

鉄道システム開発における OR の適用と課題

高橋 理

ユビキタス技術の発展により鉄道システム開発分野におけるシステム化領域も拡大しつつあるが、その多くは従来人間が機械的にやっていた作業を計算機で置き換えるものであり、人間が考える作業のシステム化はまだ始まったばかりである。本稿では、この「人間が考える作業のシステム化」に関する OR の適用事例として、乗務員運用計画の変更案を組合せ問題として定式化したうえでメタ戦略を用いて作成するケースと、列車運行管理システムの開発工程における改善の必要性を AHP によって定量評価するケースを紹介するとともに、その実用化や普及にあたっての課題について述べる。

キーワード：乗務員運用計画，スケジューリング，ソフトウェア開発プロセス，AHP

1. はじめに

日本の鉄道は、正確性の高さでも安全性の高さでも世界に広く知られているが、その実現の中核になっているのが列車運行管理システムである。列車運行管理システムは管轄するエリア内にあるすべての列車をあらかじめ定めたダイヤ通りに走行させる役割をもち、駅のポイントや信号機を制御する機能、乗客に運行情報を提供する旅客案内機能、運行乱れ時に列車ダイヤや車両・乗務員の運用スケジュールを復旧する運転整理機能など、さまざまな機能をもっている。

筆者はこの列車運行管理システムの開発に携わっており、大学時代に OR を学んだ経験を生かして、列車運行管理システムの開発に OR の適用を試みている。本稿では、2つの適用事例を通して、OR 適用の成果とその実用化や普及にあたっての課題について述べる。

2. 鉄道乗務員運用計画の変更案作成

計算機を用いた列車運行管理システムの運用は急速に広がっているが、列車運行の元になる運行計画（列車ダイヤのほか、検査・清掃計画にしたがって車両運用順序を決める車両運用計画、運転士や車掌の乗務スケジュールを決める乗務員運用計画、車両基地における作業順序を定める構内作業計画など）[1]は意外にもいまだに熟練者の手作業で作成されている。一部では計算機の導入も始まっているが、計算機は人手で作成した計画やその矛盾（制約条件違反）を画面に

表示したり、矛盾解消のための編集機能を提供するなど、人手による作成業務を支援しているにすぎない。

しかしながら、これまで列車運行計画の作成業務を担ってきた熟練社員が近年減少しているほか、列車の高密度化や鉄道事業者間での相互乗り入れなどにより運行計画はますます複雑化している。特に、災害や事故などによって列車運行が乱れた場合には、輸送混乱の拡大を防ぐために短時間での運行計画の変更が求められる。近年のコンピュータ科学の発展にともなって運行計画の自動作成システムへの期待が高まっている。

本節では、必要性の高い運行計画作成業務の1つとして、列車運行乱れ時における乗務員運用計画の変更案自動作成手法[2]について紹介する。

2.1 乗務員運用計画の変更案作成問題

図1は列車ダイヤを示したもので、縦軸に路線上の駅、横軸に時刻を配し、斜線は列車の運行状態を表す。また、斜線のうちの太線部分は、ある乗務員が勤務開始時刻から勤務終了時刻までの間に行う列車乗務スケジュールを示すもので「行路」と呼ぶ。このとき、乗務員運用計画の変更案作成とは、災害や事故などによって列車運行が乱れた場合に、乱れに合わせて修正さ

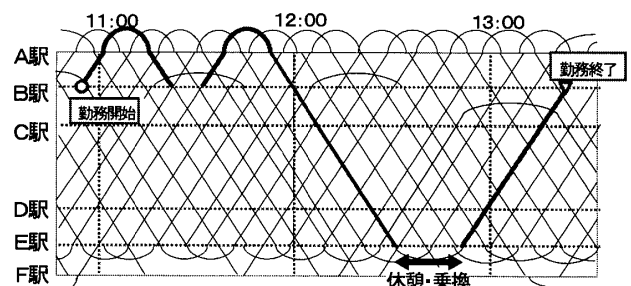


図1 列車ダイヤと行路の例

たかはし さとる
三菱電機(株) 先端技術総合研究所
〒661-8661 尼崎市塚口本町 8-1-1

れた列車ダイヤに基づき、乗務員の労働条件や移動上の物理制約などの制約条件を満たすように、すべての列車を行路に割り当て直す問題である。

次に、乗務員運用の基本計画作成においては、乗務員の労働条件を厳密に加味することが要求されるが、列車運行乱れ時における乗務員運用計画の変更案作成においては、労働条件の充足よりも列車運行の早期復旧が優先される。このため、現状では変更案作成には明確な制約条件が設定されておらず、労働条件を超過する変更案が作成されることも多かった。

