	1	0	1
3/9	0	0	1	0	0	0	1	1	1	1
3/10	1	1	1	1	1	0	1	0	1	1
3/11	1	0	0	0	1	1	1	1	0	0
3/12	1	0	1	1	0	1	1	1	1	1
3/13	1	1	0	0	1	1	0	0	1	0
3/14	0	1	1	1	1	1	1	1	1	1
3/15	1	0	0	1	0	0	0	1	1	1
3/16	1	1	1	1	1	1	1	1	0	0
3/17	0	1	1	1	0	1	1	1	1	0
3/18	1	0	0	0	1	1	1	0	1	0
3/19	1	1	1	1	0	0	1	1	0	1
3/20	1	0	1	1	1	1	0	1	1	1
3/21	0	1	1	0	1	1	1	0	1	1
3/22	1	1	0	1	0	0	1	0	1	1
3/23	1	1	1	1	1	1	1	1	0	0
3/24	1	0	1	1	1	0	1	0	1	0
3/25	1	1	1	0	1	0	0	1	1	1
3/26	0	0	0	1	0	1	1	1	0	1
3/27	1	1	1	1	1	1	1	0	1	1
3/28	0	1	1	0	1	0	0	0	1	1
3/29	1	1	1	0	1	1	1	1	0	0
3/30	1	1	1	1	0	1	0	0	1	0
3/31	0	0	0	0	0	1	1	1	1	1



④重要な制約式に重みをつける　－違反した制約－

①合計出勤日の下限を指定(新人)⇒1日下回った人数:31人、2日下回った人数:20人、3日下回った人:8人

②日曜日の出勤は3回まで(ベテラン)⇒4回出勤した人数:12人、5回出勤した人数:1人

⑦シフトの間隔が3日以上にならないようにする(ベテラン)⇒3日空いたベテラン:3人

⑦シフトの間隔が3日以上にならないようにする(ベテラン)⇒3日空いたベテラン:1人

・新人ー早番の超過出勤した合計人数：52人

・新人ー遅番の超過出勤した合計人数：110人

④重要な制約式に重みをつける　－考察－

＜新人の雇用の難しさ＞

前項「③－Ⅱ制約式の緩和」で考察した

早番・遅番の出勤超過人数の違反の重みをつけることで改善できないのか？

について

⬇

制約の違反度を見ると重みをつけたことにより、新人の超過出勤は改善されたといえる

⬇

しかし、新人の**最低出勤日の制約違反**が大きくなってしまった。

⬇

ピーク期の新人オペレーターの週の出勤日が少なくなってしまうのは仕方がないこと

であると考察できる（人件費を重視するのであれば）

⑤各ソフト制約の影響度を比較　ー概要ー

＜概要＞

各ソフト制約が目的関数値に与える影響度を算出する

⬇

算出された結果を基に

解の導出ができる制約式の組み合わせの中で

最も多くの制約式が含まれるような組み合わせを見つける

⑤各ソフト制約の影響度を比較　ー目的関数ー

＜比較方法＞

①　全てのソフト制約式を**重みを与えず、緩和も行わずに**適用させた状態で解を導出

②　ソフト制約を一つずつ除いた状態での解をそれぞれ導出

③　①の目的関数値と②の目的関数値の差を計算する

④　目的関数値の差によって各ソフト制約式の影響度を比較する

＜目的関数＞

制約の違反を変数に代入するようにし、その変数の値の最小化

（今回はソフト制約式に変数を与えないため、目的関数値は必要オペレーター数を上回った分のみ）

Minimize:$\sum_{d\in D} X_1 + \sum_{d\in D} X_2 + \sum_{d\in D} X_3 + \sum_{d\in D} X_4$

⑤各ソフト制約の影響度を比較　ー結果ー

実行ログ

名前	値
バージョン	21.1.1
ステータス	最適化正常終了
問題名	model
変数の数	24641
制約式の...	19559
整数変数...	24641
目的	最小化
アルゴリズム	単体法+分枝限...
問題種別	線形整数計画
目的関数...	276
単体法計...	整数解探索終了
単体法ピ...	165763
部分問題...	3782
実行可能...	2
メモリ利用...	136/113(13421...
経過時間...	144.15

