

# 卒業論文

## 不確実性を考慮した 対話型ファジィクリティカルパスによる 学習支援システムの開発

Development of a Learning Support System Using  
an Interactive Fuzzy-Critical Path Considering Uncertainty

富山県立大学大学 工学部 情報システム工学科

2120020 島崎圭介

指導教員 António Oliveira Nzinga René 講師

提出年月: 令和7年(2025年)2月



# 目次

図一覧	ii
表一覧	iii
記号一覧	iv
第1章 はじめに	0
§ 1.1 本研究の背景	0
§ 1.2 本研究の目的	1
§ 1.3 本論文の概要	2
第2章 教材の提供と科目推薦	3
§ 2.1 e ラーニング教材と自発的能動学修の涵養	3
§ 2.2 教材管理と学習進度管理の手法	5
§ 2.3 内容系統を考慮した学習管理システムの提案	8
第3章 学習進度のための日程計画	12
§ 3.1 クリティカルパスによる日程計画	12
§ 3.2 クリティカルパスの解法	14
§ 3.3 学習計画のためのクリティカルパスの活用	17
第4章 提案手法	21
§ 4.1 学習管理システムへのファジィの導入	21
§ 4.2 ファジィクリティカルパスによる解法	23
§ 4.3 対話型システム	25
第5章 実験結果並びに考察	29
§ 5.1 実験の概要	29
§ 5.2 実験結果と考察	31
第6章 おわりに	34
謝辞	35
参考文献	36

# 図一覧

2.1	端末利用状況 [6]	4
2.2	通塾と経済状況の関係 [7]	4
2.3	子供の成績の傾向 [11]	5
2.4	高校新課程 [12]	5
2.5	学習系統図 [16]	7
2.6	年間指導計画 [17]	7
2.7	スタディ・ログ [21]	9
2.8	積み上げ型教科 [23]	9
2.9	廻り学習 [25]	11
2.10	すらら [26]	11
3.1	PERT 図 [30]	13
3.2	CPM [29]	13
3.3	PERT 図	16
3.4	CPM	16
3.5	スケジュール作成方法	18
4.1	ファジィ集合の例 [42]	22
4.2	アルファカットの図	22
4.3	線形計画問題を解いた解	24
4.4	スケジューリング法を用いた解	24
4.5	クリティカルパス [39]	25
4.6	対話型 AI [40]	27
4.7	選択枝の例	27
5.1	新規登録, ログイン	31
5.2	勉強時間入力	31
5.3	対話型選択	31
5.4	学習日程	31

# 表一覧

3.1	CPM 関連データ . . . . .	18
3.2	スケジュールデータ . . . . .	19
3.3	圧縮率データ . . . . .	19
4.1	制約条件 [39] . . . . .	25
5.1	システムの評価基準 . . . . .	30
5.2	アンケート結果 . . . . .	33

# 記号一覧

以下に本論文において用いられる用語と記号の対応表を示す.

用語	記号
クリティカルパスにおけるタスク $i$ の開始時間	$S_i$
クリティカルパスにおけるタスク $j$ の開始時間	$S_j$
クリティカルパスにおけるタスクの所要時間	$D_i$
クリティカルパスにおけるタスクの最早終了時刻	$EF_i$
クリティカルパスにおけるタスクの最早開始時刻	$ES_i$
プロジェクト全体の完了時間	$P$
タスクの依存関係を表す集合	$E$
全タスクの集合	$V$
単元	$unit_n$
単元の理想勉強時間	$Hours(unit_n)$
学習予定時間	$study$
予定学習合計時間	Total_Study_Hours
理想合計学習時間	Ideal_Study_Hours
圧縮率	$Press_k$
レビュー $l_i$ の評価値スコア	$R\_score(l_i)$
レビュー $l_i$ の信頼性スコア	$K\_score(l_i)$
レビュー $l_i$ のサクラ性スコア	$F\_score(l_i)$
タスク $i$ から $j$ への所要時間とコスト	$T_{ij}$
問題に応じた重み付けしたスケール係数	$U$
目的関数において $x_{ij}$ の影響を調整するパラメータ	$\alpha$
ノード $i$ からノード $j$ へ移動するバイナリ変数	$x_{ij}$
各タスク $i$ と $j$ に対応する完了時間と開始時間	$y_i, y_j$

## はじめに

### § 1.1 本研究の背景

現在、少子高齢化による労働者人口の減少は1つの社会的課題となっている。人手不足による国内総生産 (Gross Domestic Product : GDP) の減少などの問題が考えられる。この問題の対策として、AI の導入や出生率の増加政策などがあげられる。その対策の1つである「人員・費用などの資源の最適な分配による生産性の向上」について着目した。また、生まれ育った環境や、両親の所得などで子どもが獲得する学力に差がつくことを学力差別と呼ぶ。その学力差別はなくさないといけない。そのために、予算のかからない全ての人が平等に使用できるシステムを提供しなければならない。そのためどの場所においても平等にシステムを使用できるとしてオンライン環境を活用する [1]。

本研究では、人手不足により様々な業界で作業効率や生産性の向上が着目されているなか、教育関係に焦点を当てる。教育関係といっても多くのサービスなどが存在する。その中でも小中高生を対象とした全ての人が使いやすい学習支援システムを提供する。

この学習支援システムでは遡り学習を取り入れている。遡り学習とはつまりいた単元に遡って学習を行うことであり、これにより効率的に理解したい単元について学習を行うことができる。また、スケジュール管理がまだ定着していない学生に対して誰でも自分通りのスケジュールを作成することができる通信教育サービスを提供する。また、対話型にすることによって相手によってどのような頻度で勉強行うかによって余裕を持ったスケジュールやもう時間のない学生に向けてのスケジュールなどの様々なケースに対応することができる形となっている。対話型になっているため自分に合った形で選択することができる。

本システムの導入により、従来の学習支援サービスと比べて以下のようなメリットが期待できる。第一に、低コストで提供可能なため、経済的な理由で塾や家庭教師を利用できない子どもでも、質の高い教育を受けることができる。これにより、学力差の縮小が期待される [2]。第二に、対話型の学習支援を取り入れることで、一人ひとりの学習スタイルに適応できる点が強みである。従来の一方的なオンライン教材とは異なり、学習者の進捗や理解度に応じたカスタマイズが可能となり、より効率的な学習が実現できる。

さらに、ファジィ変数を活用したスケジュール管理機能により、学習計画の柔軟性が向上し、計画的な学習が苦手な子どもでも無理なく学習を進められる。また、スクレイピング技術を活用した教材の提供により、最新の学習資料を継続的に取り入れることが可能となる。このように、経済的・学習環境の両面から全ての子どもに平等な学習機会を提供し、教育格差の是正に貢献するシステムを提案する。

## § 1.2 本研究の目的

既存の通信教育サービスでは、遡り学習のシステムや苦手科目の補強教材が提供されているが、スケジュールや学習単元の選択に関しては、利用者が自ら選定するか、企業があらかじめ設定した一定のスケジュールを利用することが一般的である。しかし、この方法では学習者個々の進捗や理解度に即した調整が難しく、学習計画の柔軟性に欠けるという課題がある。また、GDP の低下が見られる中、学習能力の向上によってよりよい人材ができることを期待できる。

特に、子どもの計画性は発達途中であり、時間の見通しが立てづらいため、自分が何ほどの程度進められるのかを把握していないケースは少なくない。自力でスケジュールを作成すると、過剰な学習範囲を設定してしまうことがあり、このような無理な計画は達成困難であるため、学習意欲の低下や計画倒れの原因となる。

また、目標設定が適切にできない場合、期間内に余裕をもって学習を進めることができず、優先順位を見極めることも困難になる。こうした課題を解決するために、本システムでは利用者がシステムの補助を受けながら、目標期日までに学習すべき単元をカバーできる適切なスケジュールを構築できる環境を提供する。

学習計画は、現実には完全には遂行できないことが多く、計画通りに進まなかった際にモチベーションが低下するケースも見られる [3]。こうした事態を防ぐため、本システムでは、スケジュールの柔軟性を確保し、学習進捗に応じた適応的な調整を可能にする。勉強スケジュールを立てた人が予定を守れなかった場合に即座に次善策を提示し、小さな失敗や遅れにもストレスなく対応できるようにすることで、学習への意欲を維持しやすくなる。この柔軟な学習計画の実現には、不確実性を考慮する必要があるため、そのためにファジィ変数を導入し、どの学習者の状況に応じた柔軟でかつ適切な学習スケジュールを提案するシステムを構築する。

本システムでは、遡り学習を組み込むことで、利用者が自分自身のペースで学習を進められるようにする。これにより、既習内容の理解を深めつつ、新しい単元にも積極的に取り組みられるようになり、学習の効率をあげることもでき、学習の楽しさを実感しながら知識を積み上げることができる。また、システムが自動で不確実性を考慮したスケジュールを提案することにより、自己管理能力の向上にも寄与し、ゆとりを持ったスケジュールや、時間のない時のスケジュールの組み立て方を学ぶことができ、長期的な学習習慣の定着を促す。

教育の発展には、学力差別の解消が不可欠である。経済的な理由や環境による学習格差をなくすため、本システムは低コストで誰もが利用できる仕組みを提供し、教育の機会を公平にすることを目指す。このように、本システムの導入により、個々の学習を最適化するだけでなく、社会全体の教育の質を向上させることにも貢献できると考えている。

また、柔軟性のあるスケジュールは、親の働き方や家庭の状況に合わせて学習ができるため、親のサポートが得られない状況でも子供たちが学び続ける機会を提供できる。

総じて、システムの導入は、子供たちが公平なスタートラインから学べる環境を整える一助となれる。



## § 1.3 本論文の概要

本論文は次のように構成される。

**第1章** 本研究の背景と目的について説明する。背景では，社会問題になっている教育格差の現状と悪影響について述べた。

目的は，学習者がいつでもスケジュールを変更できる柔軟で適応性があり，全ての利用者に平等な学習機会を提供できるシステムを構築することを述べた。

**第2章** 効率的な学習支援システムについて述べる。

予定作成における手法と学習支援システムについて説明する。

**第3章** クリティカルパスにおける日程計画について説明する。

学習計画のためのクリティカルパスについて説明する。

**第4章** 本研究の提案手法について述べる。ファジィクリティカルパスにおける解法を二つ紹介する。データの定式化と定式化したデータをもとに作成されたデータとその作成方法，実際のシステムのUIについて解説する。

**第5章** 本研究の実験および結果を記載する。

実験はアンケートで行い，その結果をもとにした考察を記述する。

**第6章** 本研究のまとめと今後の課題を記述する。



# 教材の提供と科目推薦

## § 2.1 eラーニング教材と自発的能動学修の涵養

2020年からの新型コロナウイルス感染症の影響により、義務教育においても対面での授業を控え、オンライン講義で対応してきている。その際にeラーニングという言葉が使われるようになっていく。eラーニングは1950年に登場したComputer Aided Instructionから発展してきたものである。コンピュータを利用して各生徒の理解度に応じた学習の内容を状況に合わせて提示するシステムであり、米国を中心に世界各国で研究、開発が盛んに行われていた。

1995年、Windows95が発売され、一般家庭にもパソコンが普及するようになり、CD-ROMを中心とした学習Computer Based Trainingが作られた。2000年ごろには政府がe-Japan構想を打ち出したことで紙などの旧メディアを電子化していくことに注目が集まり、eラーニングという言葉が日本国内で登場した。

インターネットのブロードバンド化によって、従来とは比べものにならない高速・大容量通信のインターネット通信ができるようになり、Webを利用した学習をWeb Based Trainingと呼び、学習履歴はもちろん、教材やプログラムまでもサーバ上で一括管理ができるようになった。CD-ROMでは教材改定やプログラム改定が困難だったが、教材・プログラムがWeb上に保存されているため、提供者は受講管理や教材の更新、プログラムのメンテナンスをスピーディに行えるようになり、受講者は場所と時間を選ばず最新の環境・教材で学習できるようになった。そして、現在ではスマートフォンやタブレットといったモバイル端末の登場とともにeラーニングという言葉がより一層の普及を見せるようになる[4],[5]。

従来の学習法と比べて、eラーニングには、時間や場所を選ばずに学習できる、学習者の理解度や進度に合わせて学習内容を調節できる、反復学習や課題学習など、自発的・能動的な学習が可能な教材やコンテンツを活用できるといったメリットが存在する。その結果として、学習者が主体的に学習に取り組む意欲は高まる。

これらの特徴から、eラーニングは自発的能動学習の涵養に有効な学習方法と考えられる。

### GIGA スクール構想

義務教育の場においてもeラーニングは活用されており、2019年には児童生徒向けに端末が配布され、通信ネットワークを学校に配備し、個別最適化された教育を全国の学校現場で実現させることを目標としたGIGAスクール構想が発表された。現実空間と仮想空間が一体となったSociety5.0時代を生きていく子供達にとって、教育におけるICTを基盤とした先端技術の活用は必須だといえる。変化の激しい現代においては、多様な子供達を誰

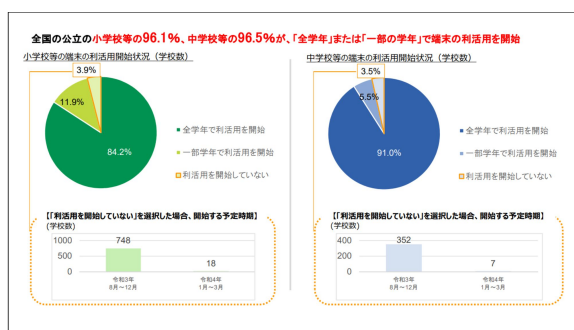


図 2.1: 端末利用状況 [6]

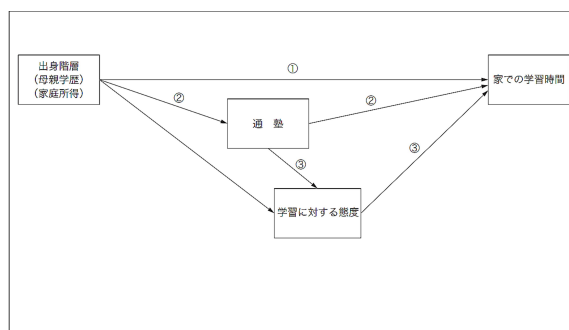


図 2.2: 通塾と経済状況の関係 [7]

一人取り残さず、個別に創造性を育む教育の実現が重要であり、ICT 教育に携わる次世代の人材を育てる必要がある [8].

