

製造業における企業内研修の最適設計

2013SE131 永田勇貴 2014SS005 荒川弘規
2014SS010 郷治若葉 2014SS022 伊藤和哉

指導教員：鈴木敦夫

1 はじめに

本研究では、企業から委託されたスケジューリング問題及び割り当て問題をオペレーションズ・リサーチ (OR) の手法を中心に用いて解決をする。OR とは、ある条件下での目的の最適化を図る方法である。例えば、需要量を満たし供給量を超えないようにするという制約がある中で輸送費の最小化をはかる輸送問題や、商品の在庫管理費とまとめ買いすることによる必要経費を考慮し、在庫を管理する在庫管理問題などが挙げられる。OR は身近なものに非常に多く扱われており、効率化を図るためには OR の手法が用いられるケースも多い。

現在共同研究を行っている企業では講習会の時間割作成やグループ作成に関して全ての作業を手作業で行っている。手作業で行うことにより莫大な手間と時間を要しており、かつ様々な条件の下で行わなければならないので、人間が割り当てを行うことによるミスの発生に悩まされている。この手作業で行われる問題について OR を取り入れることによって、作業時間の大幅な短縮を図ることを目標とする。

昨年度までの研究 [1, 2, 3] では、同企業のスケジューリング問題に取り組んでいた。そのシステムにより、手作業で多くの時間を費やしていた作業を約 80 % 程度短縮することを可能にした。OR を用いることによって、十分な結果が得られたため、今回のスケジューリング問題にはより積極的に OR の手法を取り入れる。

本研究では大きく分けて 3 種類の問題の解決に取り組む。

まず 1 つめが時間割作成問題及び担当講師の割り当て問題である。共同研究を行っている企業では、業務効率向上のため毎年業務に沿った講座が開講されている。講座は現時点で約 60 種類、合計 250 回程度、年度毎に講座が開講されている。講座はある制約の下で配置されることが条件であり、それらの講座毎に講師を配置しなければならない。講師もまた、配置される優先度が定められている。従って複数の講師が講座を担当できる場合、決められた条件の下で講師を配置する必要がある。現在企業ではこれらの作業を手作業で行っており、全ての作業を終わらせることに、複数人の担当者が 1 週間程度かけて行っている。

2 つめは、受講者割り当て問題である。先程述べた通り、企業では多数の講座が開講されている。それに伴って、受講者を講座に割り当てることも必要不可欠となってくる。担当者は受講者の講座の希望順序、各受講者の活動拠点、部署などを考慮して、最適な受講者の割り当てを考える必要がある。企業には現在延べ 500 人以上の受講者が存在してお

り、この人数を各講座に手作業で割り当てるのに要する時間が膨大な時間であることは言うまでもないだろう。また、手作業によって受講者を割り当てるため、得られた割り当て結果は受講者の希望度や、活動拠点などを十分に考慮していない可能性が考えられる。

3 つめとして、相互研鑽会の部署割り当て及びアドバイザー割り当て問題である。企業では年に 1 回テーマが近い部署毎に集まり、業務向上のためグループとなって話し合いの場を設ける。各グループにはそれぞれアドバイザーを配置する必要がある。毎年部署の組み合わせとアドバイザーを配置するにあたって、各担当者が手作業でグループ分けやアドバイザー配置を行っている。

これら 3 種類の問題を、OR の手法を用いて解決を図る。OR の手法を用いることによって時間短縮は勿論のこと、モデル化された問題の最適な解を、手作業で行うよりも正確に導き出すことが可能である。現時点でのシステムの成果として、上記にあげた問題全てを従来の半分以下の時間で解決することが可能となった。しかし、OR を用いて得られた最適解は、現実の問題をモデル化して解いたものである。従って、モデルに取り入れられなかった微妙な塩梅については解に反映されていない。これらのことは最終的に企業に打ち合わせた上で担当者の経験や知識を活用できるような機能をシステムに持たせるようにしたい。

本論文は、各問題に対してはじめにその問題の背景を説明する。次に各問題を解決するための手法を説明し、必要ならば定式化を行う。そして、システムのインターフェースについての説明を行う。インターフェースの作成はそれぞれのシステムに対して各担当者がなるべく使いやすいように作成することを目標とする。最終的に得られた実行結果に対して、考察と今後の課題を考える。

