

ジョブ割付問題のオブジェクト指向問題解決部品ライブラリー

5 J-6

濱 利行 堀 雅洋 中村 佑一

日本アイ・ビー・エム株式会社 東京基礎研究所

1.はじめに

ソフトウェアの開発方法論として、オブジェクト指向の有効性は広く認められるところである。オブジェクト指向では、抽象データ型に基づく情報隠蔽とクラスの継承機能がその有効性の要因である。しかしながら、実際のアルゴリズムの設計は容易ではない。そこで、近年では、特定の領域におけるプログラムの典型的な構造を再利用して、ソフトウェアの生産性を向上させようとする提案がなされている。このような構造は、フレームワーク[1]、コントラクト[2]などと呼ばれ、オブジェクト指向の枠組みの中ではメッセージの呼び出し関係(プロトコル)を抽象的なレベルで固定し、そのプロトコルを満たす様々なオブジェクトを入れ替えることにより、個々のアプリケーションに対応しようというものである。確かに、アプリケーションの分野をうまく選択することで、実用的にかなりの効果をあげているが、分野によっては柔軟性に欠けるために適用が難しい面もある。

我々は、これまで、スケジューリング問題の一つであるジョブ割付問題の問題解決方法を数種類の部品からの合成によって構成する方法について考察を行ってきた[3]。スケジューリング問題は、問題の性質に合わせて解法を選ぶ必要があり、フレームワークの適用が難しい分野である。そこで、メッセージの呼び出し関係にあたる制御を抽象化することによりフレームワークに柔軟性を加え、より広い分野に適用できるようにし、ジョブ割付問題の部品ライブラリーの実現に適用した。

2.ジョブ割付問題と問題解決部品

我々が定義したジョブ割付問題とは、ガントチャー

トに代表されるような資源と時間の2軸からなる2次元の平面に、ある時間帯を占有するジョブを制約を満たしながら割り付けて行く問題である。この問題のクラスに属する問題を解くエキスパートシステムを分析し、問題解決部品を抽出した結果、次の3つの部品カテゴリーを得た。

- ・ Divide and Merge
- ・ Transform and Restore
- ・ Check and Modify

また、これらの部品が操作の対象とするジョブ、資源、時区間の3つの操作対象の性質を洗いだした。操作対象は幾つかの属性によって特徴づけが可能なので、オブジェクト指向のクラスとして自然に定義できる。しかし、このようなスケジューリング問題は、ほとんどがNP完全あるいは困難なクラスに属するので、一般には問題の特徴をいかしたヒューリスティクスを使うことになり、同じカテゴリーに属する問題解決部品でも、その制御の流れは多様である。そこで、いわゆるフレームワークにあたるものを見つけることは難しい。

3.制御の抽象化

ジョブ割付問題をもう少し一般化して探索問題としてみると、N個の変数(ジョブ)がある問題であれば、N次元の空間を探索空間としてその中から制約を満たすか、ある評価関数を最適にする1点を見つける問題であると言える。探索空間の中から解となる1点を見つける方法としては、次の2つが考えられる。

- ・ 探索空間を分割して絞りこみながら解を発見する。
- ・ 探索空間内の適当な1点からはじめて、解へ向かって移動させて行く。

前節の部品カテゴリーでは、前者がDivide and Mergeに、後者がCheck and Modifyに対応する。前者の場合、空間を分割するといっても系統的に分割することになり、いわゆる木探索がこれにあたり、深さ優先、幅優先、最良優先などの戦略によって制御に違

Object-Oriented Libraries for Solving Job Assignment Problems

Toshiyuki Hama, Masahiro Hori, Yuichi Nakamura

IBM Research Tokyo Research Laboratory

1623-14 Shimotsuruma, Yamato, Kanagawa 242, Japan

いが出てくる。後者には、線形計画のシンプレックス法、遺伝アルゴリズム、Simulated Annealing法などが含まれるが、これらの解法の違いは、点を移動させる規則が異なるだけなので、制御はほとんど同じである。そこで、本論文では前者について詳しく述べる。