上記の調査から、GIGA スクール構想の 1 人 1 台の端末の普及は進んでおり、多くの子供たちが e-ラーニングを受講できる環境にあることが分かる。2021 年には全国の公立小学校の 84.2 %、中学校の 91.0 %、全学年で端末の利活用を開始しており、全国の自治体の中でも 96.1 %が端末を整備していることが分かった [6](図 2.1 参照).

## 自発的能動学習

自発的能動学習とは、学習者が自ら学習の目標や目標を設定しながら主体的に学習に取り組む手法である [9]. 現在、社会のグローバル化や、IT 産業の躍進など社会環境の変化は激しく、今使われている知識も急激に陳腐化すると考えられる。このような状況下では、既存の知識が急速に陳腐化し、新たな情報やスキルの取得が求められる。従って、学習者は自らの学びの意欲を高め、自己管理能力を向上させることが必要になる。

自発的能動学習では単に情報や知識を暗記するだけでなく、自分で情報を選別し新しい知識を取り入れながら変化する状況に適応できるスキルや能力が求められる。そのため、協調性や創造性、批判的思考など、将来の職業や社会で必要とされるスキルを養うことができる。今後ますます変化の激しい社会環境に対応するために、自発的能動学習は重要になっていくと考えられる。

また、2021 年のベネッセの教師への調査によると、小学校では 58.7 %、中学校では 44 %が「新課程で指導すべき学習内容が多くて授業で教えきれない」と回答しており、学校で提供されるカリキュラムだけではカバーできない学習内容が増加してきている [10]. 教師が授業で指導する範囲が限定されている中で、個別の生徒のニーズや興味に合わせた学習が十分に提供できない状況が続いている。

日本では出身階層と通塾と家での学習時間がいずれも正の相関が存在する [7](図 2.2 参照). 富裕層の親は通塾を子供に教育メリットを提供する手段として活用している、そのため、通塾が暗黙の了解として教育メカニズムに取り込まれ、富裕層の子供たちは学習機会に恵まれ、学力やスキルを発展させやすい環境にある。

一方で、経済的に困難な層の生徒は通塾の負担を取り入れることが難しく、学校教育のみを受けている場合がある。これにより、学習機会や資源に不均衡が生じ、相対的に学習習慣を身につけづらい状況が生まれている [4].

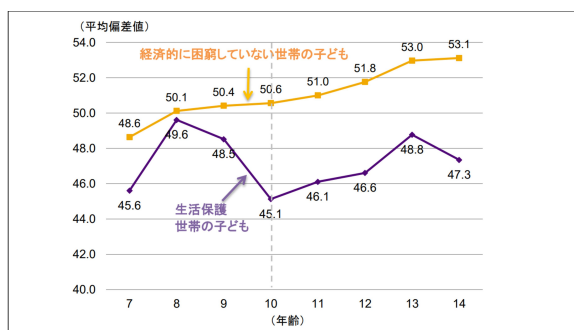


図 2.3: 子供の成績の傾向 [11]

教科	科目	単位数	必修科目	教科	科目	単位数	必修科目
国語	国語総合	4	○*	国語	現代の国語	2	○
	国語総合	3			国語総合	2	○
	現代文A	2			漢文	4	
	古典A	2			国語総合	4	
地理	地理総合	4		地理	現代地理	2	○
	世界史A	4	1科目		現代地理	2	○
	日本史A	4	1科目		現代地理	2	○
	地理A	2			現代地理	2	○
歴史	歴史総合	4		歴史	現代歴史	2	○
	世界史A	4	1科目		現代歴史	2	○
	日本史A	4	1科目		現代歴史	2	○
	歴史A	2			現代歴史	2	○
公民	現代社会	2	必修科目	公民	現代社会	2	○
	倫理	2			現代社会	2	○
	政治・経済	2			現代社会	2	○
	倫理	2			現代社会	2	○
数学	数学Ⅰ	3	○*	数学	数学Ⅰ	3	○*
	数学Ⅱ	4			数学Ⅱ	4	
	数学Ⅲ	5			数学Ⅲ	3	
	数学Ⅳ	2			数学Ⅳ	2	
外国語	外国語総合	2		外国語	外国語総合	2	○*
	英語総合	2			英語総合	2	○*
	英語総合	2			英語総合	2	○*
	英語総合	2			英語総合	2	○*
情報	情報総合	2	○*	情報	情報総合	2	○
	情報総合	2			情報総合	2	○
	情報総合	2			情報総合	2	○
	情報総合	2			情報総合	2	○

図 2.4: 高校新課程 [12]

2017年に行われた日本財団の箕面市子ども成長見守りシステムを用いた調査により、家庭の経済格差と子どもの認知・非認知能力格差との関係が明らかにされた。調査によれば、貧困状態にあると学力が低くなる傾向があり、特に小学校4年生以降で学力が大きく低下することが示された(図2.3参照)。さらに年齢が上がるにつれて、経済的に困難な世帯の平均的な学力や成績逆転のが低下し、低学力が固定化してしまう傾向があることも指摘された。この状況を踏まえ、格差が拡大する前に早期に支援を行うことが重要であるといえる。

また、生活習慣や自制心、勤勉性、外交性、協調性などの非認知能力については、貧困世帯と非困窮世帯を比べると、小学校初期から大きな差が存在している。たとえ貧困状態にあっても学力の高い子どもは、生活習慣や学習習慣などが高水準にあるため、支援には基礎的信頼や生活習慣など、学習の土台となる非認知能力の育成が重要と考えられる [11]。

2022年4月高校1年生から新学習指導要領が始まり、高等教育の課程が大きく変化する。新課程での変化について図2.4に示す。日本の学習指導要領は、時代の変化を見据えて概ね10年ごとに改訂されているが、今の日本社会は、グローバル化の進展や技術革新、生産年齢人口の急減など目まぐるしく変化しており、先を見通すことが難しい時代になっている。

そのため、予測困難な社会で必要となる自ら課題を見つけ、自ら学び、自ら考え、判断して行動できる力を身につけることを目標としている新学習指導要領では知識・技能だけでなく、思考力、判断力、表現力なども含めた力も必要になるため、自発的能動学習の重要性がより高まると考えられる [12]。

## 学習指導要領

小学校、中学校、高校には学習指導要領というものがある。これは、「子供たちが未来社会を切り拓くための資質・能力を一層確実に育成。その際、子供たちに求められる資質・能力とは何かを社会と共有し、連携する「社会に開かれた教育課程」を重視。」となっている。この基本的考えは変わらないがそれぞれの時代にあった単元や指導が用意されている [13]。2025年度大学共通テストでは、情報という科目が追加され数学ではIBCとなり年々変化していることがわかる。

## § 2.2 教材管理と学習進度管理の手法

学習内容を段階的に配置し、順序だてて学習させる指導方式を系統学習とよぶ。系統学習は一定の知識を効率よく学習するには非常に効果的な方法で、テストを利用して評価す

る傾向が強く、学習意欲が高い児童・生徒が多いときには競争心が生まれ相乗効果を期待できる。

デメリットとして、知識重視の一方通行型による学習方法のため、知識重視の一方通行型による学習方法のため、教えられる内容が生徒の興味・関心・必要とは必ずしも合致しない。結果として、学習者が自ら学ぶ意欲や思考力が育ちににくく、学習への意欲も低下してしまう [12] [14]。

一般的に一斉教授型授業の中で展開されるが、生徒の能力差・個人差に対して個別に対応することが難しいために、理解の早い学習者は授業の流れの中で流され、理解の遅い学習者は置いて行かれるということが必然的に生じる。競争から脱落してしまった学習者を救済するようなシステムが構成されずに、学習塾など学校以外の団体に頼りがちになってしまう。

一方で、問題解決学習は学習者が知識だけでなく、問題の解決能力や実践的なスキルも養う自発的な学習を重視する教育手法である。学習者のモチベーションを高く保つことが可能であるが、自発的な学習を強調しすぎると、知識の定着が阻害される可能性があるため、適切なバランスが求められる。

現在の教育課程の基準である学習指導要領は、基本的には系統学習の考え方に基づいて編成されている。そのため、テストは学年で同じ内容で行われ、優秀な結果を出すには学習範囲を網羅しておく必要がある。

自らの興味や得意不得意を考慮して、学習者のモチベーションを高める自発的な学習と最低限のテスト範囲を学習できる学力を確保できる系統学習のバランスをとることが有効になると考えた。

系統学習と自発的な学習を考慮した、学習進度管理に考慮すべき3つの要素を説明する。

## 学習系統図

学習系統図とは、単元を系統立ててつないで図示したものであり、全体を俯瞰で捉えることができ、各単元のつながりを視覚的に理解することができる(図2.5参照)。苦手単元に関連する前提になる単元や後に続く単元も学習することで、理解を深めることができる。教科書会社が教科書や参考書、Webサイトに出している場合がある。

苦手な単元など課題を洗い出すためにも活用でき、どの単元が基礎になり、どの単元が発展した単元なのかを把握することで、苦手単元に重点的に学習時数を割り当てるなど、効果的な学習計画をたてることができる。また、学年と併せて表記することで、いつ何を学ぶかが把握しやすくなり、受験勉強の際の知識の整理にも有効とされる [15]。

ある単元に対して、苦手単元ならば前提単元を見直し学習することで理解を深められ、好きな単元に対しては関連する発展した単元を学ぶことができる。興味を持った分野を深く掘り下げることで、知識やスキルのつながりを理解し、総合的な学びを促進することが期待される。

学習系統図は系統学習をサポートし、自発的な学習の涵養にも役立つと考えられる。系統図を活用することで、学習者は自分の位置を理解し、どの部分で理解が足りないかを把握し、学習者は系統学習を基にして積極的な参加と問題解決に取り組むことが期待される。

## 年間指導計画



年間指導計画とは、学年や学級において、学習の時間の学習活動の見通しをもつために、年間の流れの中に単元を位置付けるものである(図 2.6 参照)。各単元が年度全体の流れの中でどの時期に位置づけられるかが示され、学習の進行が順調に行われ、必要な知識やスキルが段階的に習得されるようになる。

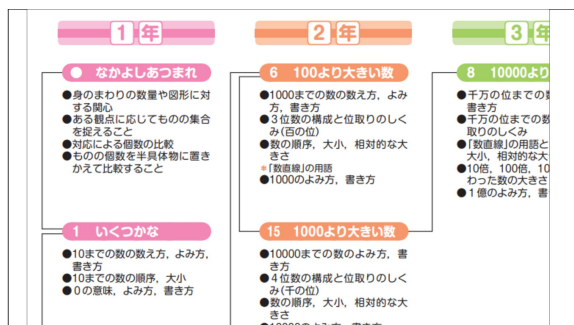


図 2.5: 学習系統図 [16]

図 2.6: 年間指導計画 [17] は、2月(上半期)と7月(下半期)の学習活動を縦横に示すマトリクス形式の図表である。縦軸は学年(1年、2年、3年)で、横軸は単元(1. 1000より大きい数、2. 10000より)で構成されている。各単元には、学習の重点となる事項がリストアップされている。

学年	1. 1000より大きい数	2. 10000より
1年	●1000までの数の数え方、よみ方、書き方 ●3位までの構成と位取りのしくみ(百の位) ●「数直線」の用語と大きさ ●「数直線」の用語 ●1000のよみ方、書き方	●10000までの数の数え方、よみ方、書き方 ●4位までの構成と位取りのしくみ(千の位) ●数の順序、大小、相対的な大きさ ●10000のよみ方、書き方
2年	●10000までの数のよみ方、書き方 ●4位までの構成と位取りのしくみ(千の位) ●数の順序、大小、相対的な大きさ ●10000のよみ方、書き方	●100000までの数のよみ方、書き方 ●5位までの構成と位取りのしくみ(万の位) ●数の順序、大小、相対的な大きさ ●100000のよみ方、書き方
3年		

図 2.6: 年間指導計画 [17]

年間指導計画に含まれる基本的な構成要素としては、単元名、各単元における主な学習活動、活動時期、予定される時数などがある。単元のねらい、生徒の意識、各教科等との関連、外部講師や異校種との関連などを記す場合もある。

時数の適切な配分は学校教育法施行規則に基づき、各単元に十分な時間が割り当てられ、学習内容が十分に網羅されるように計画される。時数のバランスは、生徒が各教科の学習に適切な時間を費やせるように配慮されている [14]。

年間指導計画は教育の効果的な実施をサポートし、生徒が豊かな学びを得るための計画策定に不可欠である。学校の授業の進行や、試験も計画に従って行われるため、学習者が優秀な結果を得るためには年間指導計画で示された範囲は履修しておく必要がある。

## スクレピング

スクレピングとは、データを収集し、かつ目的に合わせて加工することである。特に、Web 上から必要なデータを取得することを、Web スクレピングと呼ばれている。様々なツールやプログラミングでスクレピングを自動化することで、Web データの収集にかかる手間や時間は大幅に削減することが可能である。

スクレピングと似ている意味の言葉にクローリングがある。クローリングとは、Web 上で様々なサイトを巡回し、情報の保存や複製など様々なことを行うことを指す。クローリングとスクレイピングはともに情報を収集する手段であるが、クローリングが巡回に焦点を当てて一方でスクレピングは情報の抽出に焦点を当てている。

また、企業や公共機関は、情報やデータを提供してくれることもあり、その際に使われている仕組みは API と呼ばれている。クローリングやスクレイピングをする前に、必要な情報が API によって提供されているかどうかまず確認することが大切になる。

Web スクレピングに主に用いられるツールとして、Selenium や Puppeteer などがある。ログインやボタンのクリックなどの、マウス操作が必要な Web サイトや、JavaScript で記述されている Web ページのスクレピングするときは Selenium が用いられている。ページのスクリーンショット、PDF 作成、DOM の取得操作などが簡単に行うことができる Puppeteer といって様々な使い分けがされている。

## スタディ・ログ

近年、教育の情報化・デジタル化が進展し、学習者の学びの道筋や成果の多くをスタディ・ログと呼ばれるコンピュータ上のデジタルデータとして蓄積・活用している [18].