2 講座・講師時間割割り当てシステム

企業内で開講する専門教育講座の時間割を自動で作成し、講師を割り当てる。

2.1 問題背景

共同研究を行っている企業では、現在業務効率向上のために、業務に関する講座が行われている。講座は年度毎に半年間開講されており、講座は現時点で約 60 講座、合計 250 回程度存在する。講座毎に講義時間が決まっており、短い講座で 90 分、長い講座で 480 分の講座もある。また、講座にはそれぞれ担当する講師が必要である。講師は講座毎に 2 人必要であり、講師には講師経験数が属性として与えられている。現在企業はこれらの講座の時間割作成に多大な手間と時間をかけている。企業は時間割を作成する際に、手作

業で行っており、作成に莫大な時間を要している。また、講座にはそれぞれ配置する条件が定められており、人間が手作業で行うことにより、様々なエラーが予想される。これらの作業を OR を用いて解決を図る。そして時間割を作成する時間の短縮、制約を満たしたより正確な講座及び講師の割り当てを目指す。

2.2 昨年度からのシステムの改修点及び今年度から新たに追加されたシステム

昨年度のシステム [3] は、講座を希望者の数に合わせて講座を開講する回数を決め、講座を午前に 1 回、午後には 1 回、講義時間の長い講座は 1 日に 1 回講座を開講するスケジュールとなっていた。そしてその講座毎に講師を配置するというシステムとなっていた。しかし、今年から担当者の希望により、1 日を 15 分単位の時間区切りにし、講義時間に沿って講座を配置するというスケジュールに変更した。従って、昨年システムは使用できなくなってしまった。2.2 では昨年度からのシステムの変更点を項目別に列挙する。

1. 講座名入力画面

図 1 講座名入力

講座を追加するときに分類を選択できるようにすることによって、講座の分類を判別できるようにした。

2. 講座制約のユーザーフォーム画面の変更

図 2 講座制約入力画面の変更

昨年度 [3] は午前午後の指定、連続所要日数の指定があったが、今年度からはそれを取り除き、新たに講座の講義時間の入力する欄を作成した。15 分単位の時間軸に設定してあるので、講義時間は 15 分単位で入力する

必要がある。

3. 講座の割り当て順序の変更

昨年度 [3] は、午前午後の指定、連続所要日数の有無、開始指定あり、という制約があったため、講座を割り当てる際の条件が 12 種類あった。しかし今年度からは午前午後の指定、連続所要日数がなくなったため、

- (1) 講義時間が 4 時間以上かつ、開始指定がある講座
 - (2) 講義時間が 4 時間未満かつ、開始指定がある講座
 - (3) 講義時間が 4 時間以上かつ、開始指定がない講座
 - (4) 講義時間が 4 時間未満かつ、開始指定がない講座
- という講座の割り当て順序になった。

4. 講座を割り当てない時間帯の設定

講座の休講時刻	9:00	9:15	9:30	9:45	10:00	10:15	10:30	10:45	11:00	11:15	11:30	11:45	12:00

図 3 講座が開講されない時間帯の設定

講座毎に講義が入らない時間を設定できるようにした。

講座には、分類が存在し、その分類は受講者も同様に決まっている。受講者は分類毎に、昼の休憩時間が異なる。講座が開講されない時間帯が設定できるようにすることで、その分類の講座が受講者の昼の休憩時間と被らないようにすることが可能となった。

5. 講座を午前・午後 1 回ずつの割り当てから、時間軸に講座を割り当てる

2016	2	日	1	2	3	4
		曜日	月	火	水	木
		08:30				
		08:45				
		09:00				
		09:15				
		09:30	分類A-講座1		分類A-講座24	
		09:45				分類A-講座1
		10:00		分類A-講座1		
		10:15				
		10:30				
		10:45				
		11:00	分類A-講座10		分類A-講座12	
		11:15				

図 4 講座時間割割り当て画面

1 日を 15 分単位に区切って、その時間帯に講座を配置するようなシステムとなっている。講座間には 30 分空けるような設定となっている。また、同じ分類の講座は担当者側の都合により、連続しないようなシステムにした。

2.3 講座・講師割り当て方法

今回、講座・講師を割り当てるにあたって、満たすべき条件は以下の通りである。

講座割り当て

- 講座を 15 分単位で区切られた時間軸に、講義時間に沿った講座を割り当てる
- 同分類の講座は連続しないように割り当てる
- 講座間は 30 分休憩時間として空ける
- 指定された時間には講座を割り当てられないようにする
- 開始終了が指定されている講座、講義時間が長い講座は優先的に割り当てる

