

制約充足問題解決手法を用いたスケジューリングシステムの一提案

5 J-5

山内 学 今成 文明 小川 均
立命館大学理工学部情報学科

1 はじめに

スケジューリング問題は物流分野における日々の原料・製品の配送スケジューリングや、生産分野における生産スケジューリングなど、日々定常的に問題の解決が要求されている問題であり、その自動化による効果は大きい。

一方、人工知能の様々な問題を制約充足問題の特別な場合として定式化できることが知られている[1]。そこで、スケジューリング問題を制約充足問題として定式化し、制約充足アルゴリズムを用いることにより解を得ることができる。

しかし、一般にスケジューリング問題は規模が大きいため、探索空間が広くなり解を見つけ出すために非常に多くの時間を要する。また、短期的なスケジューリングを必要とする場合は、時間をかけて最適なスケジューリングを行なうよりも、望まれる時間内で解として受け入れられる近似解を得ることが要求されると思われる。そしてまた、スケジューリング問題では全ての制約を満たす解を得られるとは限らない。

そこで、本研究では、この様なスケジューリングをするために、制約を種類分けすることにより解として受け入れられる近似解を得る方法について提案を行い、また、その有効について検証を行う。

2 制約充足問題

制約充足問題の特別な場合として、人工知能の様々な問題を定式化できることが知られており、これまでに多くの実験的、理論的な研究がなされている。

一般に制約充足問題は次の様に定式化される。n個の変数 X_1, X_2, \dots, X_n と、変数のそれぞれが値をとる有限で離散的な領域 D_1, D_2, \dots, D_n 、および変数間の取り得る組合せを定義する制約の集合が存在

する。そして、この制約を満たす変数の組合せを求める問題が制約充足問題である。

制約充足問題を解く標準的なアプローチとしてバックトラックアルゴリズムがある。また、できるだけ多くの制約を満たす解を求める制約充足問題の解法として、branch and bound 法 [2] と heuristic-repair 法 [3] がある。

バックトラックアルゴリズムは、ある順序で変数を順番に具体化することによって、領域Dを調べるものである。

一方、バックトラックアルゴリズムを応用した branch and bound 法は、その時点での解の制約違反数を枝を刈る閾値とし、全ての制約を全て満たす解が存在しない場合でも、できるだけ多くの制約を満たす解を見つけだすことができる方法である。

heuristic-repair 法は、全ての変数を具体化した後、全体としての制約違反数が減少するように、変数の値を順番に繰り返して変更する方法である。

3 スケジューリング問題

スケジューリング問題にはさまざまなタイプがあるが、その一例としてゼミ割当て問題を考える。この問題は、“曜日”、“時限”という軸で表現される2次元平面の時間割表上に、“ゼミ”をそれぞれ別の所に割り当てる問題である(図1)。

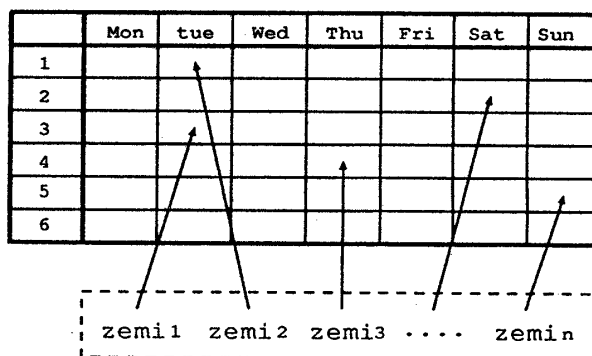


図 1: ゼミ割当て問題

A Method for the Scheduling System using Constraint Satisfaction Algorithm

Manabu Yamauchi, Fumiaki Imanari, Hitoshi Ogawa
Department of Computer Sciene, Faculty of Science and Engineering, Ritsumeikan University

また、各ゼミには複数の構成員が属しており、各構成員は複数のゼミに属している。そして、各構成員はゼミを割り当てるのにあたって次のような要求を持っているものとする。

- ・ 参加できないコマがある。
- ・ できれば参加したくないコマがある。
- ・ 3連続でゼミに参加したくない。

etc..