これまではデータの蓄積がなかったことから過去を遡って生徒が何を苦手としているかを気づくことができなかったが、蓄積されたスタディ・ログから生徒の理解度を見返すことで苦手範囲が明らかになり、早期に個別指導などの手当てが可能となっている [19]. スタディ・ログを活用することで、従来のクラス全体への一斉指導から、生徒一人ひとりに焦点を当てた個別指導への移行が容易になる。これにより、生徒が最適なペースで学び、個別の課題に集中することができる。

スタディ・ログから得られたデータを学習者自身に可視化することで、自己評価や自己管理のスキルを向上させる効果が期待される。学習者が自分の進捗や弱点を理解し、それに基づいて学習戦略を調整できるような仕組みが重要になる [20].

先行研究では、学習系統図とスタディ・ログを組み合わせ、集計表に入力した誤答数が系統図に表示される仕組みを作成し、視覚的にとらえた (図 2.7 参照).

その結果、クラス全体の誤答数を系統図に表示することで、担任は、それらの誤答数を比較することが可能となり、特に補充指導の必要がある内容を視覚的に捉えることができ、教師がクラス全体のつまずきの傾向を分析する上で有効であった。

また、個人の指導項目ごとのテストの結果を、系統図を通して視覚的に捉えることにより、どの段階からつまずいているかを把握することができる。他の教員や保護者に伝えることで指導を組織的に進め、学習内容の定着を促すことができた。

さらに、小中間での情報の引継において、つまずきや、その指導の内容についての具体的な情報を共有することができ、中学での指導にその情報を有効活用できると考える。指導の結果、児童の誤答数が減少し、つまずきを解消することができた [21].

このように、スタディ・ログと学習系統図の組み合わせの有効性は示されている。

問題解決学習と系統学習は互いに矛盾するものではなく、両者の長所を活かし、適切なバランスを取ることが可能である。学習系統図、年間指導計画、スタディ・ログの三つの要素が相互に補完し合うことで、系統学習と自発的能動学習を効果的に組み合わせることができる。これは、学習者の主体性を引き出し、学習効果を最大限に高めるための理想的な学習環境といえる。

## § 2.3 内容系統を考慮した学習管理システムの提案

教科は、前の単元を土台にし、新しい単元の知識を積み上げなければ学習を理解できない積み上げ型教科と、それぞれの単元に関係性が薄く、ほかの単元の知識をあまり使わない独立型教科に分けることができる。独立型教科の中にもそれぞれの単元の中では積み上げ型の特性があることもある。

例を挙げれば、数学は典型的な積み上げ型教科であり、前の単元で習った公式や定理を理解していないと次の単元を理解できない。英語は語彙や文法知識が積み上げられるため、積み上げ型教科の側面がある。しかし、読解や英作文など、独立型教科の側面もある。国語は、読解や作文など、独立型教科の側面が強い。しかし、漢字や古典など、積み上げ型教科の側面もある。理科や社会は、独立型教科の側面が強く、それぞれ独立した体系を持つ。



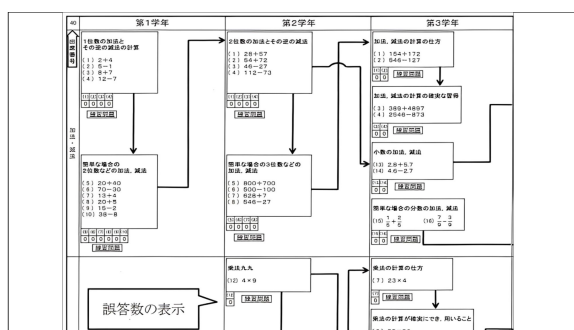


図 2.7: スタディ・ログ [21]

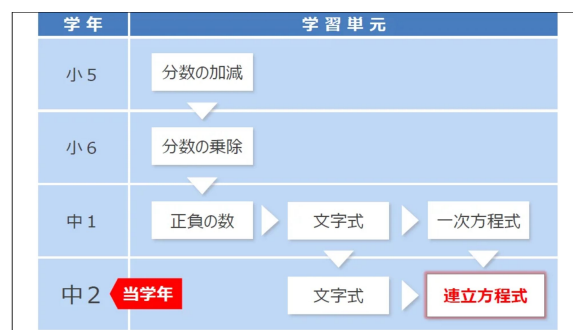


図 2.8: 積み上げ型教科 [23]

ベネッセの調査によると、高校受験を経験した高校1年生の中学生のところに一番苦手だった教科の第1位は数学、第2位は英語であり、どちらも積み上げ型の側面が強い教科であった [22]。独立型教科は単元の結びつきが弱いので、今やっている単元を最初から復習すれば、定期テストでは得点が取れるが、積み上げ型教科では抜け落ちてしまった部分を復習しなければそこから先の単元が理解できなくなってしまう [23]。

図 2.5 に積み上げ科目の例を挙げる。正の数・負の数を基礎として理解を積み上げなければ、文字の式を理解することができない。文字の知識を積み上げて、方程式や平面図形の理解につながる。しかし、これらの基礎単元を完全に理解できていないと、たとえ上の概念を学習しても、理解は分散的なものとなってしまう、安定した学力とはいえない。

中高生でも、小学生で習う割り算や分数などから理解できていないケースもあり、学校や塾などの集団学習のみではつまずきまでは解明できず、置いてきぼりになる生徒が出てしまう。このようなつまずきが解消されないことで、学習意欲が低下してしまうことがあるため、早期発見・早期対応が重要になる。

## 積み上げ式学習

積み上げ式学習とは、全ての基礎となる部分の確認と定着からしっかり行い、それぞれのわからない原因からひとつひとつをじっくり隙間なく単元を積み上げていくことで、安定感のある本物の学力を身につけられる学習方法である。

最初はなかなか先に進まないが、ある一定のラインを超えると急激に理解力が高まり、身につくまでの時間が加速度的に短縮されるため、結果的に近道になると言われる [21]。

しかし、積み上げ式学習を行ったとしても、理解できていない単元は出てしまう。また、積み上げ学習を行っておらず、どこから勉強したら良いか分からない場合も存在する。その際、遡り学習が非常に重要な役割を果たす。

## 遡り学習

学習系統図を使用して、現在理解できていない単元の前提となる単元を遡りながら復習することで一つ一つの単元の理解を積み上げていくことで、安定感のある本物の学力を身につけることができる。応用力や思考力が養われ、どんな問題にも対応できるようになる。この学習方法を遡り学習と呼ぶ [24]。

図 2.9 に遡り学習の例を示す。連立方程式の理解に詰まっているとき、一次方程式→文字式→正負の数と関連した基礎となる単元を復習していく。この繰り返しによって、理解できていない単元を特定することができる。

これまで学んできたすべての科目を復習することは現実的ではなく、かえって混乱することもあるため効率が悪い。スタディログなどによって一人一人の理解度を分析し、既に身についている単元はショートカットすることにより、必要最低限かつ最適な学習を行うことができる。必要な知識を必要なタイミングで復習することで、無駄な学習時間を減らすことができる。

また、学習系統図を用いることで学習目標に対しての達成率を可視化し、自らの成長を実感できるという効果もある。自分の理解度を明確にしておくことで、自学自習にも役立ち、学習に対する意欲が高まる。

遡り学習は、塾や個別指導ではよく使われている手法となっている。目標を明確にして、何のために遡り学習を行うのかを明確にすることで、モチベーションを維持することができる。

## 先取り学習

自分の学年で学ぶことの数ヶ月先の単元を学習したり、1 から 2 年、場合によってはそれ以上、学年を超えた先の内容を学ぶことを先取り学習と呼ぶ。自発的な学習の涵養に役立ち、自らの意思で学習を進めることで学習意欲を高め、主体的に学ぶ姿勢を育むことができる。

特に学力の高い学生にとっては、学校で習っていることがすでにしっかり身についている場合があり、同じ進度で学んでいくのは苦痛に感じることもある。先取り学習は、そのような学生にとってより深い学びを求め、学習意欲を維持するための有効な手段となる。

注意点として、学校で習っていることが不確かなまま先へ進むことは、不安定な学習になりやすく、必ずしも推奨されることではなく、基礎がしっかり身についていない状態で先取り学習を進めてしまうと理解が不十分になり、後々大きな壁にぶつかる可能性がある。また、先取り学習にばかり時間を費やすと、学校の授業がおろそかになってしまう可能性があります。学校の授業は、先取り学習で学ぶ内容の土台となるものであり、学校の授業を軽視してしまうと、先取り学習で学んだ内容も理解が浅くなってしまう可能性がある。

## 無学年学習

無学年学習とは、遡り学習と先取り学習を合わせ、子どもの学年に関係なく一人一人の学力や理解度に合わせて学習を進める学習方法である。従来の学年制では、同じ学年の子供たちは同じ内容を同じペースで学習するが、子供の能力や学習速度はそれぞれ異なり、同じペースで学習を進めることには無理が生じる。子供たちの個性や能力を尊重し、それぞれのペースで学習を進めることができるため、より効果的な学習が可能になる。

メリットとしては、能力や個人に合ったペースで学習を進めることができるため、学習への意欲を高めたり、個性を伸ばしたりできることが期待される。一方デメリットも存在し、子供一人一人に合わせたカリキュラムを作成する必要があるため、管理が難しくなる。しかし、これらのデメリットは、適切なカリキュラムや指導方法を導入することで克服することができると考えられる。

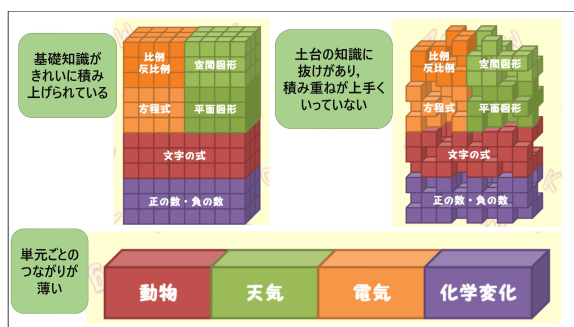


図 2.9: 遊り学習 [25]

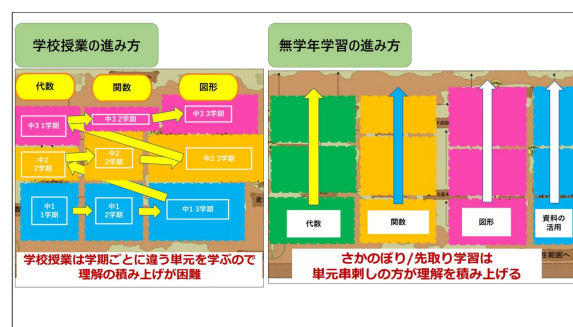


図 2.10: すらら [26]

無学年学習は、子供たちの可能性を最大限に引き出すための学習方法として今後ますます注目されるだろう [26].

単元のつながりを利用している実用例を示す.

- すらら: 無学年式の通信教育システムを考慮した結果、つまずきの元となる単元を推薦し、苦手を克服する手助けを行っている. つまずきの元となる単元を克服した後には、今度は串刺し式に類似する難易度の高い単元に進むシステムを採用しており、前単元の記憶が薄れる前に単元の理解を積み上げられる [26]. 勉強に遅れをとっている子供でも、苦手を完全克服しながら効率的に学習を進めることができる (図 2.10 参照).
- Axis: 個別指導学習システム図を使用し、今までどのような学習をしてきたか、これからどのような学習をしていくのか、各教科の単元ごとの関連性を明確にしながら苦手単元を特定、塾で個人個人に合わせた学習スケジュールを提供している. 学習スケジュールの最適化に使用している [27].

このように、単元ごとの関連を利用した教育方法は実用化されていることが分かる.