講師割り当て

- 講座には 2 人講師が必要である
- 講師には講座毎に講師経験数があり、講師経験数の最小、最大で割り当てを行う
- 講師には担当できる講座の上限が存在する
- 講座は担当できる講師が行う

講座割り当てに関しては、最適な解を求めるものではなく、全ての制約を満たしたうえでの講座のスケジュールを考えるものである。従って講座の割り当てはグリーディー法を用いる。講師割り当ては、講師経験数を最大化(最小化)して、講師の最適な配置を考えるので、定式化を考えるものとする。

2.4 定式化

定式化は、0-1 整数計画問題として解く。目的関数において、講師経験回数が多い講師から割り当てるように設定し、ベテランの講師から講座を担当していく。また、「同じ講師が連続で講座を受け持たない」、「講師のスケジュールによって、割り当てが不可能な場合がある」に関しては、すでにこれらの条件を満たすデータを入力するので、定式化には含まない。定式化を [3] に従って、以下のように行う。

定数

K : 講師の集合

L : 講座の集合

P_{kl} : 講師 k が講座 l を経験した回数 ($k \in K, l \in L$)

S : 担当できる講座の種類数

$$b_{kl} = \begin{cases} 1 & \text{講師 } k \text{ が講座 } l \text{ を担当できる} \\ 0 & \text{講師 } k \text{ が講座 } l \text{ を担当できない} \\ & (k \in K, l \in L) \end{cases}$$

変数

$$y_{kl} = \begin{cases} 1 & \text{講師 } k \text{ が講座 } l \text{ を担当する} \\ 0 & \text{講師 } k \text{ が講座 } l \text{ を担当しない} \\ & (k \in K, l \in L) \end{cases}$$

目的関数

$$Z = \sum_{k \in K} \sum_{l \in L} P_{kl} y_{kl}$$

制約条件

$$\sum_{k \in K} y_{kl} = 2, \quad (l \in L) \quad (1)$$

$$\sum_{l \in L} y_{kl} \leq S, \quad (k \in K) \quad (2)$$

$$y_{kl} \leq b_{kl}, \quad (k \in K, l \in L) \quad (3)$$

$$y_{kl} \in \{0, 1\}, \quad (k \in K, l \in L) \quad (4)$$

定式化の説明

● 目的関数

講師経験回数と変数を掛けた和を最大にすることで、それぞれの講座に対してベテランの講師から割り当てるようにする

● 制約条件

式 (1) 2 人の講師を講座 l に割り当てる

式 (2) 講師 k は最大 S 個の講座を担当できる

式 (3) 担当可能な講師 k が講座 l を行う

式 (4) バイナリ条件

2.5 実行結果

作成したシステムを用いて、最適な時間割作成、及び講師の割り当てを行うことに成功した。このシステムにより、以前まで複数人で莫大な時間を要していた作業を、入力を含めて従来の約 20 % の時間でスケジュールを作成することに成功した。実行結果は全ての制約を満たすものであり、OR を用いることによって、最適な解を導き出すことが出来た。

しかし、OR を用いて得られた解はあくまで問題をモデル化して解いたものであって、担当者が求める最適なスケジュールとは微妙に異なる場合がある。従って、ある程度手作業でスケジュールを調整できるようにシステムも改修した。今後は様々なエラーを想定し、エラーメッセージを追加したり、表示や出力方法を工夫して、担当者が扱いやすいようなシステムを目指す。

3 受講者割り当てシステム

講座・講師時間割割り当てシステムで割り当てられた時間割表(第 2 章)をもとに、OR の手法を用いて、受講者の日程・要望を集計し、自動で時間割表の講座へ割り当てをする。

3.1 問題背景

共同研究を行っている企業では、現在、作成された専門教育時間割表(第2章)を用いて、受講者の割り当てを行っている。受講者の人数は延べ500人を超え、講座開講回数は250回を超えている。

それに加えて、遠方で勤務している受講者はできる限り同日に講座を割り当てることや、ある部署を優先的に割り当てたい講座など、さまざまな条件のもと手作業で割り当てを行うため、今まで1週間を費やして割り当てを行っていた。また、受講者の希望は、受講者自ら時間割と照らし合わせながら記入するため、ヒューマンエラーも起こりやすい状態であった。

