

CAMP: 順序づけ分枝限定アルゴリズム 設計支援システム

関口 恭毅

1. はじめに

オペレーションズ・リサーチは意思決定を支援する情報活動であり、その高品質化や効率化のためにはコンピュータや通信技術の応用を含む広い意味での情報環境を総合的に整備することが期待される。そのような総合的な情報環境を以下では**ORの計算環境**と呼ぶ。

オペレーションズ・リサーチの過程全体を対象として考えるとき、モデルを構築しその効率的な解法を開発して高性能で使いやすいコンピュータ・プログラムを開発するという伝統的なアプローチの他にも、ORの計算環境を整備するさまざまなアプローチが考えられる。オペレーションズ・リサーチの実践や研究を通して蓄積されるモデル、解法、ノウハウなどの知識を体系的に整備し、その活用を容易にするのも、そのようなアプローチの1つである。たとえば、European Journal of Operations Research では1989年から無料ないし実費程度で入手可能なORのソフトウェアの紹介記事を連載して優良ソフトウェアの交換を支援している（これまでに紹介されたソフトウェアを整理した資料に[3]がある）。

Grunwald-Fortuin[1]は、専門化した多様な分野におけるORワーカーの専門知識の集成にもとづいた「ORエキスパート・システム」の構築を提案した。もちろん、事典・教科書・事例集の刊行やサーベイ論文の作成はこのアプローチの伝統的な手法といえる。Müller-Merbach[4]は近似解法構築のノウハウを整理した。本論文で紹介するCAMP (Computer Assisted design of Mathematical Programming algorithms) は解法やその構築に役立つ知識のデータベース・システムである。

CAMP が対象にするのは順序づけ問題である。この分野は汎用的な解法がなく、問題ごとに効率的な解法を構築することが必要である。解法としては分枝限定法を

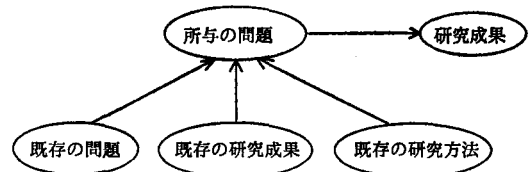


図1 解法構築における知識の相互関係

対象にしている。よく知られているように分枝限定法は列挙型解法の汎用的枠組みであり、解析的解、動的計画法なども含まれる[5]。

以下では、CAMPの概要を、その依って立つアルゴリズム設計作業のモデル、知識構造、機能、マン・マンのインターフェース、利用事例の順で紹介する。

2. 解法アルゴリズム設計作業モデル

現実の意思決定問題をオペレーションズ・リサーチによって解決する場合、作成されるモデルとその解法アルゴリズムは強く依存しあう。つまり、高効率の使いやすい解法アルゴリズムを設計できるかどうかは作成するモデルに左右され、逆に、作成されるモデルの有効性は効率的な解法アルゴリズムの開発の可否に依存する。しかし、CAMPではモデル化された順序づけ問題が所与であると仮定する。

与えられた順序づけ問題を解くアルゴリズムを設計する状況を考える(図1)。設計者は所与の問題に利用可能な解法がすでに存在するかどうかを調べる。もし存在しても改善が必要な場合もある。存在しなければ、解法構築に有効な研究成果が存在するかどうかを調べる。その場合、所与の問題を直接扱った研究ばかりでなく類似の問題を扱った研究の解法を参考にする。時には、研究推進上のノウハウを集めることもあろう。調査の範囲は、設計者の頭の中だけであるかも知れないし、さまざまな文献を読破するかも知れない。このような知識や情報の相互関係の中で行なわれる解法構築作業に対してどのような支援環境が必要かについては[6]に詳述した。

順序づけ問題は典型的な組合せ最適化問題であり、原理的には完全列挙法によって常に解くことができる。課題は効率の改善である。分枝限定法を特定の問題に適用するには、その問題向きに具体的内容を定める必要がある。完全列挙法は自明な分枝限定法であり、通常分枝限定法はなんらかの方法でその効率を改善したものと解釈できる。そこでCAMPでは以下の4段階からなる解法構築の手順を想定した。