# 学習進捗のための日程計画

## § 3.1 クリティカルパスによる日程計画

プロジェクトの進捗管理を行う上で、目標達成のためにタスクやプロセスを可視化する必要がある。特定のタスクが終わらなければ作業を開始できないタスクもあるので、タスク間の関連性を把握しておかないと致命的な遅れが発生することがあるからである。そのために、1950年代に開発されたプロジェクトマネジメント手法であるクリティカルパスメソッド (Critical Path Method : CPM) を使用する。

CPMは、プロジェクト完了のために実行しなければならないクリティカルなタスクを特定する手法であり、Critical Path(クリティカルパス)とは、プロジェクトの全工程を最短時間で完了させるために重要な作業経路のことを指す。プロジェクトの一連の工程を結んだ時、最も時間のかかる最長の経路がクリティカルパスの経路となる [28]。

プロジェクト規模が大きくなると単純にタスクの数が増え、タスク間の依存関係も複雑化するため、遅れが生じるとプロジェクト全体の遅延に繋がるようなタスクであっても把握するのが難しくなってしまう。このような遅延が許されないタスクを把握し、遅れないように対策することができる。

CPMは現代においては自動化ツールによって簡単に作成されるように進化し、プロジェクトプランニングにおいて不可欠な一部となっている [29]。

クリティカルパスを導出するメリットは複数存在する。

### スケジュール作成

まず、スケジュール作成に有用な点があげられる。クリティカルパスがそのままプロジェクト全体にかかる最も時間のかかる最長の経路となるため、作業日数や必要工数も明瞭になる。また、プロジェクト全体の流れとクリティカルパスの把握により、クリティカルパス以外のタスクで納期あるいは工数を調整するなど、効率的なスケジュール管理ができる。

また、プロジェクトの締め切りは、思いがけず前倒しになることがある。そのような突如のスケジュール変更直面したときも、前もってクリティカルパスを把握しておけば短縮できる場所を探し、柔軟に対応することができる。

クリティカルパスがそのままプロジェクト全体にかかる最も時間のかかる最長の経路となるため、作業日数や必要工数も明瞭になる、プロジェクト全体の流れとクリティカルパスの把握により、クリティカルパス以外のタスクで納期あるいは工数を調整するなど、効率的なスケジュール管理ができる。

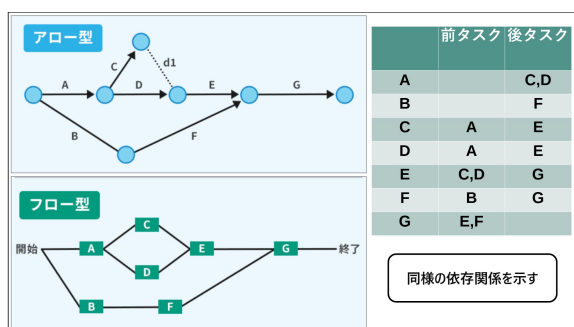


図 3.1: PERT 図 [30]

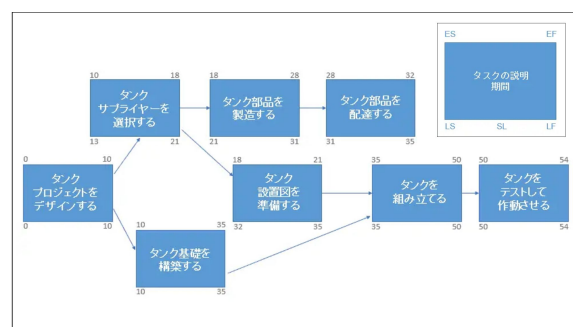


図 3.2: CPM [29]

また、プロジェクト内タスクの依存関係を矢印とノードを用いて図示できるため、並行して行えるタスクや優先すべきタスクを視覚的に分別したスケジュールの作成が可能である。

## PERT 図

プロジェクトのノードを線や矢印で結び、プロジェクトの達成に必要な業務の流れを図表にしたものを PERT 図と呼ぶ。工程と所要日数のほか、工程間の依存関係が視覚的に確認できるため、スケジュール管理や適切なリソース配分をするのに役立つ。PERT 図を活用すると、プロジェクト内で重点を置くべき工程を簡単に把握できるのが特徴であり、主にプロジェクト全体のなかでクリティカルパスを見極めるために使用される。PERT 図はアロー型とフロー型に分けられる [30]。それぞれを図 3.3 に示す。

### ● アロー型

円は、タスクの開始または終了を表し、矢印が作業内容を示しており、アローダイアグラムの形をとる。アローダイアグラムでは、ノードが2つに分かれてしまった場合、ダミー作業を点線で作図し、二股に分かれてしまったノードを1つにまとめる。実際の作業は伴わないため、作業時間の加算はゼロとして扱うことになる [31]。

作業の流れを強調して表示するため、プロジェクトの進行において各作業の関係性が明確になるが、作業の詳細やタスクの内部構造を表示するのが難しく、大規模なプロジェクトの場合には見込みにくいことがある。また、プロジェクトが複雑になると、多くのノードや矢印が交差し、見やすさが損なわれる可能性がある。

### ● フロー型

ノードをタスクや工程として扱い、タスク間の依存関係を示す矢印でつなぎ、ネットワーク図の形をとる。詳細を重視するため、全体像を理解するのにアロー型よりも時間がかかることがあるが、ノードをタスクや工程として扱うため、各タスクの詳細な作業内容をノードに属性づけることができる。また、タスク間の依存関係を明示的に示すことができ、細かな作業順序の理解がしやすいというメリットもある [32]。

本研究の学習系統図はそれほど大規模な依存関係ではない。しかし、提案手法でも説明するが、グラフ化する際にタスクに单元名、单元番号、リンク先を埋め込む必要があるため、タスクに情報を組み込みやすいフロー型を用いる。

## フロート

クリティカルパスと比べたフロートを求めることができる。フロートとは、後続するタスクやプロジェクトの終了日に影響が出ない、タスク遅延の許容範囲のことである。フロートを見極めておくと、プロジェクトの柔軟性を判断するのに役立つ。

プロジェクト内で優先度の高いタスクを把握することで今後のスケジュールを円滑に行うことが可能である。クリティカルパスの作業を早急に終わらせることによって、全体のスケジュールに余裕をもたせることも可能になる [33]。

## ボトルネック

プロジェクトにおけるボトルネックを導出できる。ボトルネックとは、ワークフロー内で停滞や生産性低下など、良くない影響を与えている箇所を示す。プロジェクトの作業工程にボトルネックがあると、それ以外の工程が円滑に進められていたとしても、プロジェクト全体を通して多くの時間を要することになってしまう。

その際、プロジェクトの全体像を把握しているとボトルネックになり得るタスクが事前に把握できるため、トラブルが発生した場合のリカバリーやフォローのシミュレーションを高い確度で行えるようになる [34]。

学習者が学習する際、当日の体調の悪化や予期せぬトラブルの発生など、学習の進度が遅れてしまう要因は多数存在する。そのとき、CPMを使用した手法を扱うことで余裕を持った学習をすることができると考えた。

## 開始終了時間

クリティカルパスから導き出せるものとして、以下各タスクの開始・終了がある。

- Early Start(ES)：最短でタスクを開始できる期間
- Early Finish(EF)：最短でタスクを完了できる期間
- Late Start(LS)：最遅でタスクを開始する期間
- Late Finish(LF)：最遅でタスクが終了する期間

これらを使用して、スケジュールを作成していくことになる [36]。図 3.4 に開始終了時間の例を示す。

## § 3.2 クリティカルパスの解法

今回、クリティカルパスの解法としてLS, LF, EF, ESを使用したものとLP問題 (Linear programming : LP) を使用する。この二つ以外にも様々な方法があるが、今回は主要な2つのやり方でクリティカルパスを解く。例題として用意したパスを解くことにした。LP問題では、目的関数と制約条件を定義してそれに添うようにEF, ES, LS, LFも定めた。その2つは手法が違うが答えが同じになるということを示した。

## 往路時間計算

まず、一つ目の方法として ES, EF, LS, LF を用いる手法である。各タスクに対して、そのタスクに前提条件がない場合、そのタスクの ES を 0 とし、EF を開始時刻にタスクの所要時間を加算して設定する。これはそのタスクがプロジェクトの開始時に開始できるタスクであることを意味する。タスクに前提条件がある場合、その前提条件に対応するタスクの EF を取得、タスクの最大の EF を ES として設定する。タスクの前提条件がすべて完了した後でなければ開始できないためである。ES を開始時刻に設定し、タスクの所要時間を加算して EF を設定する。

この手法は、最短や最長の経路を出すのに特化しておりプロジェクトを進めるための大切な指標となる。しかし、遅延などを考慮しないものが多く、あまり柔軟性のあるものとは言えず人為的なミスが多くなってしまうというデメリットがある。すべてのタスクに対して計算を繰り返した後、最も大きくなった EF を全体の所要時間とし、復路時間計算に使用する。

## 復路時間計算

現在のタスクの前提タスクの位置に配列を作成し、現在のタスクの番号を入れ込む。この作業を繰り返すことで後続タスクのリストを作成する。各タスクに対して、もし後続タスクが存在しない場合、LF をプロジェクト内の全体の所要時間とし、それからタスクの所要時間を引いたものを LS として設定する。

これはそのタスクがプロジェクトの終了と同時に終了するタスクであることを意味する。後続タスクが存在する場合、そのタスクの LS を取得、タスクの最小の LS を LF として設定する。LF からタスクの所要時間を引いたものを LS として設定する。

## フロート

クリティカルなタスクではフロートは 0 になる。フロートが正の値であるタスクは非クリティカルパスに属し、フロートの期間内であれば、遅延してもプロジェクトの完了日に影響は出ない。

フロートには 2 つのタイプがあり、それぞれで計算方法が異なる。

トータルフロートとフリーフロートについて、図??で表す。また、計算方法を以下に示す。

- トータルフロート

プロジェクト終了日の遅延や、スケジュール制約条件からの逸脱が出ない範囲で、ある作業の開始を最早開始日から遅らせることができる期間のことである。 $k$  を現在のタスクの番号とすると、

$$LS_k - ES_k \quad (3.1)$$

または

$$LF_k - EF_k \quad (3.2)$$

で算出する。



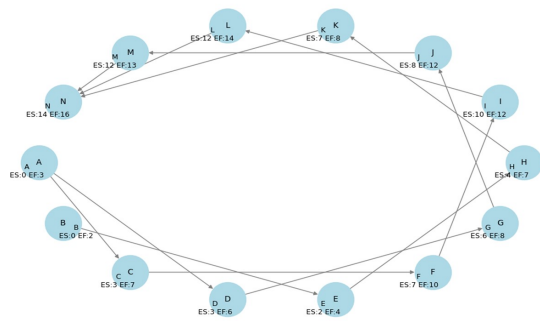


図 3.3: PERT 図

Task	ES	EF	LP Start Time
A		0	3
B		0	2
C		3	7
D		3	6
E		2	4
F		7	10
G		6	8
H		4	7
I		10	12
J		8	12
K		7	8
L		12	14
M		12	13
N		14	16
Project Duration (ES/EF) : 16.0			
Project Duration (LP) : 16.0			

図 3.4: CPM

### ● フリーフロート

後続のタスクに遅延が出ない範囲で、あるタスクの開始を遅らせることができる期間のことである。フリーフロートは、2つ以上のタスクに共通して1つの作業が後続する場合のみに使用され、

$$ES_{k+1} - EF_k \quad (3.3)$$

で算出する。全て使ったとしても、後続のタスクには全く影響が出ない。今回使った式として

$$EF_i = ES_i + D_i \quad (3.4)$$

式??とした。

## 線形計画法

目的関数および制約条件に関する全て線形であるような最適化問題のことを LP と呼ぶ。生産計画問題やダイエット問題などが LP の代表例として知られている。また、LP の解法としては、単体法や内点法が良く知られている。CPM を求める方法として最短路問題やその逆の最長路問題として定式化することができる。

LP を使いことによって決まった値に対して正確な答えを出すことができ、制約条件や目的関数が定義できるため柔軟な対応ができるとされているため人為的な計算ミスが少ないとされている。LP の定式化によっては、リソース制約や追加の最適化条件を組み込むため、単純な ES, EF の結果とは異なる場合がある。ただし、基本的には LP も ES, EF も同じ結果となる。

## 制約条件と目的関数

今回使用した CPM の制約条件と目的関数について説明する。

### 1. 依存関係

$$S_j \geq S_i + D_i(i, j) \quad \forall (i, j) \in E \quad (3.5)$$

となり E はタスクの依存関係を表すエッジの集合である。

### 2. プロジェクト完了時間

$$P \geq S_i + D_i \quad \forall i \in V \quad (3.6)$$

となり  $V$  はすべてのタスクの集合とする.

### 3. 目的関数 (最小化)

$$\min P \quad (3.7)$$

プロジェクト全体の完了時間を最小化するものである.

この LP の制約条件と目的関数とを用いてクリティカルパスを解く. 同様に ES, EF を用いた手法でも同じクリティカルパスを解く. これらはクリティカルパスの期間を示しており, 最小の実行時間を求めるものとなっている. クリティカルパスの経路を下図のように定めた. 式 3.4 と式 3.7 が同じになればよい.

### クリティカルパスメソッド

今回は上記で説明した二つの手法で同じクリティカルパスを解いた. 理論的には両方同じ答えができるが実際に確認してみた. 今回は python 3. 11 での実行を行った. 図 3. 3 は今回解いたクリティカルパスを示しており, このような依存関係がある. 図 3. 4 は LP と ES, EF での両方で解いた場合の結果を表している. 図からも分かるようにお互い同じ結果になっていることがわかる.

この結果より ES, EF を用いた手法と LP を用いた手法で同じ結果が出ているためこの二つの手法での差は制約条件などで変わるためとくに制約条件が厳しいものでない限り違う解が出る可能性は少ない. この結果よりどちらの CPM を用いても差がなくどちらを使用してもよいということがわかる.

## § 3.3 学習計画のためのクリティカルパスの活用

次に学習計画のためのクリティカルパスについて説明する. 実際のデータの例を表 4.1 に示す. このデータは, ユーザーが単元番号 (4, 5, 6, 7, 8) を学習範囲として設定したものである. ユーザー毎に科目数である 20 個の csv が必要になる.

### CPM 関連データ

学習範囲指定データを対象にして, クリティカルパスと各タスクの開始終了時間を求める. 3 章 2 節で求めたように, ES, EF, LS, LF, フロート, 後続タスク, クリティカルなタスクかどうかを求め出力している. 他の要素としては, 単元名, 単元番号, 前提タスク, 所要時間といった要素を抽出する.

実際のデータの例を表??に示す. DESCR は単元名, CODE は単元番号, PREDECESORS は前提単元, SUCCESSORS は後続単元, DAYS は指導時数, SLACK はトータルフロート, Critical はそのタスクがクリティカルか否かを表している. csv の形式は学習範囲指定データと同様である.

### スケジュールデータ

個人個人のスケジュールが入っている.  
作成方法を以下に示す.