3.2 条件

まずは受講者の割り当てを行う際に必要な条件を以下に述べる。

- 講座を優先して受講させたい部署があるため、その部署の受講者を優先的に割り当てる
- 遠方の拠点から出張で来る受講者は、出来る限り同日開講の講座に割り当てる
- なるべく多くの受講者が希望順位の高い日程で受講できるように割り当てる
- 希望を出されている日程以外の同講座には割り当てないようにする
- 各受講者が同講座を受講する回数は1回までである
- 講座毎に定員数が決まっているので超えないように割り当てる

必要な条件の上3項目については「重み」により項目内での差をつける。1つめの条件であれば「部署」を、2つめの条件は「拠点」を、3つめの条件は「希望順位」を項目とする。

例えば、愛知・岐阜・三重・静岡・東京の拠点があつたとし、東京を1番目に、静岡を2番目に優先したいとする。東京の受講者に5を、静岡の受講者に3を、その他の受講者に0というように優先したいものに「重み」を与える。これを項目毎に行ったものの総和を「受講希望度」と呼ぶことにする。

3.3 定式化

0-1 整数計画問題として問題を解く。定式化を以下のように行う。

定数

J : 受講者の集合

I : 講座の集合

K : 講座 I の開講回数の集合

T_{ik} : 講座 i 第 k 回の定員数 ($i \in I, k \in K$)

M_{ijk} : 講座 i 第 k 回に対する受講者 j の受講希望度 (部

署 + 拠点 + 希望順位) ($i \in I, j \in J, k \in K$)

$$b_{ijk} = \begin{cases} 1 & \text{講座 } i \text{ 第 } k \text{ 回を受講者 } j \text{ が受講希望するとき} \\ 0 & \text{講座 } i \text{ 第 } k \text{ 回を受講者 } j \text{ が受講希望しないとき} \\ & (i \in I, j \in J, k \in K) \end{cases}$$

変数

$$x_{ijk} = \begin{cases} 1 & \text{講座 } i \text{ 第 } k \text{ 回を受講者 } j \text{ が受講可能のとき} \\ 0 & \text{講座 } i \text{ 第 } k \text{ 回を受講者 } j \text{ が受講不可のとき} \\ & (i \in I, j \in J, k \in K) \end{cases}$$

目的関数

$$Z = \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} M_{ijk} x_{ijk}$$

制約条件

$$\sum_{j \in J} x_{ijk} \leq T_{ik}, \quad (i \in I, k \in K) \quad (5)$$

$$\sum_{k \in K} x_{ijk} \leq 1, \quad (i \in I, j \in J) \quad (6)$$

$$x_{ijk} \leq b_{ijk}, \quad (i \in I, j \in J, k \in K) \quad (7)$$

$$x_{ijk} \in \{0, 1\}, \quad (i \in I, j \in J, k \in K) \quad (8)$$

定式化の説明

- 目的関数

各受講者の希望度の総和を最大にするよう割り当てる

- 制約条件

式 (5) 講座 i 第 k 回を受講できる人数は最大 T_{ik}

式 (6) 受講者 j が講座 i を受講できる回数は最大 1

式 (7) 受講者 j が講座 i 第 k 回を受講希望無しなら割り当てない

式 (8) バイナリ条件

システム上での受講者割り当ての流れ

集めた受講希望表を集計 [4] し、拠点を入力したら「重み表」ボタンを押す。

A	B	C	D	E	
【専門教育】集計表					
1	(3) フォルダ内のファイルを集計				
2	集計	D:\卒論2017\4_専門教育時間割作成システム\2_受講者割り当			
3	(4) 割り当て重み表を作成				
4	重み表				
5					
6					
7	※削除 or 変更		削除 or 変更するNoを入力し削除ボタンをクリックして		
8	No.:	<input type="text"/>	削除	変更	
9	※追加する際は追加ボタンをクリックしてください				
10	追加				
11	講座分類				
12	講座名				
13	No.	部署	従業員No.	氏名	拠点
14	1	営業	760942	従業員1	愛知