そこで、本節に述べる変更案の自動作成手法では、基本計画作成時と同じ制約条件として、表1に示す制約条件をできるだけ満たす変更案の作成を行う。

ここで、乗務員運用計画作成の解法として集合被覆問題の利用が知られている。これは、表1に示した制約条件を満たす範囲で、列車ダイヤ上にある任意の複数本の列車をまとめたものを行路候補と呼ぶとき、すべての列車をいずれかの行路候補がカバーするような行路候補の組合せのうち、コストを最小化する組合せを求めるものである。

しかし、この解法によって現実的な時間で実用に耐えうる解を得るのは困難である。もちろんどんな計画でもよければ作成することは容易であり、例えばすべての行路候補を選べば1つの解となるが、行路数(=乗務員数)が極めて多くなり、非効率な計画になる。また、ある列車が複数の行路に重複している場合には、いずれか1つの行路だけが当該列車への乗務を担当し、残りの行路は便乗扱いにする(車内業務は行わず、次列車始発駅への移動手段として利用する)ことが考え

られるが、列車乗務と便乗では労働条件(例えば労働時間の計算方法)が異なる場合もあるので単純ではない。

2.2 変更案自動作成手法の開発

前述したように、変更案作成においては短時間のうちにすべての列車を行路に再割当することが必要である。そこで、筆者らは、乱れに応じて修正された列車ダイヤ(運転整理ダイヤ)に対して乗務員運用の修正前計画をあてはめたときに最初に制約違反が発生する時刻以降を再割当の対象とし、比較的弱い(やむを得ない場合には違反も許容される)制約条件を除いて初期解を作成した後、すべての制約条件を満たす解をメタ戦略にしたがって探索する手法を開発した。

手法の詳細については文献[2]を参照していただくことにし、以下ではメタ戦略による解探索を行う際にキーとなる①初期解の作成、②近傍探索の方法について述べる。

① 初期解の作成

すべての行路について、最初に制約違反が発生する時刻以降を白紙に戻し(計画として割り当てられていた列車を消去し)、未割当の状態になった列車を始発時刻の早いものから順に、拘束時間上下限と最終行先地を除く制約条件を満たす行路に再割当する。

このとき、まず割当先としてふさわしい行路の候補を表2の割当ルールにしたがって抽出したうえで、表3に示す選択知識を優先順位の高い順に適用して1つに絞り込む手順をとる。

なお、制約条件を満たす行路が1つも存在しない場

表1 制約条件の例

項目	内容
乗務員の人数	あらかじめ定められた人数以下(=出勤人数以下)にする
未充当の解消	乗務員の割り当てられていない列車がないようにする
乗り継ぎの一致	ある列車の終了駅と次に乗務する列車の開始駅を一致させる
食事時間	食事時間帯に必要な休憩時間を確保する
一連続乗務時間	休憩なしで列車に連続乗務する時間を上限値以下にする
拘束時間上下限	勤務終了時刻の計画からのずれを閾値以下にする
最終行先地	勤務終了駅を計画と一致させる

表2 割当ルールの例

ルール名	内容
空き行路割当	列車が割り当てられていない行路
連続乗務割当	休憩なしで乗り継げる行路
休憩最適割当	30分程度の休憩をとれる行路
計画行路割当	計画上で割り当てられていた行路
最短食事割当	食事時間をぎりぎり確保する行路

表3 選択知識の優先順位例

順位	知識
1	計画行路に戻ることができる
2	決められた折返ルールに対応する
3	拘束時間が計画に達していない
4	食事休憩を確保する
5	連続乗務が継続する

合には、当該列車の前に別の列車への便乗をはさむことによって割当可能となる行路の探索を行い、便乗を用いてもなお割当先が見つからない場合には乗務区所で待機する予備乗務員に割り当てる。