表 3.1: CPM 関連データ

DESCR	COD E	PREDEC ESSORS	SUCCESSORS	DAYS	ES	EF	LS	LF	SLACK	CRITICAL
Start	0		['5', '4']	0	0	0	0	0	0	YES
いくつといくつ	4	['0']	['8']	7	0	7	14	21	14	NO
ぜんぶでいくつ	5	['0']	['6']	10	0	10	0	10	0	YES
のこりはいくつ	6	['5']	['7']	8	10	18	10	18	0	YES
どれだけおおい	7	['6']	['8']	3	18	21	18	21	0	YES
10よりおおいかず	8	['4', '7']	['999']	10	21	31	21	31	0	YES
End	999	['8']		0	31	31	31	31	0	YES

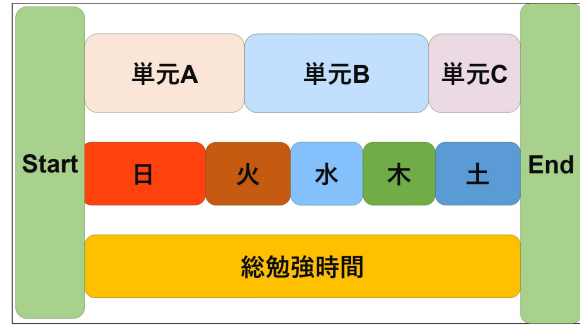


図 3.5: スケジュール作成方法

1. スケジュール作成日から先で最も直近なイベントの開始日までの日付を計算する．例えば，スケジュール作成日が2024年2月1日であり，最も直近なイベントの開始日が2024年2月5日であれば，その期間は2月1日から2月4日までとなる．
2. 期間を計算された日付範囲を曜日ごとに分割する．例えば，2月1日が水曜日であれば，2月1日，2日，3日，4日のそれぞれの曜日が水曜日，木曜日，金曜日，土曜日になる．
3. 各曜日における学習予定時間の合計を求める．学習予定時間データを用いて各曜日に対する予定学習時間を合計する．例えば，水曜日の予定学習時間が2時間，木曜日が3時間，金曜日が4時間，土曜日が0時間であれば，その週の水曜日から土曜日までの学習予定時間の合計は9時間となる．

この予定学習合計時間を  $\text{Total\_Study\_Hours}$  とし，その求め方を式 3.8 に示す．スケジュール作成日の曜日を  $\text{today}$  とすると，

$$\text{Total\_Study\_Time} = \sum_{i=0}^{x-1} \text{study}(\text{today} + i \bmod 7) \quad (3.8)$$

4. 理想合計学習時間の計算を行う学習範囲データの指導時数の合計を求め，理想合計学習時間  $\text{Ideal\_Study\_Time}$  として扱い，その求め方を式 3.9 に示す．

$$\text{Ideal\_Study\_Time} = \sum_{i=1}^n \text{Hours}(\text{unit}_n) \quad (3.9)$$

5. 各単元の指導時数の比率の計算各単元の指導時数を，理想的な合計学習時間に対する比率として扱う．各単元の指導時数を合計したものを理想的な合計学習時間で割ることで，各単元の学習予定時間における割合とする．
6. 単元を LS に基づいて早い順にソートし，計画する上での基準とする．フロートは優先度を示す値ではあるが，開始時刻からどれだけ離れているかを考慮していないため，スケジュールのソートに使用することはできない．代わりに LS が早い作業ほど他の作業の開始に影響を与える可能性が高いため，優先的に実施されるべきであると考え

表 3.2: スケジュールデータ

Day	Date	DOW	Unit	Study Hours	Description
1	2024-01-25	Thu	0	0	Start
1	2024-01-25	Thu	5	1.84210526315789	ぜんぶでいくつ
1	2024-01-25	Thu	6	1.47368421052632	のこりはいくつ
1	2024-01-25	Thu	4	0.68421052631579	いくつといくつ
2	2024-01-26	Fri	4	0.605263157894737	いくつといくつ
2	2024-01-26	Fri	7	0.552631578947369	どれだけおい
2	2024-01-26	Fri	8	0.842105263157895	10よりおおいかず
3	2024-01-27	Sat	8	1	10よりおおいかず
3	2024-01-27	Sat	999	0	End

表 3.3: 圧縮率データ

科目名	圧縮率
小学校国語	0.264705882352941
小学校算数	0.161764705882353
小学校英語	0
小学校理科	0.338235294117647
小学校社会	2.04411764705882

た．優先度の並び順として LS の大小を使用し，作業をスケジュールすることで全体の予定を最適化し，遅延を最小限に抑えることができる．表??を例とすると， $5 \rightarrow 6 \rightarrow 4 \rightarrow 7 \rightarrow 8$  の順番にソートが行われる．

7. 各日における学習スケジュールを生成する．学習予定時間データに基づき，各日の学習可能な時間を考慮し，各日における学習スケジュールを生成する．学習可能時間が週の学習予定時間データから取得され，各日の学習スケジュールに順番にあてはめていく．各単元の学習時間が各日の学習可能時間を超える場合，その単元は複数の日にまたがって学習される (図 4.5 参照)．

日付とそれに対応する単元名，単元番号，単元の学習時間が入れられる．実際のデータの例を表 3.2 に示す．Day はスケジュール作成当日を 1 として，1 日に 1 増加する．

Date はスケジュールを作成する際にカレンダーと対応する日付，DOW は曜日を表している．Unit は単元番号，Description は単元名を，Study Hours は特定の日付 (Day) に特定の単元 (Unit) をどれだけの日数学習するかを書き込んでいる．科目毎に csv が存在し，1 行が Day に行う 1Unit を表す．

## 圧縮率データ

スケジュールを学習範囲単元の指導時数の合計時間からどれだけ伸縮したかを記録している [37]．科目  $k$  の圧縮率を  $Press_k$  とおき，その求め方を式 3.10 に示す．

$$Press_k = \frac{\text{Total\_Study\_Hours}}{\text{Ideal\_Study\_Hours}} \quad (3.10)$$

で求められる．

実際のデータの例の一部を表 3.3 に示す．圧縮率が小さいほど余裕のないスケジュールになってしまっている．そのため，小学校社会では他の科目よりも学習の内容が多いことがわかり，反対に小学校英語は学習を行わないことを示している．csv はユーザー毎に作られ，1 行が 1 単元を表す．

## 信頼性データ

各教材の信頼性スコア  $K\_score(i)$  を求め，記録する．

信頼性スコアの求め方を式 3.11 示す．まず，3 章 3 節で述べたように，各レビューのサクラ性スコア  $F\_score(l_i)$  を計算し，平均化して， $F\_score(i)$  を求める．

$$K\_score(i) = \frac{F\_score(i)}{R\_score(i)} \quad (3.11)$$

教材データと同じ csv ファイルに書き込まれる．



# 提案手法

## § 4.1 学習管理システムへのファジィの導入

学習支援システムにおける CPM の計算の中で単元にかかる時間が決められているが、これには柔軟性が求められる。この単元にかかる時間について不確実性を考慮することによってある程度の想定外の時間のズレに対応できる。考慮するにあたって通常の CPM では不確実性の考慮されていない場合がほとんどだが、ファジィ数を導入したファジィクリティカルパスメソッドを適応することによって不確実性を考慮したスケジュール作成ができる。このスケジュールは工事現場などの天候などに工事日程が左右される現場の場合悪天候などで工事が止まったりするケースが多々見られる。この工数がずれる場合があるのにも対応できるのがファジィ数を考慮した CPM である [37]。

### ファジィ変数

ファジィ変数とは、ファジィ論理やファジィ集合に関連する概念で、二値論理とは異なり、部分的に真であるという状態を考慮に入れるものである。ある値が真または偽だけではなく、0 から 1 の間で真理度を持つ場合に使われる。この変数は従来の変数が取りうる特定の値を持つのに対し、その範囲内で曖昧さを含むような値を取る変数となる。特に人間の直観や自然言語を扱う際に有用で、システムがあいまいな情報を処理するときの手段として用いられています。

### ファジィクリティカルパスメソッド

ファジィクリティカルパスメソッドとは、プロジェクトのスケジュール管理において、タスクの所要時間や関係性が明確でない場合に用いる、CPM の拡張版となっている。

通常の CPM では、タスクの所要時間を固定値として扱い、最も時間がかかるクリティカルパスを特定するものとなっている。しかし、現実のプロジェクトではタスクの所要時間が不確実であることが多いため、ファジィクリティカルパスではファジィ理論という曖昧さを扱う数学的手法を用いて、タスクの所要時間を範囲的に表現するものとなっている。

通常の CPM では決まったパスのみがクリティカルパスになるが、ファジィ CPM ではある程度クリティカルな経路も見えてくるといったものとなっている。特に建築現場の CPM では、天候などの影響で作業時間が変動する場合があるためその場面での使用が有効的である [6]。

この手法も様々な計算方法があるが基本的には CPM の計算方法にファジィ数を追加するものが多く見られる。ファジィ数を追加したものを  $\alpha$ -cut する。

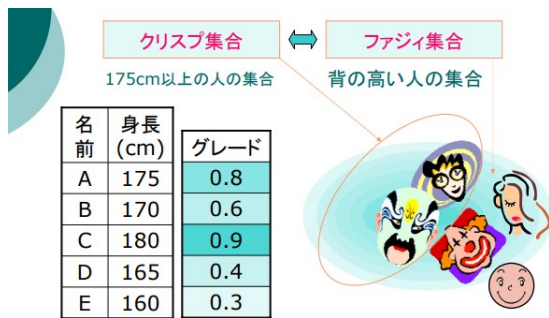


図 4.1: ファジィ集合の例 [42]

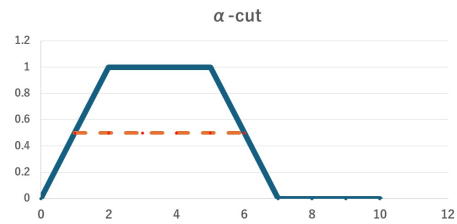


図 4.2: アルファカットの図

## ファジィ集合

ファジィ集合は、数学の集合の概念を拡張したものである。図??のように身長ごとにわけるとき、175cm 以上の人の集合という普通の集合があるが、ファジィ集合では、背の高い人の集合というのがファジィ集合である。

ファジィ集合を数学的に厳密に取り扱うために、各要素毎にグレードという数値を導入する。この数値は論理値のようなもので0と1の間の実数値である。1に近いほどその要素は子の集合に属している度合いが強く、0に近ければあまり属していない。ファジィ集合と明確に区分する必要があるときには、ふつうの集合をクリस्प集合と呼ぶ。

このように、ファジィ集合は従来の厳密な数学的枠組みでは捉えきれない不確実性や曖昧さを含むデータを適切に処理し、より柔軟な意思決定やシステム制御を可能にする重要な概念である。

## メンバーシップ関数

メンバーシップ関数とは、ファジィ集合において各要素がその集合にどの程度属しているかを数値で表す関数である。通常の集合では、要素は「属する (1)」か「属さない (0)」のどちらかに分類されるが、ファジィ集合では、要素が部分的に属することを考慮し、0から1の範囲でその程度を示す。

例えば、タスクの所要時間が「おおよそ7日」と表現される場合、5日や10日もある程度含まれるが、7日に最も強く属すると考えられる。メンバーシップ関数には、三角形型やガウス型などさまざまな形式があり、ファジィ論理やファジィ制御、AIなどの分野で、不確実性を扱うために広く活用されている。

### $\alpha$ -cut

$\alpha$ -cut とは、ファジィ集合において、ある閾値  $\alpha$  ( $0 \leq \alpha \leq 1$ ) を設定し、その値以上のメンバーシップ値を持つ要素の集合を取り出す手法である。

ファジィ集合では、各要素が「どの程度その集合に属するか」を示すメンバーシップ関数によって定義される。しかし、実際の意思決定や計算を行う際には、ファジィな情報がある程度明確にする必要がある。そのため、 $\alpha$ -cut を用いることで、特定の信頼度以上で属している範囲のみを取り出し、より明確なクリस्प集合として扱うことができる。



例えば、あるタスクの所要時間が「最短5日、最も可能性の高い値7日、最長10日」として三角ファジィ数  $A = (5, 7, 10)$  で表される場合、 $\alpha = 0.5$  の  $\alpha$ -cut を求めると、その時点でのメンバーシップ値  $0.5$  以上を持つ範囲が  $A(0.5) = [6, 8.5]$  となる。つまり、「50%以上の確率で所要時間が6～8.5日の間にある」と解釈できる。

このように、 $\alpha$ -cut は、ファジィ集合を部分的にクリस्प集合へ変換する手法として、プロジェクト管理、意思決定、リスク評価などさまざまな分野で活用されている。特に、ファジィクリティカルパス法やファジィ最適化において、タスクの不確実性を考慮したスケジューリングや計画策定に役立つ。

## 学習履歴におけるファジィ数

学習履歴におけるファジィ数として、各单元ごとに時間が決められている。その時間にファジィ数を適用して不確実性を含ませた学習時間とすることで柔軟に対応できる形とした。

図4.5が  $\alpha$ -cut することによって ES, EF に upper と lower にわけることによって複数の解が出てくるようにした。複数の解を出すことによってユーザーに様々なスケジュールを作ることによってどのようにスケジュールを立てるかを複数提示することによってユーザーのスケジューリング能力向上も見込める。またユーザーにわかりやすいように CPM の答えをそのまま出すのではなく、圧縮率といった数値で見てわかりやすい形で提示することでユーザーのストレスにならないようにした。

## § 4.2 ファジィクリティカルパスによる解法

今回の手法のファジィ CPM についての解法を説明する。ここで使用するのは LP と ES, EF, LS, LF の二つの手法を用いた方法で、この二つの解法を用いて行った。また、その実行結果として同じ解が見られた。

### ファジィ線形計画問題

今回行ったファジィ LP として目的関数を下のように定義した。

$$\tilde{D} = \max \sum_{i=1}^n \sum_{j=1}^n (T_{ij})_{\alpha}^U x_{ij} \quad (4.1)$$

$$s. t \quad \sum_{j=1}^n x_{1j} = 1, \quad (4.2)$$

$$\sum_{j=1}^n x_{ij} = \sum_{k=1}^n x_{kj}, i = 2, \dots, n-1 \quad (4.3)$$

$$\sum_{j=1}^n x_{kn} = 1 \quad (4.4)$$

$$x_{ij} \geq 0, (i, j) \in A. \quad (4.5)$$

実行結果

$\alpha$	経路										最小時間(days)	最大時間(days)
$\alpha=0.0$ , Path:	['P1', 'P3', 'P4', 'P5', 'P6', 'P9', 'P12', 'P14', 'P15', 'P21']										150.0,	165
$\alpha=0.0$ , Path:	['P1', 'P3', 'P4', 'P5', 'P6', 'P9', 'P12', 'P14', 'P25']										129.0,	142
$\alpha=0.1$ , Path:	['P1', 'P3', 'P4', 'P5', 'P6', 'P9', 'P13', 'P14', 'P25']										121.9,	143.9
$\alpha=0.2$ , Path:	['P1', 'P3', 'P4', 'P5', 'P6', 'P9', 'P17', 'P19', 'P26', 'P27']										125.5,	153.2
$\alpha=0.3$ , Path:	['P1', 'P3', 'P4', 'P5', 'P6', 'P9', 'P23', 'P24', 'P27']										117.7,	149.7
$\alpha=0.4$ , Path:	['P1', 'P3', 'P4', 'P5', 'P6', 'P10']										103.1,	135
$\alpha=0.5$ , Path:	['P1', 'P3', 'P4', 'P5', 'P6', 'P9', 'P18']										114.0,	156
	-											
	-											
	-											
$\alpha=1.0$ , Path:	['P1', 'P3', 'P4', 'P5', 'P6', 'P9', 'P18']										101.0,	170