図5 「重み表」の作成

部署・拠点・希望順位の一覧表が作られる。この時デフォルトで部署・拠点の重みは0、希望順位の重みは第1希望から順番に5、4、3、2、1と付けられている。

部署	分類A	分類B	分類C	拠点	拠点重み	希望	希望重み
営業	0	0	0	愛知	0	第1希望	5
事務	0	0	0	岐阜	0	第2希望	4
開発	0	0	0	三重	0	第3希望	3
広報	0	0	0	東京	0	第4希望	2
情報	0	0	0	静岡	0	第5希望	1
総務	0	0	0				

図6 作成された重み表

優先するものの重みを変更し、「割り当て開始」ボタンを押すと割り当て結果が一覧表となって出る。結果は3.4の実行結果で述べる。

部署	分類A	分類B	分類C	拠点	拠点重み	希望	希望重み
営業	5	0	0	愛知	0	第1希望	5
事務	0	5	0	岐阜	0	第2希望	4
開発	3	0	0	三重	0	第3希望	3
広報	0	0	5	東京	5	第4希望	2
情報	0	0	0	静岡	3	第5希望	1
総務	0	0	0				

図7 「割り当て開始」ボタンをクリック

3.4 実行結果

今までは受講者自らがどの講座がいつ開講しているかを作成された時間割表から探さなければならなかった。しかし、今回作成したシステムを用いることによって、簡単に講座毎の開講日を知ることができるようになった。また、申込表において講座の受講希望をドロップダウンリスト [5] から選択できるようにしたため、受講希望者の負担が減るとともにヒューマンエラーが起きにくくなる。そして、ボタン1つで簡単に全受講者の希望を一覧にでき、ボタン1つでシステムによる割り当てを行うことができるので、従来の受講者割り当てに比べ約80%の時間短縮に繋がった。受講者の割り当ては制約を満たす最適解で割り当てられるが、担当者が受け入れられる割り当てかどうかは優先度を表す重みの付け方によって変わってくる。そのため今後は担当者が求める割り当て結果になる重みの付け方を模索していく。

講座名	氏名	拠点	2016/1/1 (金) 08:00~09:45	2016/1/1 (金) 13:15~14:00	2016/1/4 (月) 08:00~09:45
1	菅原 760842	岐阜県1			
2	菅原 862797	岐阜県2			
3	菅原 892026	岐阜県3			
4	菅原 100144	岐阜県4			
5	菅原 200921	岐阜県5			
6	菅原 871473	岐阜県6			
7	菅原 370343	岐阜県7			
8	菅原 116991	岐阜県8			
9	菅原 136466	岐阜県9			
10	菅原 153111	岐阜県10			
11	菅原 660480	岐阜県11			
12	菅原 697025	岐阜県12			
13	菅原 486287	岐阜県13			
14	菅原 446412	岐阜県14			
15	菅原 648206	岐阜県15			
16	菅原 699297	岐阜県16			
17	菅原 328951	岐阜県17			
18	菅原 081110	岐阜県18			
19	菅原 696880	岐阜県19			
20	菅原 258976	岐阜県20			
21	菅原 439176	岐阜県21			
22	菅原 661212	岐阜県22			
23	菅原 084046	岐阜県23			

図8 割り当て結果

4 相互研鑽会の部署・アドバイザー割り当て

4.1 問題背景

相互研鑽会とは、部署毎に業務上の問題点や課題を持ち寄り、年に1度行われる、部署同士高め合うディスカッションのことである。似たような「テーマ」を持つ部署を集め、他部署の意見を加味することで、より柔軟に幅広い見解を業務に取り入れ、活用していくことでより良い職場の環境作りや業務の効率化など、職場の改善活動に繋げることが目的となっている。

アドバイザーとは、相互研鑽会を行う上で、部署同士を取り繋ぐ役割を担う担当者のことである。相互研鑽会を行う部署に向けた手法の提案などを行っている。アドバイザーには、A~Cのクラスが存在する。新人はクラスCからスタートし、経験と実績からB、Aへクラスが昇格していく。また、ベテランであるクラスAのアドバイザーは新人であるクラスCのアドバイザーへ指導を相互研鑽会内で行う。

共同研究を行っている企業では、担当者が相互研鑽会の部署、アドバイザーの割り当てを手作業で行っている。しかし、割り当てられた部署の業種が大きく異なっている、経験年数が浅いアドバイザーのみが割り当てられてしまうなどの割り当て方をすると、研鑽会の主旨から反れてしまい、相互不理解を招き、研鑽会そのものが意味をなさなくなってしまう。そういった問題が発生しないように、研鑽会を行う部署同士のテーマが類似した内容になっていることやアドバイザーの専門、アドバイザーの経験年数などに配慮しながら割り当て作業を行っているため、莫大な時間と手間が費やされている。