② 近傍探索の方法

初期解作成では考慮しなかった制約条件における違反の程度を評価値とする。また、ある解に対して、1つの列車を別の行路に移動するか、2つの行路間で列車を入れ換えたものを近傍とし、解の近傍を探索して得られる最良解への移動を繰り返す。最良解が改悪となる場合も一定の終了条件を満たすまでは移動させる。

2.3 実際の路線への適用

実際の路線（駅数 39、列車本数 750）で発生した運行乱れ 6 ケースに対して本手法を適用した結果を表 4 に示す。なお、本路線には 162 種類の行路があり、乗務員は 2 つの乗務区所に分かれて所属している。表に示すように、いずれのケースにおいてもすべての列車を行路に割り当てることができ、弱い制約違反の残った 1 ケースを除き、すべての制約違反を解消する変更案を作成することができた。

一方、本手法の実用化にあたっては、現在業務に従事している熟練者の評価を得ることが不可欠である。そこで、熟練者が作成した結果との比較やヒアリングを行ったところ、以下の課題が明らかになった。

① 目的関数や制約条件が明確でない

熟練者にとっての理想的な変更案（本問題の最適解）の定義があいまいであるため、目的関数を定義することが難しい。定性的には、乗務員の残業時間が少ない方がよい、早期に乱れを収束する方がよい、乗務員が乱れの影響を公平に被る方がよいなどといった考え方があがるが、これらを多目的関数として定式化する場合に、どのような重み付けをするべきかは明確でない。

表 4 運行乱れに対する変更案の自動作成結果

試験 No.	適用前 違反数	列車 割当率	適用後 違反数	計画 比率	処理 時間
1	24	100%	0	96%	21 秒
2	47	100%	0	95%	22 秒
3	64	100%	1	88%	22 秒
4	19	100%	0	99%	15 秒
5	29	100%	0	99%	15 秒
6	40	100%	0	99%	16 秒

動作環境： CPU：PA8500-400MHz，OS：HP-UX10.20

い。

特に、乗務員の労働環境を確保するためにも変更前の計画から大きく変わっても制約条件を満たしたいという意見がある一方で、乗務員の混乱を避けるために多少の制約違反はあっても変更量は小さい方がよいという意見もあり、熟練者の間でもコンセンサスがとれていないのが現状である。

このような状況下では、目的関数や制約条件を対話的あるいは状況に応じて動的に変化させながら試行錯誤することが必要であり、目的関数や制約条件を容易に変更できる手法が有用であると思われる。

② 必ず解を出すことが要求される

計算機による変更案は作成結果だけが出力される形になるので、熟練者にとってしばしば変更箇所や変更理由が分かりにくい。そのため、計算機が熟練者の満足する解を出していない場合には、かえって解修正に手間がかかることから、計算機には常に満足できる解を出すことが要求される。

一方、乗務員数の少ない夜間などには制約充足解が存在しない場合もあるため、解がないことを短時間に得られる仕組みがあるとよいと思われる。また、そういった場合には、制約条件を緩和する（制約違反を許容する）ことが考えられるが、その緩和の仕方や重み付けも難しい問題の 1 つである。

③ 計算機と人間の役割分担

計算機は与えられた問題を解くことは得意であるが、問題を先読みすることは不得意である。列車ダイヤ乱れの状況は時々刻々と変化することが多く、熟練者は将来何が起こりうるかを予測しながら変更案を作成しており、例えば、臨時列車が増えることを予想してあらかじめ手を打ったり、（今の状況では必要なくても）あらかじめ非番の乗務員を呼び出している。こうした経験に基づく作業を計算機でどう実現するか、あるいは計算機と人間の役割分担をどう分けて対応するかといった検討も、実用化にあたっての大きな課題である。

3. AHP を用いたソフトウェア開発プロセスの改善

前述したように、列車運行管理システムは多くの機能をもつ大規模なシステムであり、路線によっては数百人もの開発者が数年がかりで開発することもある。

こうしたシステム開発においては、開発プロセスの改善によって開発効率（生産性）を向上させることが必要である。この開発プロセスの改善では、開発者に

新たな作業を強いる場合もあるため、トップダウンで活動を推し進めることが必要とされるが、近年、改善活動の成否は、実際の活動を行う開発者の現状プロセスに対する満足度に大きく依存し、現場の開発者が不満に感じている作業に対する改善ほど活動の効果が大きく、開発者に不満のない作業に対する変更を行うと、かえって開発効率が低下することが分かってきた[3].