第1段階(問題の記述)：所与の問題がCAMPになれば、CAMPのスタイルに合わせて記述し入力する。

第2段階(初期アルゴリズム構築)：CAMPに既存の解法があればそれを、なければ完全列挙法を構築してそれを初期アルゴリズムとする。

第3段階(問題・アルゴリズムの分析)：CAMPの知識を検索利用し、(類似の)問題や分枝限定法を本格的に分析して初期アルゴリズムの改善に有効な性質や方法を発見し、CAMPに入力する。

第4段階(アルゴリズムの構築)：第3段階の結果にもとづいてアルゴリズムに改善をほどこす。結果をCAMPに追加する。

以上の4段階は決して逐次的なものではなく相互に行き来する複雑な経過をたどると考えられる。

3. CAMPの知識構造

CAMPはIBM PC-AT上で第4世代言語であるMainstayを利用して開発された。約80枚の作業画面、約90個の手続きからなる。本節と引き続き2節においてその概要を紹介する。その詳細は[7][9]を参照されたい。

CAMPは図2に楕円で示した各種のファイルないしデータベースを有する。問題、アルゴリズム、ノウハウ、それらの引用文献などのデータベースである。設計者はこれらのデータベースの内容だけで既存の成果のエッセ

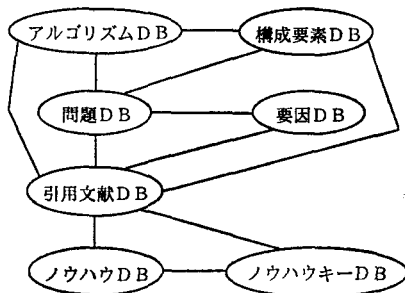


図2 CAMPの知識ベース

ンスを知ることができる。それらはデータと呼ぶよりは知識あるいは情報と呼ぶにふさわしく、CAMPは知識ベースをもつといてよい。しかし、いわゆる自動推論は一切せず、創造的な仕事はすべて設計者に依存する。文献データベースは証明方法、事例、関連研究などのさらに突っ込んだ調査のために利用する。

これらの知識の入力に際して、効率的な検索が可能ないように用語を十分に標準化することが必要であり、一方、知識の具体的な内容を曖昧さなく正確に記述しなければならない。こうした相反する要求を満たすために、CAMPでは項目-詳細アプローチが採用された。

●問題：各順序づけ問題は、一方では問題の特徴づけの27個の要因(factor)の組合せとして項目記述され、他方でその詳細を任意の長さのテキストで記述する。要因は、順序づけの対象である仕事(job)に関するもの、仕事を処理する工場(shop, machine)に関するもの、目的関数に関するものに分類される。要因は要因データベースに蓄えられ、個々の問題の記述にはそこから検索して利用する。

●アルゴリズム：分枝限定法は分枝する問題の選択操作、限定操作などのいくつかの構成要素(constituent)の組合せとして項目記述することが可能である。構成要素への分け方は関口[5]に依った。各構成要素は構成要素データベースに蓄積される。個々の分枝限定法は構成要素データベースから検索された構成要素の組合せとして記述されるとともに任意の長さのテキスト(あるいは構造化した一定形式の記述)でその詳細が記述される。

●ノウハウ：解法構築に有益な知識であれば何であっても任意の長さのテキストとしてこのデータベースに入力できる。現状では次の6つのカテゴリーに分類されている。

一般規則	分枝限定法構築の一般的知識
改良	分枝限定法改善のためのノウハウ
等価性	問題相互間の等価的關係
近似性	問題相互間の近似關係
数理モデル	問題の数理的性質
定義	関連する専門用語の解説

ノウハウはこのカテゴリー名にノウハウの適用対象となる構成要素・分枝限定法の性能尺度・問題特性を加えた4項目(これらをノウハウキーと呼ぶ)によっても記述される。ノウハウキーは検索する際に検索キーとして使われる。ノウハウキーは必要に応じてノウハウキー・データベースに追加され、ノウハウを入力する際にはそ

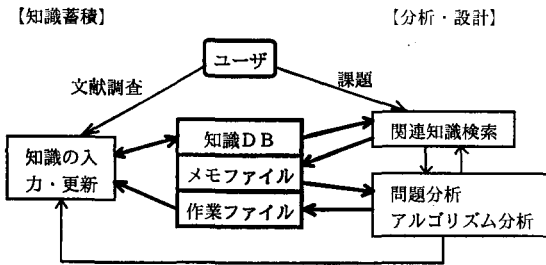


図3 CAMPの機能概要

こから検索して利用する。

これらのデータベースは、図2に線で示したような相互の参照関係を有し、柔軟な情報検索が可能である。たとえばアルゴリズムは構成要素の組合せ、問題、引用文献によって検索できる。逆に各アルゴリズムから、それに関連する構成要素、問題、引用文献を検索することができる。