そこで、本研究では、この相互研鑽会組み合わせ表を作成するにあたり、部署、アドバイザー割り当て時間短縮のために、担当者が割り当てに必要な最低限の情報を入力することで、自動で割り当てをするシステムを考えた。具体的な内容は、入力された情報を基に、部署、アドバイザー、テーマそれぞれのリストを自動作成する。その後、そのリストのワード毎に重みをつけて、数値に変換し、最適な割り当てを行う。

4.2 条件

今回のシステムで使用したORの手法から述べる。計算を2段階に分け、相互研鑽会組み合わせ表を作成した。ス

ステップ1として、各部署の専門分野と拠点、テーマ名を入力する。

さらに、そのテーマに付随する3つのキーワード（キーワード1、キーワード2、キーワード3）を担当者が決める。そして、相互研鑽会を行う部署、グループの上限、下限を定める。それらに基づき、同グループ内に存在する部署の5次元座標のユークリッド距離（専門分野、拠点、キーワード1、キーワード2、キーワード3）の総和を非線形0-1計画法で測ることで、総和が最小になるように部署の割り当てを行う。

ステップ2として、アドバイザーを、グリーディー法を用いて割り当てた。

割り当てを行う際に考慮する条件を以下に述べる。
部署割り当て

- 専門分野が近いテーマの部署を同じグループに割り当てる
- できる限り、拠点が近い部署を割り当てる
- 同グループに存在する部署数は上限と下限に基づく
- 専門分野、拠点、キーワード（3つ）のそれぞれに任意で重みを付けることをできるようにする

アドバイザー割り当て

- クラスCのアドバイザーは最優先で割り当てる
- クラスCのアドバイザーが担当する場合、必ずクラスAのアドバイザーを1人指導役として配属しなければならない
- クラスB、Cのアドバイザーのみの組み合わせは存在してはならない
- クラスAのアドバイザーは同グループに2人以上存在してはならない
- 各アドバイザーは、担当可能な専門分野、拠点が存在し、担当可能でなければ配置してはならない
- クラスB、Cのアドバイザーは、同グループに2人以上配置してはならない
- クラスAのアドバイザーの中には、その年は担当できないアドバイザーが存在する
- クラスAのアドバイザーの中には、複数のグループが担当できるアドバイザーが存在する

これらを、考慮したグリーディー法のプログラムを作成し、アドバイザーの配置を行った。アドバイザーの配置をグリーディー法で計算する際の優先順位の付け方を以下に述べる。

- (1) クラスCのアドバイザーで、経験年数が浅く、担当できるグループが少ない
- (2) クラスCのアドバイザーで、担当できるグループが

少ない

- (3) クラスCのアドバイザーで、担当できるグループが多い
- (4) クラスBのアドバイザーで、経験年数が浅く、担当できるグループが少ない
- (5) クラスBのアドバイザーで、担当できるグループが少ない
- (6) クラスBのアドバイザーで、担当できるグループが多い
- (7) クラスAのアドバイザーで、経験年数が浅く、担当できるグループが少ない
- (8) クラスAのアドバイザーで、担当できるグループが少ない
- (9) クラスAのアドバイザーで、担当できるグループが多い
- (10) クラスAのアドバイザーで、担当できるグループが多く、複数のグループが担当可能

(1)~(10)の順番で割り振ることにより、アドバイザーの配置を完成させ、各グループの担当を決めた。

4.3 定式化

部署が掲げているテーマ内容が、可能な限り近い部署が同グループに集まるように最適化の計算を行う。そのため、専門分野、拠点、キーワード（3つ）のそれぞれに任意で重みを付けることをできるようにした。したがって、似たテーマの部署が近くの座標に集まりやすくなる。よって、目的関数はユークリッド距離の総和を最小にすることである。上記で述べた条件を数式で表現すると次のようになる。

定数

I : テーマの集合

J : 研鑽会を行うグループの集合

$R_{ii'}$: テーマ*i*からテーマ*i'*までの座標距離 ($i, i' \in I$)