↑全ての制約式

実行ログ

名前	値
バージョン	21.1.1
ステータス	最適化正常終了
問題名	model
変数の数	24641
制約式の数	19405
整数変数...	24641
目的	最小化
アルゴリズム	単体法+分枝限定...
問題種別	線形整数計画
目的関数値	181
単体法計...	整数解探索終了
単体法ピボ...	188541
部分問題数	4103
実行可能...	1
メモリ利用...	133/113(1342123...
経過時間(...	435.91

↑①の制約式を除いたログ

実行ログ

名前	値
バージョン	21.1.1
ステータス	最適化正常終了
問題名	model
変数の数	24641
制約式の数	19405
整数変数...	24641
目的	最小化
アルゴリズム	単体法+分枝限定...
問題種別	線形整数計画
目的関数値	262
単体法計...	整数解探索終了
単体法ピボ...	290276
部分問題数	11139
実行可能...	1
メモリ利用...	139/110(1342111...
経過時間(...	660.76

↑②の制約式を除いたログ

実行ログ

名前	値
バージョン	21.1.1
ステータス	最適化正常終了
問題名	model
変数の数	24641
制約式の数	15401
整数変数...	24641
目的	最小化
アルゴリズム	単体法+分枝限定...
問題種別	線形整数計画
目的関数値	276
単体法計...	整数解探索終了
単体法ピボ...	133836
部分問題数	4355
実行可能...	1
メモリ利用...	116/101(1342128...
経過時間(...	125.23

↑③の制約式を除いたログ

実行ログ

名前	値
バージョン	21.1.1
ステータス	最適化正常終了
問題名	model
変数の数	24641
制約式の数	19251
整数変数...	24641
目的	最小化
アルゴリズム	単体法+分枝限定...
問題種別	線形整数計画
目的関数値	276
単体法計...	整数解探索終了
単体法ピボ...	21762
部分問題数	101
実行可能...	2
メモリ利用...	126/109(1342131...
経過時間(...	43.13

↑④の制約式を除いたログ

実行ログ

名前	値
バージョン	21.1.1
ステータス	最適化正常終了
問題名	model
変数の数	24641
制約式の数	14939
整数変数...	24641
目的	最小化
アルゴリズム	単体法+分枝限定...
問題種別	線形整数計画
目的関数値	276
単体法計...	整数解探索終了
単体法ピボ...	23161
部分問題数	245
実行可能...	1
メモリ利用...	120/102(1342131...
経過時間(...	29.48

↑⑤の制約式を除いたログ

実行ログ

名前	値
バージョン	21.1.1
ステータス	最適化正常終了
問題名	model
変数の数	24641
制約式の数	19528
整数変数...	24641
目的	最小化
アルゴリズム	単体法+分枝限定...
問題種別	線形整数計画
目的関数値	276
単体法計...	整数解探索終了
単体法ピボ...	461404
部分問題数	14403
実行可能...	1
メモリ利用...	143/40(13421084...
経過時間(...	1018.77

↑⑥の制約式を除いたログ

実行ログ

名前	値
バージョン	21.1.1
ステータス	最適化正常終了
問題名	model
変数の数	24641
制約式の数	15093
整数変数...	24641
目的	最小化
アルゴリズム	単体法+分枝限定...
問題種別	線形整数計画
目的関数値	272
単体法計...	整数解探索終了
単体法ピボ...	270201
部分問題数	9350
実行可能...	1
メモリ利用...	124/44(13421182...
経過時間(...	550.81

↑⑦の制約式を除いたログ

⑤各ソフト制約の影響度を比較　－結果－

＜結果＞

・目的関数値の差(高い順で並べ替え)

①(95)＞②(10)＞⑦(4)＞③=④=⑤＝⑥(0)

＜参考＞

・計算時間の差(長い順で並び替え)

⑥(+874.62)＞②(+516.61)＞⑦(+406.66)＞①(+291.76)＞③(-18.92)