図 4.3: 線形計画問題を解いた解

実行結果												
$\alpha$	経路									最小時間(days)	最大時間(days)	
$\alpha=0.0$ , Path:	['P1',	'P3',	'P4',	'P5',	'P6',	'P9',	'P12',	'P14',	'P15','P21']	150.0,	165	
$\alpha=0.0$ , Path:	['P1',	'P3',	'P4',	'P5',	'P6',	'P9',	'P12',	'P14',	'P25']	129.0,	142	
$\alpha=0.1$ , Path:	['P1',	'P3',	'P4',	'P5',	'P6',	'P9',	'P13',	'P14',	'P25']	121.9,	143.9	
$\alpha=0.2$ , Path:	['P1',	'P3',	'P4',	'P5',	'P6',	'P9',	'P17',	'P19',	'P26','P27']	125.5,	153.2	
$\alpha=0.3$ , Path:	['P1',	'P3',	'P4',	'P5',	'P6',	'P9',	'P23',	'P24',	'P27']	117.7,	149.7	
$\alpha=0.4$ , Path:	['P1',	'P3',	'P4',	'P5',	'P6',	'P10']				103.1,	135	
$\alpha=0.5$ , Path:	['P1',	'P3',	'P4',	'P5',	'P6',	'P9',	'P18']			114.0,	156	
	-											
	-											
	-											
$\alpha=1.0$ , Path:	['P1',	'P3',	'P4',	'P5',	'P6',	'P9',	'P18']			101.0,	170	

図 4.4: スケジューリング法を用いた解

$$D_x^L = \min y_n - y_1 \quad (4.6)$$

$$s. t \quad y_j \geq y_i + (T_{ij})_{\alpha}^L, (i, j) \in A, \quad (4.7)$$

$$y_j, y_i \forall (i, j) \in A. \quad (4.8)$$

このように最大最小を求める式を定めた。これを用いてクリティカルパスを解く。[6]

## ファジィスケジューリング法

前章で述べた手法の ES, EF, LF, LS, を用いた手法である。

$$ES_i = \max(EF_j) \quad (4.9)$$

$$EF_i = ES_i + Duration_i \quad (4.10)$$

この式 4.9, 式 4.10 は, EF が最小時間なので最も作業時間の短いパスを求める式となっている。

$$LF_i = \min(LS_j) \quad (4.11)$$

$$LS_i = LF_i - Duration_i \quad (4.12)$$

この式 4.11, 式 4.12 は, LF はあるタスクが遅くとも終了できる時間である。よってこれは最も長いパスを求める式となっている。

式 4.1 と式 4.9 は最小値を求める式となっており, 式 4.6, 式 4.11 は最大値を求める式となっている。

この二つの手法を使った時のクリティカルパスを図 4.5, 条件を表 4.1 とした。今回は最小作業時間と最大作業時間を求めた。両方の式に対して 0~1 の 0.1 刻みで  $\alpha$ -cut を行いその結果として全てのパスに対してどのくらいの時間がかかるのかを求めた。また, その両方の式の値の比較を行った。また, 今回の環境として python 3.1.1 で行った。

表 4.1: 制約条件 [39]

Activity Item	Activity Description	Precedence Item	Fuzzy Operation Times			
			L	U	ls	rs
P1	Concrete works foundation	-	28	30	3	5
P2	Insulation works	P1	4	4	1	1
P3	Parking area + Roads + Landscape	P2	29	30	4	5
P4	Back filling works	P3	7	12	4	3
P5	Sub-base	P4	6	6	1	4
P6	Steel structure erection	P5	30	35	4	5
P7	Under ground drainage system	P5	10	10	3	3
P8	Water tank – civil works	-	21	21	6	4
P9	Steel structure testing	P6	3	4	1	1
P10	Roofing works	P6	10	12	1	3
P11	Water tank - finishing	P8	7	8	1	2
P12	HVAC works – 1 <sup>st</sup> fix	P9	14	12	2	2
P13	Fire fighting works 1 <sup>st</sup> fix	P9	9	11	2	1
P14	Electrical system works – 1 <sup>st</sup> fix	P12,P13	6	7	1	3
P15	Flooring	P14	9	11	2	1
P16	HVAC work – 2 <sup>nd</sup> fix	P9	14	12	2	2
P17	Fire fighting works – 2 <sup>nd</sup> fix	P9	9	11	2	1
P18	Cladding works	P9	24	25	9	5
P19	Electrical system works – 2 <sup>nd</sup> fix	P16,17	6	7	1	3
P20	Water tank - MEP	P11	11	12	2	2
P21	Finishing works	P15	18	18	3	2
P22	HVAC works – 3 <sup>rd</sup>	P9	14	12	2	2
P23	Fire fighting work – 3 <sup>rd</sup> fix	P9	9	11	2	1
P24	Electrical system works – 3 <sup>rd</sup> fix	P22,P23	6	7	1	3
P25	Plumbing works – 1 <sup>st</sup> fix	P14	6	6	1	2
P26	Plumbing works – 2 <sup>nd</sup> fix	P19	6	6	1	2
P27	Plumbing works – 3 <sup>rd</sup> fix	P24	6	6	1	2
P28	Water tank testing	P20	2	2	1	1
P29	Testing and commissioning	P28	2	2	1	1
P30	Snag list and Initial handling	P29	7	7	2	2

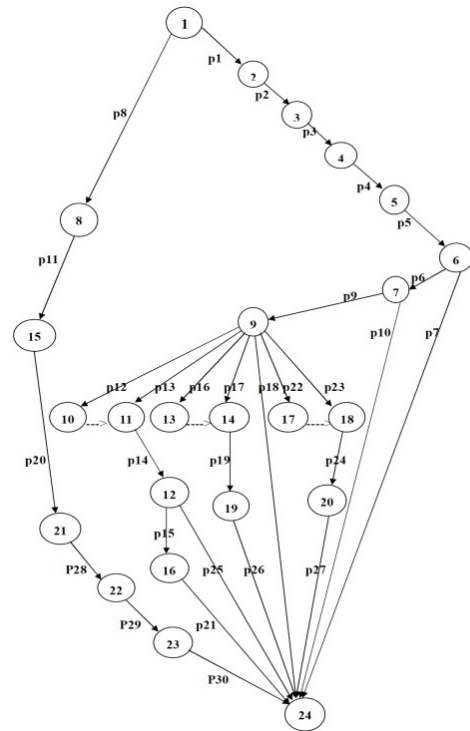


図 4.5: クリティカルパス [39]

図 4.3, 図 4.4 を比較すると同じ値が出たことがわかる. 理論では同じ結果が出ることがわかっていて実際にプログラムを動かしてみると同じ結果ということがわかった. これにより, 学習履歴の活用としてどちらの手法を用いても同じ結果となることが予想される.

## § 4.3 対話型システム

今回行った手法では, 0~1 の 0.1 刻みで  $\alpha$ -cut を行ったため CPM の解が 10 個出る形になった. この複数個の解を有効に柔軟性を持たせるために対話型を用いた. 対話型にすることによってユーザーの満足度が向上すると考えられる.

### 対話型

対話型といっても様々な種類が存在する. ユーザーの入力に対してシステムが即座に応答し, 次の操作を促すような仕組みだったり, プログラムの実行時にユーザーの入力を受け付け, それに基づいた処理を進めるようなシステムだったり様々なシステムが存在する. 最近では図 4.6 のように対話型 AI などが導入され, 普段の生活の中でも見ることの多いものとなっている.

対話型を使った授業などが今現在取り入れられている学校が多い. 特に国語の授業では, 対話型を取り入れることによってより基礎的・基本的な知識をより一層学習することができるといったメリットがある. このことから対話型の AI や対話型システムを導入することによって学習にとって良いものになると考え対話型を取り入れた.

### 対話型 AI

対話型 AI とは、ユーザーとの会話を通じて情報交換やタスクの実行を行う人工知能システムのことである。特に言語を使って人間とのやり取りを行うことに特化しており、近年では多くの場面で利用されている。対話型 AI は、主に自然言語処理（NLP）技術を活用して、ユーザーが入力するテキストや音声を理解し、適切な応答を生成する。この技術により、AI はユーザーが伝えたいことを解釈し、それに基づいて回答やアクションを提供する。

また、対話型 AI は音声認識技術や音声合成技術を組み合わせることによって、音声を通じてのやり取りも可能にする。例えば、スマートフォンの音声アシスタントである Siri や Google Assistant, Amazon の Alexa などがその代表的な例である。これらの AI は、音声で指示を出すことで、天気予報の確認や音楽の再生、カレンダーの管理といったタスクを実行する。これにより、ユーザーは手を使わずに情報を取得したり、操作を行ったりすることができ、より直感的で便利な体験が提供されている。

対話型 AI には、タスク指向型と会話型という二つのタイプがある。タスク指向型 AI は、特定の目的に対して実行されるタスクに焦点を当て、ユーザーの指示に従ってそのタスクを完了させることを主な目的としている。一方、会話型 AI は、より自由な会話を行い、ユーザーとの対話を楽しむことを目的としている。このような AI は、雑談や質問応答を通じてユーザーとの関係を深め、エンターテインメントやリラックスの手段として活用されることもある。

対話型 AI の利用例としては、カスタマーサポートやパーソナルアシスタントなどが挙げられる。カスタマーサポートにおいては、AI がユーザーの質問に即座に応答し、問題解決を手助けする。これにより、24 時間対応が可能となり、企業側は効率的なサービス提供を実現できる。また、パーソナルアシスタントでは、ユーザーのスケジュール管理や情報検索、リマインダーの設定など、日常的なタスクを AI がサポートする。これにより、ユーザーは時間を節約し、より効率的に生活することができる。

対話型 AI のメリットは、何と言ってもその効率性にある。ユーザーは必要な情報を迅速に得ることができ、AI が常に対応してくれるため、時間や労力を大幅に削減することが可能である。また、対話型 AI はユーザーごとに個別対応ができるため、パーソナライズされたサービスが提供される。ユーザーが何を求めているのか、どのような回答が適切かを学習し、よりの確に応答できるようになるため、体験はどんどん快適になる。

ただし、対話型 AI にも課題がある。例えば、AI は言語の微妙なニュアンスや感情を完全に理解することが難しいため、時には不自然な応答や誤解が生じることがある。また、ユーザーとの会話が長くなると、AI が意図を見失うことがある。さらに、AI が収集するデータに対するプライバシーの懸念やセキュリティの問題もある。個人情報がどのように扱われ、どれだけ安全に管理されているかがユーザーにとって重要な関心事となるため、この部分の改善が求められる。

それでも、対話型 AI はますます進化を続けており、私たちの日常生活における重要なツールとしての地位を確立しつつある。これからも技術の進化によって、さらに多くの分野での利用が進み、より深いレベルで人間と AI の協力が進むと考えられる。

このようなメリットとデメリットがあるが、ChatGPT などの対話型 AI を使用しようとするとお金がかかるパターンがある。これは最初に掲げたどのユーザーにも使いやすいようにしたいといった思いから離れてしまう。安価であるためお金のかからない対話型シェルを採用した。

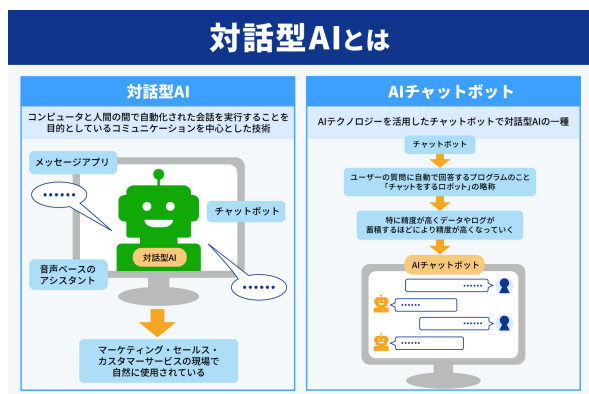


図 4.6: 対話型 AI [40]

Day	Date	Day of Week	Unit	Study Hours	Description
0	1 2025/2/4	Tue	0	0.000000	Start
1	1 2025/2/4	Tue	1	1.818182	い
2	1 2025/2/4	Wed	2	1.181818	おはな
3	2 2025/2/5	Wed	2	0.636364	おはな
4	2 2025/2/5	Wed	3	0.909091	あつまつ
5	2 2025/2/5	Wed	4	0.909091	えん
6	2 2025/2/6	Thu	5	0.545455	えん
7	3 2025/2/6	Thu	5	2.000000	ど
8	4 2025/2/6	Thu	5	0.181818	ど
9	4 2025/2/7	Fri	6	1.818182	なん
10	4 2025/2/7	Fri	999	0.000000	End
0	1 2025/2/4	Tue	0	0.000000	Start
1	1 2025/2/4	Tue	1	1.818182	い
2	1 2025/2/4	Tue	2	1.181818	おはな
3	2 2025/2/5	Wed	2	0.636364	おはな
4	2 2025/2/5	Wed	3	0.909091	あつまつ
5	2 2025/2/5	Wed	4	0.909091	えん
6	2 2025/2/6	Thu	5	0.545455	えん
7	3 2025/2/6	Thu	5	2.000000	ど
8	4 2025/2/7	Fri	5	0.181818	ど
9	4 2025/2/7	Fri	6	1.818182	なん
10	4 2025/2/7	Fri	999	0.000000	End
0	1 2025-02-04	Tue	0	0.000000	Start
1	1 2025-02-04	Tue	1	1.818182	い
2	1 2025-02-04	Tue	2	1.181818	おはな
3	2 2025-02-05	Wed	2	0.636364	おはな
4	2 2025-02-05	Wed	3	0.909091	あつまつ
5	2 2025-02-05	Wed	4	0.909091	えん
6	2 2025-02-05	Wed	5	0.545455	えん
7	3 2025-02-06	Thu	5	2.000000	ど
8	4 2025-02-07	Fri	5	0.181818	ど
9	4 2025-02-07	Fri	6	1.818182	なん
10	4 2025-02-07	Fri	999	0.000000	End