M : 同グループに割り当てることができる最多のテーマ数

N : 同グループに割り当てて必要がある最少のテーマ数

変数

$$z_{ij} = \begin{cases} 1 & \text{テーマ } i \text{ がグループ } j \text{ で研鑽会に参加する} \\ 0 & \text{テーマ } i \text{ がグループ } j \text{ で研鑽会に参加しない} \end{cases} \quad (i \in I, j \in J)$$

目的関数

$$Z = \sum_{i, i' \in I} \sum_{j \in J} R_{ii'} z_{ij} z_{i'j}$$

制約条件

$$\sum_{j \in J} z_{ij} = 1, \quad (i \in I) \quad (9)$$

$$N \leq \sum_{i \in I} z_{ij} \leq M, \quad (j \in J) \quad (10)$$

$$z_{ij} \in \{0, 1\}, \quad (i \in I, j \in J) \quad (11)$$

定式化の説明

- 目的関数
グループ内テーマ毎のユークリッド距離の総和の最小を求めることで、似たテーマが同じグループに割り当てられるようにする
- 制約条件
式 (9) テーマ i が割り当てられるグループは 1 つのみ
式 (10) 各グループに割り当てられるテーマ数は N 以上 M 以下
式 (11) バイナリ条件

システム上での部署とアドバイザーの割り当ての流れ

入力画面で相互研鑽会へ参加する部署、テーマ名、専門分野、拠点、3つのキーワードをそれぞれ担当者が入力する。その後、相互研鑽会に参加するグループ数、1グループに入ることができるテーマ数の上限下限、キーワード毎の重みを設定する。

図9 部署・テーマ情報入力画面

図10 部署・テーマ情報設定画面

図11 部署・テーマ情報設定画面2

図12 重みの設定

テーマ一覧表では、入力画面で入力した情報の確認と訂正を行う。その後、「定式化」を行い、テーマをグループ毎に割り当てする。

削除	部署43	テーマ12	開発	岐阜	なし	改善	方針
削除	部署44	テーマ13	開発	岐阜	なし	改善	方針
削除	部署53	テーマ14	事務	岐阜	なし	改善	日常
削除	部署54	テーマ15	事務	東京	その他	福利厚生	その他

図13 テーマ情報一覧

最後に、割り当て開始を行うことで部署の割り当てが行われる。

アドバイザー一覧表では、アドバイザーの割り当てる為の個人情報(部署、従業員ナンバー、氏名、役職・資格、アドバイザー認定年度、経験年数)を入力し、割り当てたグループ毎にアドバイザーを条件に沿うように割り当てる。

生産部	141	従業員18			2015		2
生産部	147	従業員19			2015		1
生産部	153	従業員20			2015		2
開発部	159	従業員21			2012		4

図14 アドバイザー個人情報入力画面

アドバイザー割り当てに必要な情報(担当グループ数、クラス、配置(希望)可能拠点)を入力する。

生産部	141	従業員18			2015		2		
生産部	147	従業員19			2015		1		
生産部	153	従業員20			2015		2		
開発部	159	従業員21			2012		4	○	

図15 アドバイザー個人情報設定画面

入力後、「部署割り当てとリンク」、「部署毎の割り当て可能不可能表作成」、「割り当て開始」の順にボタンを押す。

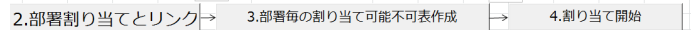


図16 アドバイザー割り当て実行手順

4.4 実行結果

先述のシステムを用いて、部署、及びアドバイザーの割り当てを行うことに成功した。このシステムにより、従来の約50%の時間で研鑽会のための割り当て表を作成することに成功した。実行結果は全ての制約を満たした解を導き出した。しかし、非線形0-1計画法によって導き出された解

であり、局所最適解であるため、最適解であるとは限らない。したがって、手作業による調整を可能にするべく、システムを作成した。今後はヒューマンエラーを想定し、エラーメッセージを追加し、表示や出力方法をユーザー目線で考え、担当者が扱いやすいシステムを目指す。

Gr	部署	在籍拠点	相互研鑽会実施方法	テーマ名	問題解決アドバイザー		アドバイザー	
					部署	氏名	部署	氏名
1	部署23	栗知		開発		開発部	従業員21	
	部署24	栗知		開発		開発部	従業員22	
	部署59	栗知		開発				
	部署60	栗知		開発				
	部署13	岐阜		事務		SS部	従業員19	
2	部署35	岐阜		生徒		開発部	従業員14	
	部署61	東京		開発		A	従業員45	
3	部署1	三重		事務		生産部	従業員20	
	部署2	三重		事務		生産部	従業員17	
	部署33	三重		事務		A	従業員40	
	部署66	三重		事務				