＞④(-101.02)＞⑤(-114.67)

・制約式の数の差(多い順で並び替え)

⑤(4620)＞⑦(4466)＞③(4158)＞②=④(308)＞①(154)＞⑥(31)

⑤各ソフト制約の影響度を比較　－結果－

＜最適解を見つけづらくする影響度が高い順＞

① 合計出勤日の下限を指定

② 日曜日の出勤回数は3日まで

⑦ シフトの間隔が3日以上にならないようにする

↓以下は差が0だったため、影響は与えていないと仮定

③ 連続勤務日数が5日を下回るようにする

④ 早番・遅番の最低出勤日を指定

⑤ 遅番の次の日に早番は入れないようにする

⑥ 禁止ペアが一緒にならないようにする

①、②、⑦の制約を除いて実行

⬇

解が導出される

⬇

目的関数値への影響が数ない順で制約を減らす

⬇

①、②の制約を除いた場合で解が導出された

⬇

次項⑥で含まれた制約(③～⑦)と
除かれた制約(①、②)との違いを検証

⑥　オペレーターの希望とシフト作成のしやすさ ー概要ー

＜概要＞

前項⑤で得られた制約の組み合わせを基に

解導出に**含めることができた制約**と**含めることができなかった制約**の違いを検証

⬇

どのようなオペレーターの希望だとシフト作成がしやすいかを考察する

⑥　オペレーターの希望とシフト作成のしやすさ
ー制約式の比較ー

＜含めることができなかった制約＞

・合計出勤日の下限を指定(③－Ⅱ,④で行った研究結果の制約違反から新人に原因があると考える)

・日曜日の出勤回数は3回まで(③ーⅡ,④で行った研究結果の制約違反からベテランに原因があると考える)

＜含めることができた制約＞

・シフトの間隔が3日以上にならないようにする

・連続勤務日数が5日を下回るようにする

・早番・遅番の最低出勤日を指定

・遅番の次の日に早番は入れないようにする

・禁止ペアが一緒にならないようにする

⑥　オペレーターの希望とシフト作成のしやすさ
ー考察ー

前ページの制約式の比較から考察出来ることが2つある

（※ピーク期という入電数の差が激しい状況において）

ⅰ）「合計出勤日の下限を指定」・「日曜日の出勤回数は3回まで」という制約を含められなかった

⇒ピーク期のように入電数の差が激しい状況だと、新人の必要オペレーター数が著しく少なくなってしまう日があるのに対して、ベテランはもともとの人数が少ないので、毎日人員が必要になる。そのため

出勤に関する希望はベテランの方が反映される可能性が高く、

休みに関する希望は新人の方が反映される可能性が高いと考えられる

⑥　オペレーターの希望とシフト作成のしやすさ
ー考察ー

ⅱ）「早番・遅番の最低出勤日を指定」・「遅番の次の日に早番は入れないようにする」という制約を含めることが出来た

⇒早番か遅番かという制約をどちらも含められた理由として、早番と遅番の**制約の重みが同じ**だったからと考えられる。そのため、

オペレーター(ベテラン・新人ともに)が**特定の日に早番で出勤したいか遅番で出勤したいかを希望する制約**は反映されやすいと考えられる

＜考察のまとめ＞

上記2点に適用できる希望であればオペレーターの希望をかなえつつ必要人員も確保したシフト表を作成できるはずである

本研究のまとめ・課題

本研究では「**オペレーターの希望を取り入れつつ、必要な人員も確保したシフト作成の手法を確立する**」ことを目的とし、シミュレーションと整数計画問題によるシフト作成を実施した。

現場によって取り入れるべき制約式や各制約の優先順位、違反した制約にかける重み、影響の度合いは異なるであろう。本研究の数値実験では、その数値や順位のある特定のケースについて計算結果を示したが、その**数値や順番を変えた場合**でも同様に計算をすることで、オペレーターの希望や必要人員を確保したシフト作成ができるよう、手法を確立した。