図 4.7: 選択肢の例

## 対話型シェル

対話型シェルとは、ユーザーがコマンドを入力すると、その場で即座に処理が実行され、結果が返されるシステムである。主にターミナルやコマンドラインで動作し、ユーザーとリアルタイムにやり取りができる環境を提供する。

このシェルの特徴として、即時応答が挙げられます。ユーザーが入力したコマンドは、スクリプトのように一度にまとめて実行されるのではなく、一行ずつ処理され、その結果が即座に表示されます。このため、動作確認やデバッグを行う際に非常に便利である。また、過去に入力したコマンドの履歴を遡ったり、自動補完機能を活用したりすることもできる。

対話型シェルは、主にオペレーティングシステムやプログラミング言語の環境で使用される。例えば、Linux や macOS で使用される Bourne Again Shell は、基本的なコマンドを入力することで、ファイルの操作やシステム管理を行うことができる。また、Python の対話型シェル (REPL: Read-Eval-Print Loop) を使えば、Python コードを一行ずつ試しながら実行することが可能である。同様に、Node.js には JavaScript を対話的に実行できる REPL があり、Ruby には IRB (Interactive Ruby) が用意されている。

スクリプトと対話型シェルの違いとしては、スクリプトはあらかじめ記述されたコードをまとめて実行するのに対し、対話型シェルはその都度コマンドを入力しながら実行していく点が挙げられる。このため、対話型シェルは試行錯誤しながら作業を進めるのに向いており、一方でスクリプトは複雑な処理や繰り返しのタスクを自動化するのに適している。

実際の活用例としては、Linux のシェルを使ってファイルやディレクトリの管理を行ったり、Python の対話型シェルを使って計算処理やライブラリの動作確認をしたりすることが挙げられる。また、データベース管理では、MySQL や PostgreSQL の対話型シェルを利用して、SQL クエリを直接実行することができる。

このように、対話型シェルは即時実行という特性を活かし、試行錯誤をしながら作業を進める際に非常に有用なツールである。システムの管理、プログラムの開発、データ処理

など，さまざまな分野で活用されている．

## 選択肢

今回は，出た解全てを見せるのではなくユーザーにわかりやすいように3つの値を見せることとした．3つの値というのは圧縮率であり，どの程度スケジュールに余裕があるのかまたは，厳しいスケジュールなのかが一目でわかるようにした．こちらが提案する圧縮率は最大値と最小値と中央値として提案する．

また，図 4.7 のように3つの圧縮率によって日程が変わるためその圧縮率の日程を見せることによって，よりどのようなスケジュールなのかわかりやすいようにすることによって満足度が向上する．このシステムを導入することによってどのようにスケジュールを立てているのかわかりやすいため，スケジュール作成をする小中高生に見えるようにすることによってスケジュール作成を今後自分自身でする手助けになりと考える．

# 実験結果並びに考察

## § 5.1 実験の概要

本研究の数値実験として、システムの有用性の検証、科目推薦の有効性を行う。システムの有用性の検証では実際にシステムを使用してもらい、アンケートに答えてもらう。アンケートの項目は全部で10個あり、また、アンケートと同時にコメントを記入できる欄を設けておき、自由にコメントをできるようにする。このアンケートを持って本システムの有用性の検証を行う。以上のコメントを5段階のリッカート尺度で評価してもらう。

リッカート尺度とは、あるトピックに対して、多段階の選択肢を用いたアンケートを取り、回答者がどの程度同意するかを測定する手法のことである [?]. 両極に位置する選択肢の間に段階的な選択肢を設定することで、単純な2択では測定しにくい質問でも回答を集めやすいという特徴がある。

今回のアンケートでは5段階のうち、1をとてもそう思う、2をそう思う、3をどちらでもない、4をそう思わない、5をとてもそう思わないといったように回答を設定し、アンケートに答えてもらう。

アンケート項目全体を通して、基本的にはシステムの使用感に関する質問を多くしてある。本来なら本システムを使用し、学習効果が向上するのを確認することでシステムの有効性を検証するべきだが、確認するためには開発したシステムを長期間使用してもらわないと学習効果が向上したかを確認することができない。そのため、本研究ではアンケート調査を用いてシステムの有効性を示す。

調査の対象は同研究室の修士1年生が1名、学部4年生が5名、3年生が3名の合計9名に実際に開発したシステムを使用してもらい、アンケートを答えてもらった。行った質問を表5.1に示す。実際に使用してもらうにあたり、システムの使用手順について説明を行い、実際に使用してもらう。

手順は以下に記してある通り、新規登録から個人データの入力、スケジュール作成と系統図グラフ作成、教材のレビューまでの一連の流れを説明した。

### 実際に使用したシステムの操作ガイド

1. 新規登録画面で、ユーザID、パスワード、ニックネームを入力してください (図5.1 参照).
2. 登録したユーザー情報を使用してログインしてください (図5.1 参照).

表 5.1: システムの評価基準

Q1	システムの操作性はわかりやすいか	Q2	システムの機能は理解しやすいか
Q3	レイアウトは親切か	Q4	デザインは見やすいか
Q5	利用にストレスを感じたか	Q6	対話型は便利か
Q7	自分通りのスケジュールになったか	Q8	対話型によって好きなスケジュールを選べたか
Q9	学習の効率が上がるか	Q10	学習のモチベーションが上がるか

3. 右上の画面で勉強可能な時間帯をクリックして選択してください (図 5.2 参照). 入力完了後, 「合計時間を保存」 ボタンを押して保存してください.
4. 左下の画面で「予定追加」 ボタンを押して, 表示される青いボックスをドラッグして次のテストの日程に配置してください. 配置が完了したら, 「書き込み」 ボタンを押して保存してください.
5. 「理解度チェック」 ボタンから好きなものを選択し, 単元のバーを動かしたりチェックボックスをつけたりしてください.
6. 再ログイン後, 「スケジュール日程作成」 ボタンを押して, 出たきた圧縮率とスケジュールの中で好きなものを選んでください (図 5.3. 表示されたカレンダーの日付をクリックして確認してください (図 5.4 参照)).
7. 「系統図グラフ作成」 ボタンを押して, プルダウンメニューから理解度チェックで登録した科目を確認してください. グラフに理解度とテスト期間, クリティカルパスが反映されていることを確認してください.
8. グラフで好きなノードから出るボックスをクリックし, 教材メニューに移動してください. レビューと評価を行い, 教材の信頼度スコアが変化することを確認してください.

実際に操作してもらう UI に関して説明を行う.

## UI の解説

### 1. 図 5.1

ログイン画面を表しており, 新規登録の際は同じ名前は存在できても, 同じユーザー ID は登録できないようになっており, ユーザー ID の被りが出現すると, 警告メッセージを出力するようになっている. 誤ログインの防止に役立ち, アカウントのセキュリティ向上に貢献する. システム側もユーザーを容易に識別できるというメリットがある. また, ユーザーデータの整合性を保つため, ユーザー ID は int 型である必要がある. 新規登録で半角数字以外が入力された場合, 警告メッセージを出力する. また, 登録されていないユーザーだとログインができないようになっている.

### 2. 図 5.2



ログイン	新規登録
ユーザー名	新しいユーザー名
パスワード	新しいパスワード
ログイン	新規登録

品目別 品名 単位

	四角目	円角目	六角目	三角目	一角目	金貨	土貨
00:00							
01:00							
02:00							
03:00							
04:00							
05:00							
06:00							
07:00							
08:00							
09:00							
10:00							
11:00							
12:00							

品目別品名と時刻  
 四角目: 3:00 時刻  
 円角目: 4:00 時刻  
 六角目: 3:00 時刻  
 三角目: 4:00 時刻  
 一角目: 2:00 時刻  
 金貨: 4:00 時刻  
 土貨: 1:00 時刻

図2 時刻表作成

図 5.1: 新規登録, ログイン

図 5.2: 勉強時間入力

Date		Date of Week	Unit	Study Hours	Description
Day	2025/7/4	Tue	0	0.000000	Start
	2025/7/5	Tue	1	0.181818	たふさくはたふさく
	2025/7/6	Wed	2	1.181818	おはしはたふさくしない
	2025/7/7	Thu	3	0.909091	おはしはたふさくしない
	2025/7/8	Med	3	0.909091	おはしはたふさくしない
	2025/7/9	Wed	4	0.909091	おはしはたふさくしない
	2025/7/10	Thu	5	2.000000	おはしはたふさくしない
	2025/7/11	Fri	5	1.181818	おはしはたふさくしない
	2025/7/12	Fri	6	1.181818	おはしはたふさくしない
	2025/7/13	Fri	6	1.181818	おはしはたふさくしない
Day	2025/7/4	Tue	0	0.000000	Start
	2025/7/5	Tue	1	0.181818	たふさくはたふさく
	2025/7/6	Tue	2	1.181818	おはしはたふさくしない
	2025/7/7	Wed	3	0.909091	おはしはたふさくしない
	2025/7/8	Med	3	0.909091	おはしはたふさくしない
	2025/7/9	Wed	4	0.909091	おはしはたふさくしない
	2025/7/10	Thu	5	2.000000	おはしはたふさくしない
	2025/7/11	Fri	5	1.181818	おはしはたふさくしない
	2025/7/12	Fri	6	1.181818	おはしはたふさくしない
	2025/7/13	Fri	6	1.181818	おはしはたふさくしない
Day	2025-02-01	Tue	0	0.000000	Start
	2025-02-02	Tue	1	0.181818	たふさくはたふさく
	2025-02-03	Wed	2	1.181818	おはしはたふさくしない
	2025-02-04	Wed	3	0.909091	おはしはたふさくしない
	2025-02-05	Med	3	0.909091	おはしはたふさくしない
	2025-02-06	Wed	4	0.909091	おはしはたふさくしない
	2025-02-07	Thu	5	2.000000	おはしはたふさくしない
	2025-02-08	Fri	5	1.181818	おはしはたふさくしない
	2025-02-09	Fri	6	1.181818	おはしはたふさくしない
	2025-02-10	Fri	6	1.181818	おはしはたふさくしない

[illegible]

図 5.3: 対話型選択

図 5.4: 学習日程

1週間分の曜日を列に、1日を分割した表形式で表示している。セルをクリックすることで色が変化する。色が変わったセルは、その時間帯に勉強を予定することを表している。また、列ごとに合計した数値は曜日ごとの合計勉強時間として扱われ、勉強時間データとして保存される。この表では、15分、30分、60分の3つの区切りで時間を分割することができる。ユーザーは直感的に空いている時間帯を考えながら入力することが想定される。

## 3. 図 5.3

選んで貰った単位をもとにファジィCPMを求め、それに伴って圧縮率を3つ提示し、また、それにあったスケジュールを提示し、好きなスケジュールを選んでもらう。

4. 図 5.4 コマンドプロンプトに表示されていたスケジュールを実際のサイトに表示させ、自分のすべきものが何かを見やすく表示させている。また、出ている単元を選んで貰うとその単元に関する youtube や、ホームページに飛べるようになっている。

## § 5.2 実験結果と考察

合計すると、肯定的な評価が52件、否定的な評価が20件となった。表5.2にアンケート結果を示す。それぞれの質問項目について以下で考察する。

## アンケート項目の考察

1. システムの操作性はわかりやすいか

結果として、肯定的な評価が5件となった。システム全体を通して単元の評価の仕方や時間の入力やスケジュールを作成した際の単元がわかりやすいような設計になっていると示唆された。

## 2. システムの機能は理解しやすいか

結果として、肯定的な評価は8件で否定的な評価はなかった。シンプルなページの作りとなっているので、全体的に理解はしやすいようになった。また、複数のスケジュールを提案する場所で機能として見やすいものとなったことによって理解しやすいものになった。

## 3. レイアウトは親切か

結果として、肯定的な評価が3件となった。この結果は、ユーザー遷移を最小限に抑えたシンプルなレイアウト設計が功を奏したと考えられる。ユーザーにとって親切的なレイアウトであると示唆されている。

## 4. デザインは見やすいか

結果として、肯定的な評価が5件となり、見やすいデザインであることが評価された。しかし、否定的な評価が2件となった。cssやjava scriptを用いて、カーソルを合わせている部分の色を変える、画面をスタイリッシュに装飾するような、視覚的に分かりやすい工夫が評価されていると考えている。また、系統図グラフのノードの色も彩度を落として長時間見ても目が疲れなくなっていることも評価を上昇させている可能性がある。しかし、対話型の部分がデザインのなさで否定的な評価が増えた。

## 5. 利用にストレスを感じたか

果として、肯定的な評価が3件に対し、否定的な評価は6件出てしまった。ユーザーがストレスを感じる要因として、個人情報を入力に時間がかかり、ログアウトが必要な場面が存在することが挙げられる。本実験の回答者は実際の対象者とは無関係であり、実際に扱うわけではないデータを入力することが一因となったかもしれない。また、対話型の部分がコマンドプロンプトになっているためページを押しているのに違う場所を押さなければならない場面でストレスに感じ、否定的になったかもしれない。