そこで、本節では、多目的意思決定手法の1つであるAHP[4]を用いて、列車運行管理システムのソフトウェア開発プロセスに対して開発者が考える改善の必要性（不満度）を定量化する手法[5]について紹介する。

3.1 ソフトウェア開発プロセス改善

開発プロセスの評価は、第3者による客観評価や国際規格などに定められている標準モデルとの比較によって行うことが多く、さらなる改善が必要な箇所に対して教育やツール整備などの改善策を立案・実行する。

CMM[6]は、ソフトウェア開発プロセス評価モデルの1つであり、組織の開発プロセスの成熟度を5段階に分け、各段階で確立すべき達成項目や具体的な活動内容を記している。各組織はこれにしたがった改善活動を実施することによって、開発レベルをステップアップさせることができるとされている。

ただし、この改善活動にはリソース（費用・人・時間など）を要するだけでなく、劇的な変更は開発者の混乱を招く可能性もあるため、改善を実施する作業を決めて行うのが通常である。

3.2 AHPを用いた改善必要性評価

本節では、開発者の主観評価を定量化する手法として階層化意思決定法（AHP）の改良モデルであるDescriptive AHP[7]を用い、開発者が最も改善の必要性があると考えられる開発工程を抽出する。

AHPは、評価構造を評価基準や代替案からなる階層構造に表現したうえで、各項目の一对比較によって、それぞれの相対的重要度を決定する意思決定手法の1つである。人の主観を定量化する手法としては、多属性効用理論や関連マトリクス法などもあるが、AHPはこれらの手法に比べて実施が容易で手順が理解しやすく、定性的なデータを扱うことができるという特徴がある。また、Descriptive AHPは、AHPにおいて代替案の追加や削除によって選好順位が入れ替わるといった矛盾を解消するモデルの1つであるが、各評価基準の下での満足・不満足境界を示す希求水準を設定することを特徴とし、その結果、代替案の重要度を満

足度に帰着させることができるという利点を持っている。

(1) 階層構造

多くの開発組織では、設計・製作・試験・保守の順に作業を分割して進めるウォーターフォール型の開発工程が採用されている。そこで、評価対象である代替案にはこれら4つの開発工程を想定し、評価基準にはCMMのレベル2に定義されている評価項目および活動内容を書き下して図2のように配置した。

なお、図を見やすくするため、最下層にあるすべての評価基準から各代替案に伸びる枝は省いている。

(2) 評価基準の一对比較

同じ階層にある評価基準対について、開発者に改善の必要性の比を表5から選択させ、対応する値を割り当てた一对比較行列を作成する。

また、行列の主固有ベクトルについて成分和を1とする正規化を行い、評価基準の重要度とする。

(3) 希求水準の想定

開発者が各評価基準における要求を達成するために必要だと考える最低限の作業内容を希求水準として想定してもらう。

例えば、評価基準の1つである「仕様書のレビュー」は仕様書に対する品質を確保する作業であり、仕様書のすべての記述に対してチェックを行い、不備があれば書き直す。この作業は、理想的には「すべての仕様書」の「すべてのページ」を「不備がなくなる」まで繰り返す必要があるが、実際には作業時間などを考慮し、例えば「すべての仕様書ではなく、特に重要な仕様書だけでいいのではないか」「不備がなくなるまでではなく、5件以下になるまでいいのではないか」などと条件を緩和して考えている場合があり、このように開発者が緩和した条件が希求水準である。

この希求水準は開発者によって異なり、実際の開発環境が希求水準より好ましい状態であれば、この開発

表5 一对比較における選択肢

選択肢	値
改善の必要性がとても強い	7
改善の必要性が強い	5
改善の必要性がやや強い	3
同程度に改善の必要性がある	1
改善の必要性がやや弱い	1/3
改善の必要性が弱い	1/5
改善の必要性がとても弱い	1/7