4. CAMPの機能

CAMPの機能は3種類に大別される(図3□部分)。

第1のものは、データベースの知識の入力・更新機能である。多様で相互に関連する知識の効率的な操作が可能のように、検索はもちろん入力・更新手順もきわめて柔軟に作られている([5]参照)。問題やアルゴリズムなどの知識本体の詳細記述であるテキスト入力にはフルスクリーン・エディタ機能が提供される。

第2は、知識の検索機能である。知識は、現に課題となっている問題とその解法ばかりでなく、項目記述の各項目を検索キーとして利用することで関連問題、関連問題の性質や解法アルゴリズムなども効率よく検索できる。問題の分析や解法の構築は、高度に知的な作業であり、検索した知識を多面的に検討し活用できることが必要である。その便宜のために、検索した知識をメモファイルと呼ぶ保存用ファイルに必要な期間保持する機能がある。

第3は、検索した知識を活用するなどして問題を分析しアルゴリズムを構築するための支援機能である。具体的にはフルスクリーン・エディタを利用し、分析や構築の作業記録などを作業ファイルに作成する。

CAMPの有効性はその知識ベースが充実しているか否かにかかっている。知識の陳腐化を防ぐためには常に新しい知識を補充することが必要であり、図3の左半分に示したように、利用者は日常的に文献その他を通じて獲得する知識をCAMPに入力して知識ベースの充実に

努めなければならない。また、CAMPを利用して得る研究成果も入力される必要がある。これは第2節で述べたアルゴリズム設計手順の第4段階である。各知識の詳細記述の入力に相当の労力を要するのが難点である。図3の右側は第1-3段階においてはCAMPの知識ベース、メモファイル、作業ファイルが利用されることを示している。

5. CAMPの対話構造

CAMPの各機能は階層化したメニューから選択する。第2階層の各選択肢は作業画面を表示する。図4は順序づけ問題検索のための作業画面の1つである。

どの作業画面も4つの部分から成る。最上部の2行はシステム・メッセージの表示欄である。下から2行目はこの作業画面で利用可能な機能に対応する機能キーとともに示す。たとえば、F2のFactorInputはカーソル位置に入力すべき要因が、要因データベースの中に存在しない場合にそれを追加入力する機能である。F6は必要なら引用文献を変更する機能である。F10はどの画面でもその画面を終了して1つ前の画面にもどる。どの作業画面でも各機能キーは類似の機能を起動するように配慮されている。

各作業画面から、さまざまな機能を起動できることがCAMPを柔軟なシステムにしている。たとえば、アルゴリズムを入力する途中で新しい構成要素の入力を起動(図5のF1キー)し、そのために、さらに引用文献を入力するなどである。このようにメニューの階層を深く入り込むことがあり、メニューの中で迷子になる恐れがある。現にいる作業画面に到達するまでに選択してきた各階層のメニューや作業画面の選択肢の系列をメニュー経路と呼ぶ。作業画面の最下行はメニュー経路の表示欄である。図4では、メインメニューから知識入力、問題入力、既存データの変更による入力、要因による検索、検索結果の修正の各機能をこの順に選択してきたことを示している。