ここで、まず、各ゼミを変数 $\{Zemi_1, \dots, Zemi_n\}$ と考え、時間割上のコマを各ゼミ変数の領域を $\{(mon, 1), \dots, (sun, 6)\}$ と考え、各構成員の要求をその変数の制約とすることにより制約充足問題としゼミ割り当て問題を定式化することができる。

同様に、一般的なスケジューリング問題についても、変数、変数の領域、各変数の関係として問題を表現することによって制約充足問題として定式化することができる。

ところが、先程述べた様に、スケジューリング問題は一般に探索空間が広いため、非常に多くの時間を必要とする。また、この様なスケジューリング問題では、全ての制約を満たすことが不可能な場合も生じる。

そこで、本研究では制約を種類分けすることにより、この問題を解決するアプローチをとる。つまり、解として絶対に満たさなければならない絶対的制約と、できれば満たしたい希望的制約に分ける。そして、絶対的制約を満たす解をまず求め、解として受け入れられる近似解を求める。その後、希望的制約を満たしていくことにより、解を完全なものに近づけていく。このような方法を用いることにより、迅速なスケジューリングを必要とする場合にも近似解を得ることができ、また、十分な時間を与えれば完全な解に近づけることができる。

4 実験と結果

制約を種類分けすることによって近似解を得る有効性を検証するために、まず、バックトラックアルゴリズムを用いて絶対的制約を満たす近似解を求め、その後、希望的制約を満たす解法として

- (1) branch and bound 法
- (2) heuristic-repair 法

を用いた実験を行なった。得られた結果をそれぞれ図2の(1)と(2)に示す。横軸は時間を示し、縦軸は絶対的制約を満たした後の希望的制約違反数を示す。

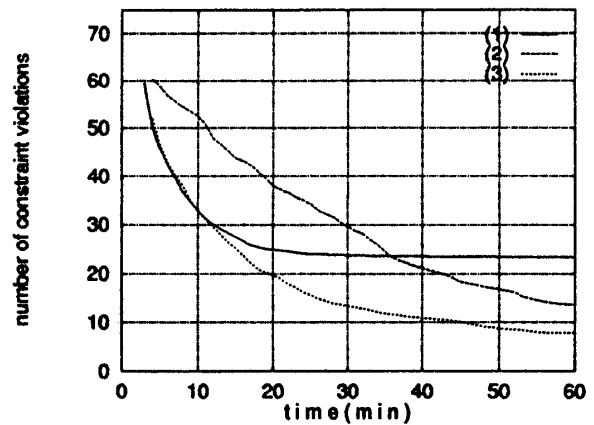


図2: 実験結果

図2から(2)は、時間とともに制約違反数がほぼ一定に減少し、(1)とはその応答性が異なる。また、初期段階においては、(1)の効率が良いが、その後は、(2)の効率が上回る。それゆえ、比較的短時間で解を必要とする場合は、(1)が有効であり、そうでないときは、(2)が有効的であることが分かる。

このような実験結果を下に、(1)を用いて希望的制約違反数を30以下にした後、(2)を用いる実験を行なった。その結果を図2の(3)に示す。この結果より(1)と(2)の特徴が活かされ、効率良く制約違反数が減少していることが分かる。

また、制約を種類分けせずバックトラックアルゴリズムを用いて6時間実験を行なったが解を得ることができなかった、これらの結果より、制約を種類分けすることによって近似解を得る有効性を確認することができた。

5 おわりに

本研究では、制約充足アルゴリズムをスケジューリング問題に適用し、短時間で解を必要とするようなスケジューリング問題に対して、近似解を得る方法を提案し、実験によりその有効性を確認した。今後の課題としては、各解法の応答性の解析や、問題のパラメータ設定による影響が挙げられる。

参考文献

- [1] Vipin Kumar: Algorithms for Constraint Satisfaction Problems A Survey, AI MAGAZINE SPRING, pp. 32-44(1992).
- [2] Eugen C.Freuder, Richard j. Wallace: Partial constraint satisfaction, Artificial Intelligence 58, pp. 21-70(1992). pp. 89-98(1992).
- [3] 横尾真; “実時間制約充足問題とその解法”, MACC'93, pp. 1-12(1993).