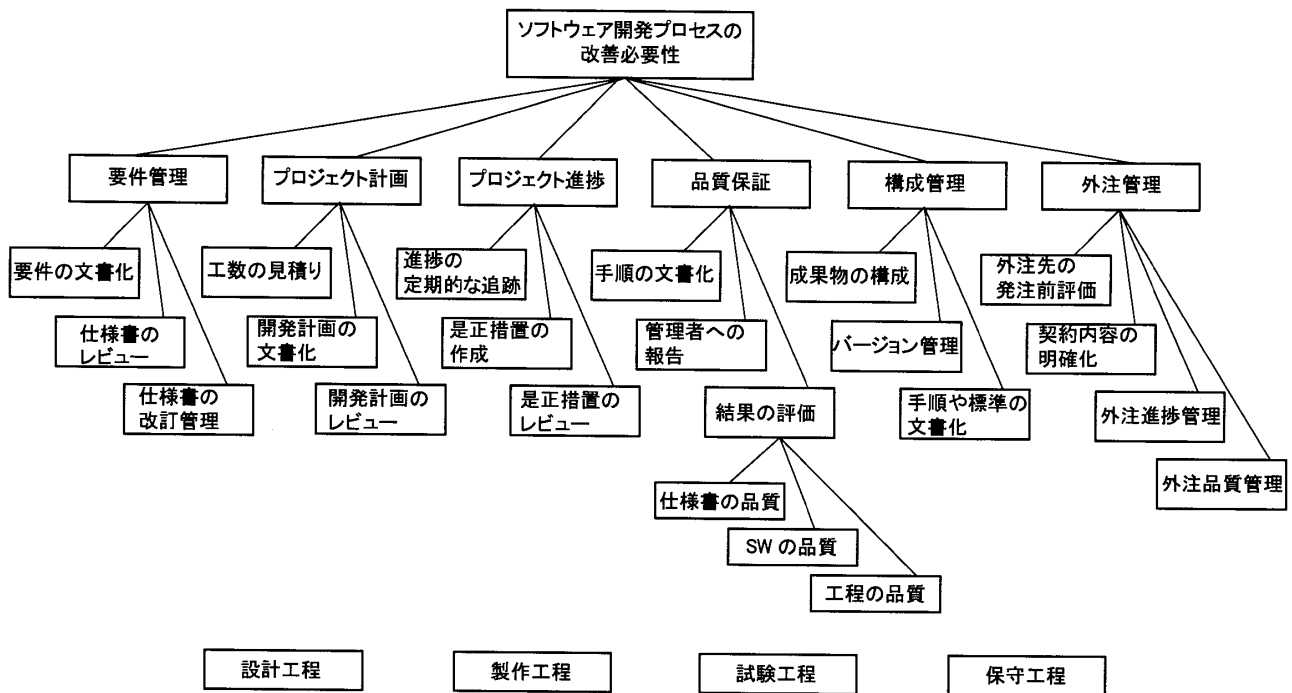


図2 ソフトウェア開発プロセスの改善必要性評価を行うための階層構造

者は現状に満足しており、実際の環境が希求水準に満たなければ開発者は不満をもっていることになる。

(4) 代替案の一対比較

各評価基準の下で4つの代替案に希求水準を加えた計5つの要素について一対比較を行う。また、一対比較行列の主固有ベクトルについて、希求水準の重要度を1とする正規化を行って、代替案の重要度とする。

(5) 総合重要度の計算

加法系統合ルール（重みつき和）によって代替案の総合重要度を得る。

このとき、すべての評価基準のもとで希求水準と同等の評価をもつ代替案の総合重要度は1になることから、総合重要度が1を超える開発工程には総じて改善の必要性を感じており、1未満の開発工程には全体としてすでに満足しているとみなすことができる。

3.3 実際の開発チームへの適用

本手法で必要な一対比較をアンケートにまとめ、列車運行管理システム開発メンバーのうち、すべての工程を横断的に見る立場にあるリーダーらを被験者として回答を依頼した。

表6は、ある被験者への適用結果として、各代替案における第1層の評価基準の下での重要度と総合重要度を示しており、この被験者は製作工程と試験工程に対する改善の必要性を強く感じていると言える。