これらの行を除く中間の領域はこの画面における作業領域であり、図4の場合、問題記述のための要因の組合せを所与の問題に合うように修正することができる。

6. CAMPによる作業例[8]

3機械のフローショップ問題に対するアルゴリズムの構築例を紹介する。

第1段階：CAMPを検索すると、いわゆる Johnson

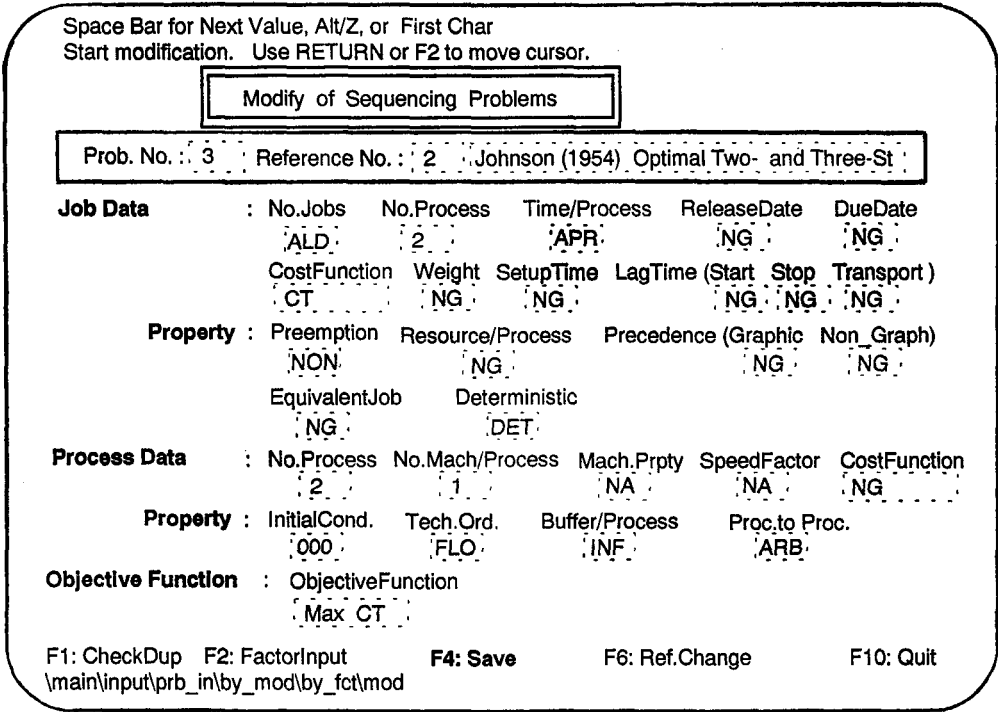


図 4 作業画面 (問題入力の場合)

の2機械問題[2]が含まれているので、これを修正して
 入力した。図4はその途中の2機械問題が検索された時
 の作業画面の様子を示したものである。

第2段階：アルゴリズムの概築機能を選択し、構成要
 素を選択してアルゴリズムを新規に構築するモードを選
 ぶ。各構成要素に対して、構成要素データベースから適

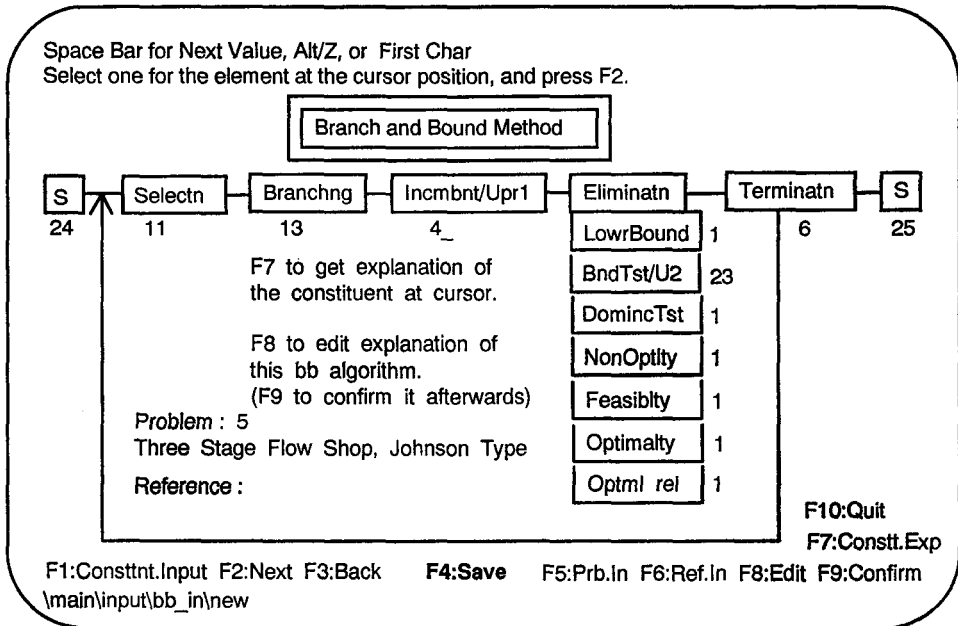


図 5 完全列挙法の構築

当な構成要素を選んで完全列挙法のアルゴリズムを構成する(図5)。図5は、組み合わせた各構成要素の番号を示している。カーソルをF2, F3キーで任意の構成要素欄に移動すれば、そこで構成要素データベースにある該当の構成要素の一覧を見ることができ、F7キーで各構成要素の詳細記述を読むことができる。また、アルゴリズムが解く問題(F5)や引用文献(F6)を入力する機能を起動できる。F8でデフォルトとして各構成要素の説明文を機械的につないだだけの説明文をつくり、アルゴリズムの詳細記述を入力する作業画面に移行できる。