また、制約式の選定・優先順位付けに際し、どのような制約が適用させやすいのかについて調べて結果を示し、ユーザーが利用する際に参考とできるデータとして整理した。

今回は入電数がピーク期のインフラ(ガス・電気)会社に設置されているコールセンターを現場として実験を行ったので、ユーザーが使用する時期(閑散期等)、業種(飲食店等)によってはそのまま簡単に適用できない可能性がある。条件の狭い範囲ではなく、**もっと広い範囲で使えるよう一般化・汎用化すること**が、今後の課題として挙げられる。

今後の研究計画

今回の現場では一ヶ月の中で入電数の差が激しい業種で研究を行ったため、どうしても「新人の合計出勤日の下限を指定」という制約を違反する結果となってしまった。しかし、同じような問題を抱えている会社は多いはずである。

今後の研究としてこの問題を、**チャットボット**を導入することで改善はできないのか、という研究を行いたいと考えている。

具体的にはチャットボットの能力を新人より少し設定し、総入電数の3割はチャットボットが対応すると仮定した上でシミュレーションをし、新人を何人削減できるかを検証する方法が挙げられる。また、チャットボットは**出勤上限を設定する必要**や**休日を与える必要**もなく**一度にたくさんのお客様を対応**することができる。そのため、オペレーターへの制約が緩くなり、オペレーターの希望が取り入れやすくなる可能性が高い。しかし、難しい対応をチャットボットはできないため、**ベテランへの負担が増える**と予想される。上記の懸念を考慮しつつ、**チャットボットがどのようにオペレーターやSVに対して支援をすることができるのか**を研究することが今後の研究計画である。

参考文献

[1]IT用語辞典e-words

http://e-words.jp/w/コールセンター.html

(最終閲覧日 2019/10/24)

[2]コールセンターの課題は人手不足と人材育成、解決の一助はAI？

https://www.zendesk.co.jp/blog/problem-of-call-center-jp/

(最終閲覧日 2019/10/24)

[3]コールセンター業界は人手不足？背景にある問題とは？

https://denwadaikou.jp/column/callcenter/000122

(最終閲覧日 2019/10/24)

[4]「コールセンター白書」著者：月刊コンピューターテレフォニー編集部　出版社：リックテレコム(2011)

[5]「ナーススケジューリング：問題把握とモデリング」著者：池上敦子　出版社：近代科学社(2018)

定式化

集合

・D={1,2,…,d}:日の集合

・D={1,2,…,d2}:日の集合

・S_1={1,2,…,s_1}:ベテランの集合

・S_2={1,2,…,s_2}:新人の集合

・D_1_1={d_1_1}:ベテランの希望休の集合(1日目)

…

・D_1_31={d_1_31}:ベテランの希望休の集合(31日目)

・D_2_1={d_2_1}:新人の希望休の集合(1日目)

…

・D_2_31={d_2_31}:新人の希望休の集合(31日目)