図 17 割り当て結果

5 おわりに

講座・講師時間割割り当てにおいて、共同研究を行った企業ではスケジュール作成を手作業で行っており、多くの時間を費やしていた。それを昨年度の先行研究 [3] で作成されたシステムによって、スピーディーかつ正確に制約を満たす時間割の作成に成功した。しかし、今年度は時間割作成方法が変わってしまったため昨年度のシステムでは対応できなくなってしまった。本研究では応用が効くように昨年度のシステムを改修した。具体的には午前・午後で講座を割り当てていたものを 15 分単位で割り当てられるようにした。この様に形式が変わっても柔軟に使用できるようにするために、今後はさらに以下の点を改善していかねばならない。

- 講師の日程、使用できる会場を考慮したうえで講座を割り当てる
- 複数講座の同時開講を考慮する

一方本年度から新たに研究に加わった受講者の割り当て、相互研鑽会の部署・アドバイザー割り当てにおいては、自動で制約を満たした最適な割り当てができるので、作業時間の大幅な短縮に繋がった。さらに担当者が受け入れられる割り当てにするために企業側の経験的要望を「重み」として付ける必要がある。今後はその「重み」の付け方を模索していくとともに、汎用性があるシステムにしていく。

今回、私達のグループは企業と共同研究を行った。以下に、私達が共同研究を行う中で学んだことを 3 つ挙げる。

1. コミュニケーション

1 番大切だと実感したのはシステムを作る上で、企業の方々や、グループ内でのメンバーとのコミュニケーションであった。企業の方とは、3 週間に 1 回しか会って打ち合わせをすることが出来ない。その中でニーズを聞き出し、自分達が行った研究の成果をわかりやすく伝えなければならない。限られた時間の中で、伝えたいことを伝え、聞きたいことを聞く難しさを身をもって感じた。

また、グループ内でのコミュニケーションは、システムを分担して作成する上でのとても大切なことであった。限られた時間内でシステムを作成するうえで、情報の共有を少しでも怠ると僅かな認識のずれが生じることがあった。その経験から報告・連絡・相談は必要不可欠であることを知った。そのことから、情報の共有を密にしていくことが研究を円滑に進めていくうえで大切であることを学んだ。

2. スケジュール管理

4 人で研究を行っているため、急に集まらなくてはいけない時に全員が揃わず、順調に進まない時があった。研究が予定通りに進まないことも、想定してスケジュールを決めることも大切だと感じた。

3. システムを使用者の立場になって考える

進めていくうちにニーズから外れていないか確認することが大切だと感じた。システムの機能が多いからといって、相手が求めているシステムだとは限らない。最低限の機能があり、単純で扱いやすいシステムを求めていることもある。相手の目線になり、限られた時間の中で、ニーズを察し応えることが大切だと感じた。

参考文献

- [1] 葛岡季絵, 寺本剛:『企業講習会日程の自動作成システムの試作』, 南山大学情報理工学部情報システム数理学科, 2014 年度卒業論文, 2015.
- [2] 川口祐貴, 松崎佳人, 西尾諭:『企業内セミナースケジュールリングシステムの試作』, 南山大学情報理工学部情報システム数理学科, 2015 年度卒業論文, 2016.
- [3] 長崎有純, 小川智也, 大谷拓也, 佐藤優里香:『企業内研修に関するスケジュールリング問題の実際の解法』, 南山大学情報理工学部情報システム数理学科, 2016 年度卒業論文, 2017.
- [4] 井ノ上陽一:『EX-IT「ひとりしごと」の時間とお金:複数ファイル(ブック)の数値をまとめる Excel マクロ』, [https://www.ex-it-blog.com/Excel-macro-book\(2017年10月参照\)](https://www.ex-it-blog.com/Excel-macro-book(2017年10月参照)).
- [5] 伊藤 潔人:『インストラクターのネタ帳:VBA でドロップダウンリストを作成する』, [http://www.relief.jp/docs/excel-vba-making-data-validation-drop-down-list.html\(2017年10月参照\)](http://www.relief.jp/docs/excel-vba-making-data-validation-drop-down-list.html(2017年10月参照)).