## 6. 対話型は便利か

結果として、6件の肯定的な評価となった。圧縮率に対してスケジュールを表示させているため、どのようなスケジュールが組まれるのかを理解しやすいというコメントをいただいたので、その部分の対話型は優れているということがわかる。

## 7. 自分通りのスケジュールになったか

結果として、肯定的な評価が4件あった。否定的な評価はなかったが、どちらでもないが5件となっているため、3つの提示だけでは少なかったのかと考える。もう少し選択肢を増やすことによって自分通りのスケジュールになる確率が上がるため3つよりも増やしていくという改善策も出てきた。

表 5.2: アンケート結果

質問内容	とてもそう思う	そう思う	どちらでもない	そう思わない	とてもそう思わない
Q1	1	4	3	1	0
Q2	2	6	1	0	0
Q3	3	2	2	1	1
Q4	4	4	1	0	0
Q5	2	4	0	2	1
Q6	3	3	0	2	1
Q7	2	2	5	0	0
Q8	2	2	2	2	1
Q9	0	4	4	1	0
Q10	5	0	0	3	1

#### 8. 対話型によって好きなスケジュールを選べたか

結果として、4件の肯定的な評価があった。Q7と同じように、好きなスケジュールの可能性が少し少ないように感じたため、3つ以上に選択肢を増やす必要があると示された。

#### 9. 学習の効率が上がるか

結果として、4件で肯定的な評価があった。こちらのアンケートの結果としては短期的なもので長期になるとアンケート結果がどのようになるかわからないが、短期であってもスケジュールを作成して、少しばかりか学習の手助けとなっていることがわかる。

#### 10. 学習のモチベーションが上がるか

結果として、肯定的な評価が5件となった。目標を明確にしてスケジュールを設定することや、自分が興味のある単元の類似単元を系統グラフから参照し、スムーズに教材を提供できることが高評価につながったと考える。しかし、否定的な評価も1件存在する。コメントでは実際に問題を解くプロセスが存在せず、学習が身につかないことを指摘された。

#### 11. 自由記入であるコメント欄

コメントとして、スケジュール提案の場所の実行がコマンドプロンプトではなくサイト上によって見やすさと操作性が向上するかもしれないといった意見があった。実行時のサイトとコマンドプロンプトを行き来しなければならないのでその部分の意見があったが、その部分は改善していかなければならない場所である。この部分を改善することによってユーザーの使用時のストレスや、快適さが格段に変わると予想される。

以上のアンケート結果の総括として、本システムはユーザーにとって十分有用であることを示せた。



# おわりに

本研究では、小中高生に向けたいつでも作成できる柔軟なスケジュールかつ不確実性にも考慮し、自分の好きなスケジュール組み立てられるように対話型を用いてのスケジュール作成システムの開発を行った。

具体的には自分で勉強する時間を選んでもらい自分のしたい単元を選んでもらいその勉強時間と単元にかかる時間にファジィ変数を導入して、スケジュール作成を行うことを行った。また、そこで出される圧縮率を提示し、かつ圧縮率に対してどのようなスケジュールになるかを提示し、選んでもらいそれによってスケジュールを自分の好きなように作成できるようになった。今回はES, EF, LS, LFを用いるファジィクリティカルパスを用いた。これによって不確実性を考慮したCPMを解くことができた。出てきた単元をクリックするとその単元の説明を行っているyoutubeや、ホームページを即座にいくことができるため1人で自主学習を行う際でも自分のわからなかった詰まったところに素早く効率よく勉強ができるように教材として提供することができた。

もう一つの目標としてまだスケジュール作成においては未熟な小中高生に対して、 $\alpha$ -cutを行い解を複数だし、それごとのスケジュールを出すことによってこちらが指定した一つのスケジュールにこだわるのではなく、複数のスケジュールを見ることによってスケジュールを立てることに対しての参考になればよいと思う。

実験では、学生用システムを複数人に使用してもらいアンケートに答えてもらった。その結果、アンケートを通して肯定的な意見が多く、このことから開発したシステムの有用性を示した。しかし、これは大学生であり長期の調査になっていないためシステムが短期のものには有用性があるが長期になると有用性があるか調べる必要がある。また、システムの流れがわかりにくいといった意見があったため説明を一連の流れを説明せずに使えるように完全しなければならない。

今後の課題として、現在は対話型としてコマンドプロンプトに数値を入力してもらうようになっているが、これはあまりデザインとしてはきれいなものとは言えないためページ上に表示させる形にできることができればよいものとなると思う。また、今回は不確実性のみ考慮しているが、不確立性にも考慮しなければならない。不確立性を考慮することによってどのくらいさぼるかなどを考慮することによってより小中高生にあったスケジュール提案システムになる。さらに、短期のみの調査なため今後は小中高生を対象にした中間テストや期末テストなどに焦点を当てたスケジュール作成を行いそれに伴っての成績が向上したかどうかの調査が必要となってくる。



# 謝辞

本研究を遂行するにあたり，多大なご指導と終始懇切丁寧なご鞭撻を賜った富山県立大学工学部情報システム工学科 António Oliveira Nzinga René 講師, 奥原浩之教授，に深甚な謝意を表します。最後になりましたが，多大な協力をして頂いた，奥原・René 研究室の同輩諸氏に感謝致します。

2025 年 2 月

島崎 圭介





## 参考文献

- [1] “国内人口推移が 2030 年の「働く」にどのような影響を及ぼすか ”, <https://www.recruitms.co.jp/research/2030/report/trend1.html>, 閲覧日 2025.1.20.
- [2] ベネッセ教育情報, “学力格差とは? ”, [https://benesse.jp/educational\\_terms/1.html](https://benesse.jp/educational_terms/1.html), 閲覧日 2025.2.2.
- [3] ベネッセ教育情報, “計画性がない子の特徴と原因は? 計画性を鍛えるためにすべきこと ”, <https://benesse.jp/kyouiku/202108/20210820-2.html>, 閲覧日 2025.2.2.
- [4] デジタルナレッジ, “E ラーニングとは ”, <https://www.digital-knowledge.co.jp/el-knowledge/e-learning/>, 閲覧日 2025.2.2.
- [5] 駿台グループ, “e ラーニングとは ”, <https://satt.jp/e-learning/e-learning.html>, 閲覧日 2025.2.2.
- [6] 文部科学省, “GIGA スクール構想に関する各種調査の結果 ”, pp.1-34, 2021.
- [7] 垂見裕子, “小学生の学習習慣の形成メカニズム 日本・香港・上海の都市部の比較 ”, 比較教育学研究, 第 55 号, pp.89-110, 2017.
- [8] MASTER EDUCATION, “【徹底解説】今さら聞けない GIGA スクール構想とは? 基本から実施スケジュール、文部科学省の支援まで解説 ”, [https://master-education.jp/column/about\\_giga\\_school/](https://master-education.jp/column/about_giga_school/), 閲覧日 2025.2.2.
- [9] コエテコ, “アクティブラーニングとは? 文部科学省が推進している理由や事例を紹介 ”, <https://coeteco.jp/articles/10663>, 閲覧日 2025.2.2.
- [10] ベネッセ教育総合研究所 学び・生活研究室, “「小中学校の学習指導に関する調査 2021」小学校版集計表 ”, pp.1-17, 2021.
- [11] 公益財団法人 日本財団, “家庭の経済格差と子どもの認知・非認知能力格差の関係分析”, pp.1-20, 2017.
- [12] 河合塾, “新課程入試のポイント ”, <https://www.keinet.ne.jp/exam/2025/point/>, 閲覧日 2025.2.2.
- [13] 文部科学省, “幼稚園教育要領、小・中学校学習指導要領等の改訂のポイント ”, [https://www.mext.go.jp/content/1421692\\_1.pdf](https://www.mext.go.jp/content/1421692_1.pdf), 閲覧日 2025.2.2.
- [14] HOME 個別指導塾, “数学・英語・理科の電車風学習系統図の使い方説明書 ”, <https://home-kobetsu.com/?p=258>, 閲覧日 2025.2.2.
- [15] 教育出版, “資料ダウンロード ”, <https://www.kyoiku-shuppan.co.jp/2020shou/sansu/category04/download.html>, 閲覧日 2025.2.1.

- [16] 教育出版, “H32 小学算数系統表.indd ”, [https://www.kyoiku-shuppan.co.jp/2020shou/sansu/pdf/sansu4\\_keitou.pdf](https://www.kyoiku-shuppan.co.jp/2020shou/sansu/pdf/sansu4_keitou.pdf), 閲覧日 2025.2.1.
- [17] 文部科学省, “年間指導計画の作成”, 今、求められる力を高める総合的な学習の時間の展開 (中学校編), pp.62-68, 2022.
- [18] みんなの教育技術, “「スタディ・ログ」とは? 【知っておきたい教育用語】 ”, <https://kyoiku.sho.jp/122126/>, 閲覧日 2025.2.2.
- [19] DNP, “蓄積された学習履歴 (スタディ・ログ) を見とり、よりきめ細かい指導を実現 大阪市 3 年間のスマートスクール事業の成果から ”, [https://www.dnp.co.jp/biz/case/detail/10157721\\_1641.html](https://www.dnp.co.jp/biz/case/detail/10157721_1641.html), 閲覧日 2025.2.2.
- [20] 超個別指導まがつく, “積み上げ式学習 ”, <https://www.matsugaku.co.jp/stocked/>, 閲覧日 2025.2.2.
- [21] 櫻井研介, “学びの系統性・連続性を踏まえた学習指導 - 算数・数学科のレディネステストを活用して - ”, ,16, pp.49-54, 2018.
- [22] ベネッセ教育情報, “苦手トップ2は「数学」と「英語」! 苦手科目克服のために最適な勉強法とその理由とは? ”, <https://benesse.jp/kyouiku/201907/20190712-2.html>, 閲覧日 2025.2.2.
- [23] 富田塾, “第 32 回 大きく 2 つに分かれる教科の特性 ”, <https://www.s-lab-tomita.com/cont/column/no32tumi.html>, 閲覧日 2025.2.2.
- [24] 家庭教師のマスター, “【第 89 回】 次の学年に「苦手」を持ち越さない! 3 月からでも間に合う「さかのぼり学習」のコツ ”, <https://www.u-master.net/blog/2022/03/03/>, 閲覧日 2025.2.2.
- [25] note, “さかのぼり学習用教材 ”, <https://note.com/miraie2017/n/n9aaffbc368ed>, 閲覧日 2025.2.1.
- [26] 学び場 A to Z, “無学年式オンライン教材「すらら」とは? 教材、学習方法、料金、他教材との違いや「こんな子どもにおすすめ!」など徹底解説 ”, <https://p-cure.net/sulala/>, 閲覧日 2025.2.2.
- [27] 個別指導 Axis, “一人ひとりの目標から逆算した学習提案 ”, <https://axis-kobetsu.jp/attractions/planning/>, 閲覧日 2025.2.2.
- [28] jooto, “クリティカルパスとは? 使い方や求め方を解説 ”, <https://www.jooto.com/contents/critical-path/>, 閲覧日 2025.2.2.
- [29] smartsheet, “クリティカル パス法への究極のガイド ”, <https://jp.smartsheet.com/critical-path-method>, 閲覧日 2025.2.2.

- [30] ビズクロ, “プロジェクト管理の PERT 図とは？構成要素や書き方・注意点をわかりやすく解説”, <https://bizx-elb.chatwork.com/project-management/pert-pm/>, 閲覧日 2025.2.2.
- [31] 中小企業診断士試験 一発合格道場, “プロジェクト管理素人でも解ける！アローダイアグラム集中講座 by はっしー”, <https://rmc-oden.com/blog/archives/179724>, 閲覧日 2025.2.2.
- [32] ものづくりドットコム, “「PERT / CPM」とは”, [https://www.monodukuri.com/gihou/article\\_list/96/](https://www.monodukuri.com/gihou/article_list/96/), 閲覧日 2025.2.2.
- [33] Lychee, “【初心者必見】クリティカルパスとは？求め方・見つけ方の詳細を解説！”, <https://lychee-redmine.jp/blogs/project/tips-criticalpath/>, 閲覧日 2025.2.2.
- [34] asana, “クリティカルパスとは？プロジェクト管理に役立つ手法を解説”, <https://asana.com/ja/resources/critical-path-method>, 閲覧日 2025.2.2.
- [35] “Critical Path Method in Python — Early Time, Late Time, Critical Activities and Project Duration”, Math Hands-On with Python, 2021.02.26, 00:00:00–00:10:14, YouTube, [https://www.youtube.com/watch?v=I6MLKMirZ2E&ab\\_channel=MathHands-OnwithPython](https://www.youtube.com/watch?v=I6MLKMirZ2E&ab_channel=MathHands-OnwithPython)
- [36] ワークマネジメントオンライン, “クリティカルパスの書き方は？具体的な使い方や求め方”, <https://www.work-management.jp/blog/how-to-write-critical-path.html>, 閲覧日 2025.2.2.
- [37] 蓮池隆, “CPM を用いた不確実・不確定状況下におけるクリティカルパスの求解”, 数理解析研究所講究録, 第 1829 巻, pp 72-79, 2013.
- [38] 中市新太, “効率的な遡り学習も考慮できる目標逆算型積み上げ式学習支援システム”, 富山県立大学学士論文, 2024.
- [39] M. F. EL-Santawy and S. Abd-Allah, “The Longest Path Problem in Fuzzy Project Networks : A Case”, *Gen. Math. Notes*, Vol . 3, No. 2, pp. 97-107, 2011.
- [40] asana, “対話型 AI とは？導入して得られるメリットや課題について解説”, <https://cognigy.tdse.jp/conversational-ai/what-is-ai-chatbot/>, 閲覧日 2025.2.2.
- [41] GMO, “RESEARCH リッカート尺度とは？メリットと設問作成時の注意点を解説”, <https://gmo-research.jp/research-column/likert-scale>, 閲覧日 2025.2.2.
- [42] 北海道大学オープンコースウェア, “ファジィ推論”, <https://ocw.hokudai.ac.jp/wp-content/uploads/2016/01/IntelligentInformationProcessing-2005-Note-09.pdf>, 閲覧日 2025.2.2.