・SEVEN={0,7,14,21,28}:7の倍数の集合

・G={4,5}:禁止ペアの集合

定数

・E_E:ベテラン_早番の最低出勤人数

・E_L:ベテラン_遅番の最低出勤人数

・R_E_d:新人_早番の最低出勤人数

・R_L_d:新人_遅番の最低出勤人数

・C:連続勤務日数の上限

・E:シフトの間隔の上限

・Sunday:最初の日曜日の日付

・P:違反した際の違反度に付ける重み

変数

・$Shift\_1_{d,s\_1}$:ベテランs_1が日dに早番として出勤する時１、しない時０

・$Shift\_2_{d,s\_1}$:ベテランs_1が日dに遅番として出勤する時１、しない時０

・$Shift\_3_{d,s\_2}$:新人s_2が日dに早番として出勤する時１、しない時０

・$Shift\_4_{d,s\_2}$:新人s_2が日dに遅番として出勤する時１、しない時０

・ $X1_d$、 $X2_d$、 $X3_d$、 $X4_d$:最低出勤人数を上回った分の違反度を代入

・ $X5_{s\_1}$、 $X6_{s\_2}$:ソフト制約①の違反度を代入する

・ $X7_{s\_1}$、 $X8_{s\_2}$:ソフト制約②の違反度を代入する

・$X9_{d,s\_1}$、 $X10_{d,s\_2}$:ソフト制約③の違反度を代入する

・ $X11_{s\_1}$、 $X12_{s\_1}$、 $X13_{s\_2}$、 $X14_{s\_2}$:ソフト制約④の違反度を代入する

・ $X15_{d,s\_1}$、 $X16_{d,s\_2}$:ソフト制約⑤の違反度を代入する

・ $X17_d$:ソフト制約⑥の違反度を代入する

・ $X18_{d,s\_1}$、 $X19_{d,s\_2}$:ソフト制約⑦の違反度を代入する

目的関数

Minimize:

$$\sum_{d\in D} P * X_1 + \sum_{d\in D} P * X_2 + \sum_{d\in D} P * X_3 + \sum_{d\in D} P * X_4 + \sum_{s\_1\in S\_1} X_5$$

$$+ \sum_{s\_2\in S\_2} X_6 + \sum_{s\_1\in S\_1} P * X_7 + \sum_{s\_2\in S\_2} P * X_8 + \sum_{d\in D} \times \sum_{s\_1\in S\_1} X_9 +$$
$$\sum_{d\in D} \times \sum_{s\_2\in S\_2} X_{10}$$

$$+ \sum_{s\_1\in S\_1} X_{11} + \sum_{s\_1\in S\_1} X_{12} + \sum_{s\_2\in S\_2} X_{13} + \sum_{s\_2\in S\_2} X_{14} +$$
$$\sum_{d\in D} \times \sum_{s\_1\in S\_1} X_{15}$$

$$+ \sum_{d\in D} \times \sum_{s\_2\in S\_2} X_{16} + \sum_{d\in D} X_{17} + \sum_{d\in D} \times \sum_{s\_1\in S\_1} X_{18} +$$
$$\sum_{d\in D} \times \sum_{s\_2\in S\_2} X_{19}$$

制約式(必須)

・一日の中で早番と遅番は一回しか担当できない

$Shift\_1_{d,s\_1} + Shift\_2_{d,s\_1} \leq 1$

$Shift\_3_{d,s\_2} + Shift\_4_{d,s\_2} \leq 1$

・週の勤務日数が5日を超えてはならない

$(\sum_{d=1}^{d=7} Shift\_1)+(\sum_{d=1}^{d=7} Shift\_2) \leq 5$、$(\sum_{d=8}^{d=14} Shift\_1)+(\sum_{d=8}^{d=14} Shift\_2) \leq 5$、$(\sum_{d=15}^{d=21} Shift\_1)+(\sum_{d=15}^{d=21} Shift\_2) \leq 5$、
$(\sum_{d=22}^{d=28} Shift\_1)+(\sum_{d=22}^{d=28} Shift\_2) \leq 5$、$(\sum_{d=1}^{d=7} Shift\_3)+(\sum_{d=1}^{d=7} Shift\_4) \leq 5$、$(\sum_{d=8}^{d=14} Shift\_3)+(\sum_{d=8}^{d=14} Shift\_4) \leq 5$、
$(\sum_{d=15}^{d=21} Shift\_3)+(\sum_{d=15}^{d=21} Shift\_4) \leq 5$、$(\sum_{d=22}^{d=28} Shift\_3)+(\sum_{d=22}^{d=28} Shift\_4) \leq 5$

・ベテランー早番の希望休

$Shift\_1_{1,d\_1\_1} = 0 \quad \cdots \quad Shift\_1_{31,d\_1\_31} = 0$

制約式(必須)

・ベテラン**遅**番の希望休

$Shift\_2_{1,d\_1\_1} = 0 \quad \cdots \quad Shift\_2_{31,d\_1\_31} = 0$

・新人ー早番の希望休

$Shift\_3_{1,d\_2\_1} = 0 \quad \cdots \quad Shift\_3_{31,d\_2\_31} = 0$

・新人ー**遅**番の希望休

$Shift\_4_{1,d\_2\_1} = 0 \quad \cdots \quad Shift\_4_{31,d\_2\_31} = 0$

・合計出勤回数の上限の指定

$(\sum_{d\in D} Shift\_1)+(\sum_{d\in D} Shift\_2) \leq 22$

$(\sum_{d\in D} Shift\_3)+(\sum_{d\in D} Shift\_4) \leq 22$

制約式(必須)