次に、被験者から回答を得る過程や評価結果に対する説明の際にヒアリングを行ったところ、以下に示す

表6 開発者Aの評価結果

	要件管理	P J 計画	P J 進捗	品質保証	構成管理	外注管理	総合
	0.23	0.17	0.08	0.39	0.05	0.08	1.00
設計	1.31	1.42	4.38	2.25	1.00	2.40	2.02
製作	3.00	2.28	1.57	4.90	4.38	2.26	3.50
試験	2.36	3.80	5.22	4.65	2.69	3.08	3.81
保守	1.62	2.69	1.27	2.25	2.69	0.72	2.00

意見や疑問点を得た。

① 本手法が妥当な評価方法なのか分からない

被験者の中には手法の中身を問うことなく、機械的に回答をこなす人もいたが、手法が妥当なものであるかが分からなければ時間を割いて回答する意味がないといった人もいた。そこで、シンプルな問題を例にAHPを説明して感覚的に理解してもらった。

② 一対比較回数が多くて回答してもらえない

AHPを用いると良く似た質問が多く、回答項目数も多いため、前向きに回答してもらえない場合もあった。そこで、回答が難しい場合には無理に回答を要求せず空けておくことを許容し、不完全一対比較行列からの重要度算出法[8]を利用した。

③ 前の回答を意識してしまう

被験者はよく似た質問が出ると、以前の回答を思い出して、整合性の取れた回答をしようとする傾向があ

った。これを防止し、できるだけ正直な評価を聞きだすため、質問項目をランダムに並び替える、1問あたりの回答時間を制限する、質問とは関係のない雑談を挟んで、前の回答を忘れてもらうなどの処置を行った。

④ 結果が正しいのか分からない

得られた結果が実態を反映したものであることが分からなければ、結果を改善活動に用いることはできないという意見があった。そこで、被験者には開発工程(代替案)について直接に順位付けしてもらった結果と比較することで結果の妥当性を理解してもらった。

また、代替案の重要度について、数値自体が大きな意味をもつものではないが、数値が大きいほど改善の必要性の度合いが強いことを説明したところ、開発者間で議論をする際の材料として用いられ、合意形成に役立つことが分かった。

⑤ 評価を行う側の負担

前述したように、被験者への回答依頼にあたっては、手法の妥当性説明や質問方法の工夫に相当の時間を要した。このため、被験者からは有効な回答を得られた一方で、評価を行う側の負担は大きくなり、特に多くの被験者を相手にする場合などにはより簡便な評価手法の開発が望まれる。

4. おわりに

ユビキタス技術の発展により鉄道システム開発分野におけるシステム化領域も拡大しつつあるが、その多くは従来人間が機械的にやっていた作業を計算機で置き換えるものであり、人間が考える作業のシステム化はまだ始まったばかりである。

本稿では、この「人間が考える作業のシステム化」

に関するORの適用事例として、乗務員運用計画の変更作成と列車運行管理システム開発プロセスの改善必要性評価について紹介した。ORの適用により、従来は定性的あるいは人間の主観で決められていた作業を数学的にモデル化して定量的に解析することができるようになる。しかし、人間の主観をいかに負荷なくモデル化するか、そしてその妥当性をどのようにして示すかについては今後の課題である。

参考文献

- [1] 勸鉄道総合技術研究所, 鉄道のスケジューリングアルゴリズム, NTS, 2005.
- [2] 高橋理, 片岡健司, 鉄道分野におけるソフトコンピューティングの適用事例, 電気学会全国大会講演論文集, 2006.
- [3] G. Yamamura, Process Improvement Satisfies Employees, IEEE Software, Vol. 16, No. 5, 83-85, 1999.
- [4] 木下栄蔵 編著, AHPの理論と実際, 日科技連出版, 2000.
- [5] 高橋理, 駒谷喜代俊, 田村坦之, 階層化意思決定法を用いたソフトウェア開発プロセスの改善必要性評価, システム制御情報学会論文誌, Vol. 16, No. 7, 2003.
- [6] M. C. Paulk et al., Capability Maturity Model for Software, Ver. 1.1, SEI-93-TR 024, 1993.
- [7] 田村坦之, 高橋理, 鳩野逸生, 馬野元秀, 階層化意思決定法(AHP)の記述的モデルの提案と選好順位逆転現象の整合的解釈, Journal of the Operations Research Society of Japan, Vol. 41, 214-227, 1998.
- [8] P. T. Harker, Alternative modes of questioning in the analytical hierarchy process, Mathematical Modeling, Vol. 9, 353-360, 1987.