木探索の問題解決では、アルゴリズムを特徴づける制御戦略として、次の3つが考えられる。

- ・ 探索木のどの節を延ばして行くか (Node Ordering)
- ・ 各節で何を基準に探索空間を分割して新しい節を生成するか (Variable Ordering)
- ・ 各節で分割された空間のどれを新しい節とするか (Value Ordering)

この3つの制御戦略を抽象クラスとして定義することにより、木探索の問題解決方法は1つのフレームワークとして表現可能である。これにより、問題毎に制御のクラスを具体化して調整することによる問題解決アルゴリズムの合成が可能となる。

具体的には、探索木のレベルで見ると、問題解決はNodeオブジェクトとNode Orderingオブジェクトから構成され、Node Orderingオブジェクトは新しく生成されたNodeオブジェクトを受け取り、次に枝を延ばすべきNodeオブジェクトに探索空間分割のメッセージを送る。Nodeオブジェクトは、探索空間を表現するSpaceオブジェクト、Variable Orderingオブジェクト、Value Orderingオブジェクトから構成され、探索空間分割のメッセージを受け取ると、Variable Orderingオブジェクトに分割の基準となる変数を問い合わせ、Value Orderingオブジェクトに変数の値を問い合わせ、縮小された探索空間とともに新しいNodeオブジェクトを生成する。このように、探索木、節に対応するオブジェクトのメソッドは制御オブジェクトと個体オブジェクトから構成され、アルゴリズムの構成が簡素化し、見通しがよくなる。

4. 操作対象の抽象化

前節では制御の抽象化について述べたが、探索の操作対象の抽象化も部品ライブラリーを実現する上で重要な要件である。ジョブ割付問題の操作対象となるジョブ、資源、時区間は属性の集合として定義できるので、クラスの継承機能を用いて容易に抽象クラスにまとめることができる。しかし、探索空間は、

これらの操作対象の集合として表現されるのであるが、時区間のように単純に個体の集合として表現できない場合もある。一般的に探索問題で言えば、変数の取りうる値が連続的であったり、無限集合の場合には、 $\{x \mid 0 < x < 10\}$ のように内包的な表現を用いて集合を定義することになる。そこで、内包集合と外延集合を統一的に扱えると便利である。

これは、制約のクラスの実現とも関係が深い。制約のクラスは個々の変数への値割り当てをチェックするだけでなく、制約伝播の処理では値の集合から制約を満たす部分集合を選択するフィルターの機能を持つ必要がある。外延集合であれば、チェック機能だけでフィルターを構成できるが、内包集合を対象とすると内包表現を直接操作できる機能が必要である。ジョブ割付問題では、時区間の集合の表現をこのような観点から、離散的な場合と連続的な場合を統一的に表現し、同じ枠組みでのアルゴリズムの構成を可能としている。

5. 終りに

本論文では、操作対象の抽象化と制御の抽象化によりジョブ割付問題の問題解決アルゴリズムを一つのフレームワークで実現するオブジェクト指向のライブラリーについて述べた。様々な探索アルゴリズムをフレームワークの具体化で実現することで、アルゴリズムの構成を理解しやすくし、問題に合わせたアルゴリズムの調整も容易にできるようになる。問題点は、一般化された枠組みでアルゴリズムを構成するので実行効率が多少低下することである。構成されたアルゴリズムに最適のデータ構造を提供するにはアルゴリズムのデザイン（仕様）と実現を分離する必要があり今後の課題である。

参考文献

- [1] J. M. Vlissides and M. A. Linton, "Unidraw: A Framework for Building Domain-Specific Graphical Editors", ACM Trans. on Information Systems, Vol 8, No.3, July 1990, Pages 237-268
- [2] I. M. Holland, "Specifying reusable components using Contracts", EOOP '92, Pages 287-308
- [3] M. Hori, Y. Nakamura and T. Hama, "Methodology for Configuring Scheduling Engines with Task-Specific Components", JKAW '92, Pages 215-229