・ベテランー早番の最低出勤人数(最低出勤人数を超えた分は違反として変数に代入する)

$\mathrm{E\_E} + X1_d \leq (\sum_{s\_1 \in S\_1} Shift\_1)$

・ベテランー遅番の最低出勤人数(最低出勤人数を超えた分は違反として変数に代入する)

$\mathrm{E\_L} + X2_d \leq (\sum_{s\_1 \in S\_1} Shift\_2)$

・新人-早番の最低出勤人数(最低出勤人数を超えた分は違反として変数に代入する)

$\mathrm{R\_E} + X3_d \leq (\sum_{s\_2 \in S\_2} Shift\_3)$

・新人ー遅番の最低出勤人数(最低出勤人数を超えた分は違反として変数に代入する)

$\mathrm{R\_L} + X4_d \leq (\sum_{s\_2 \in S\_2} Shift\_4)$

制約式(※緩和する際の制約式を記載)

・合計出勤回数の下限の指定

18 - $X5_{s\_1} \leq (\sum_{d \in D} Shift\_1) + (\sum_{d \in D} Shift\_2)$

14 - $X6_{s\_2} \leq (\sum_{d \in D} Shift\_3) + (\sum_{d \in D} Shift\_4)$

・日曜日は出勤3回まで

$(\sum_{seven}^{seven+Sunday} Shift\_1 + \sum_{seven}^{seven+Sunday} Shift\_2)$,(seven+Sunday≦31) $\leq 3 + X7_{s\_1}$

$(\sum_{seven}^{seven+Sunday} Shift\_3 + \sum_{seven}^{seven+Sunday} Shift\_4)$,(seven+Sunday≦31) $\leq 3 + X8_{s\_2}$

・連続勤務日数が5日を下回るようにする

$\sum_{d2=d}^{d+C} Shift\_1 + \sum_{d2=d}^{d+C} Shift\_2 \leq C + X9_{d,s\_1}$,d+C ≦ 31

$\sum_{d2=d}^{d+C} Shift\_3 + \sum_{d2=d}^{d+C} Shift\_4 \leq C + X10_{d,s\_2}$,d+C ≦ 31

制約式(※緩和する際の制約式を記載)

・早番の出勤回数と遅番の出勤回数の調整

$7 - X11_{s\_1} \leq \sum_{d \in D} Shift\_1$

$7 - X12_{s\_1} \leq \sum_{d \in D} Shift\_2$

$5 - X13_{s\_2} \leq \sum_{d \in D} Shift\_3$

$5 - X14_{s\_2} \leq \sum_{d \in D} Shift\_4$

・遅番の次の日に早番は入れないようにする

$Shift\_1_{d+1,s\_1} + Shift\_2_{d,s\_1} \leq 1 + X15_{d,s\_1}$,d ≤ 30

$Shift\_3_{d+1,s\_2} + Shift\_4_{d,s\_2} \leq 1 + X16_{d,s\_2}$,d ≤ 30

制約式(※緩和する際の制約式を記載)

・禁止ペアが一緒にならないようにする

$(\sum_{g\in G} Shift\_3)+(\sum_{g\in G} Shift\_4) \leq 1 + X17_d$

・シフトの間隔が3日以上にならないようにする

$1 - X18_{d,s\_1} \leq \sum_{d2=d}^{d+E-1} Shift\_1 + \sum_{d2=d}^{d+E-1} Shift\_2$(d+E-1 ≤ 31)

$1 - X19_{d,s\_1} \leq \sum_{d2=d}^{d+E-1} Shift\_3 + \sum_{d2=d}^{d+E-1} Shift\_4$(d+E-1 ≤ 31)