第3段階：データ検索のモードを選択し、既存のデータの中から参考になりそうなものを選択してメモファイルに記憶する。ノウハウを検索すると62番目(m 機械問題と遅れ時間つき2機械問題が等価になるための条件)と61番目(遅れ時間つき2機械問題を利用すれば最大完了時刻の下界値を計算できること)がある。そこで問題データベースでフローショップ問題を再度検索すると問題の15番に遅れ時間(輸送遅れ)つきの2機械フローショップ問題が登録されている。これを検索すれば、輸送遅れつきの問題の完了時刻の計算式がわかる。さらに、この問題に関するノウハウを検索したが、1つも発見できなかった。そこで、この問題の解法アルゴリズムの構成要素を検索したところ、選択則と最適性テストの2つが発見されたので、最適性テストの内容を読むと輸送遅れつき2機械フローショップ問題に対するJohnsonの最適順序づけ規則が判明した(Johnsonの規則であることは、この構成要素の引用文献から推定できる)。これらの知識を活用して、新しい下界値の計算方法を考案することができた。

第4段階：新しい下界値を利用する界値テストを第2段階で作ったアルゴリズムに追加すればよい。また、選択則は深さ優先型であるが、第2優先順位の選択基準として最良界値型を追加する。

さらなる改善が必要な場合には再度第3段階を実行すればよい。この例の場合、たとえば、優越テストを導入するなどの新たな改善方法を見つけることができる。

7. ま と め

順序づけ問題に対する分枝限定法を設計するのに有益な知識をデータベース化し、これを効率よく活用するための知識支援システムであるCAMPの概要を紹介した。CAMPの利用例はこのような知識支援システムが

オペレーションズ・リサーチの支援環境として有効であることを示している。

CAMPの思想は他の問題分野やアルゴリズム分野にも応用できるであろうし、知識の内容や表現方法を加減すればCAI(Computer Assisted Instruction)にも利用できるであろう。

参 考 文 献

- [1] Grunwald, H. J. and Portuin, L.: DSS and ES in the "information organization"——Back to the roots of OR," *European Journal of Operational Research*, Vol.41 (1989) 142-150.
- [2] Johnson, S. M.: Optimal two-and three-stage production schedules with setup times included. *Naval Research Logistics Quarterly*, Vol.1, No.1 (1954), 61-68.
- [3] ORSEP-Operations Research Software Exchange Program. 日本オペレーションズ・リサーチ学会研究部会「ORの計算環境」配布資料(1992年7月).
- [4] Müller-Merbach, H.: Heuristics and their design: a survey. *European Journal of Operational Research*, Vol.8, No.1 (1981), 1-23.
- [5] 関口恭毅：多段決定過程の設計技術に関する研究の現状。電気学会システム制御研究会資料 SC-78-7 (1978, 7, 21 北海道大学), 1-10.
- [6] 関口恭毅：順序づけアルゴリズム設計者を支援するワークベンチの開発＝基本構想＝。北海道大学経済学部ディスカッションペーパー, シリーズB, No.4 (1991)
- [7] 関口恭毅：順序づけアルゴリズム設計者を支援するワークベンチの開発＝CAMP システムの概要＝。北海道大学経済学部ディスカッションペーパー, シリーズB, No.5 (1991)
- [8] 関口恭毅：順序づけアルゴリズム設計者を支援するワークベンチの開発＝CAMP システムの試用と評価＝。北海道大学経済学部ディスカッションペーパー, シリーズB, No.6 (1991)
- [9] 関口恭毅：順序づけアルゴリズム設計者を支援するワークベンチの開発＝CAMP システム利用マニュアル＝。北海道大学経済学部ディスカッションペーパー, シリーズB, No.7 (1991)