

局所探索と再加熱を伴う焼きなまし法による 大学時間割問題について

福島 誠*

Makoto FUKUSHIMA

Local Search and Simulated Annealing with Reheating Schedules for the University Timetabling Problems

あらまし 大学の時間割問題の解法として、焼きなまし法 (Simulated Annealing) に局所探索法 (Local Search) を導入した効果について検討し、さらに局所解からの脱出のための焼きなまし法の温度の再加熱 (Reheating) 効果についても報告している。時間割問題は、ITC (International Timetabling Competition) で公開されている例題 (Instance) の一部を採用した。結果として、焼きなまし法に局所探索法を組み入れることで、効率よく実行可能な時間割の解が得られることがわかったが、再加熱の効果については特に顕著な効果は認められなかった。また、最適な時間割を得るための結果を、2002年度のITC参加者の結果と比較すると、全問題数の半数についての比較ではあるが、最良値の場合では上位6位程度の結果になることがわかった。

Abstract: A method for solving the university timetabling problems by simulated annealing (SA) with local search algorithms and reheating schedules for escaping local minima is presented. Half of the university timetabling instances presented by ITC (International Timetabling Competition) are adopted for evaluating our algorithm. From the experimental results, it is found that SA with local search is effective to obtain a feasible timetable that satisfies the hard constraints of the timetabling, but the reheating schedules are not so effective. Although only half of 20 instances have been used in our experiment, we have compared our results with those of the participants of ITC on the reduction of the penalties of the soft constraints and found that our best result has placed in 6th among the results of ITC2002 participants.

【キーワード：大学時間割編成，焼きなまし法，局所探索法，再加熱】

【Keywords : University Timetabling, Simulated Annealing, Local Search, Reheating】

1. まえがき

時間割問題の解法については多くのアルゴリズムが提案されており^[1]、これらのアルゴリズムのなかで、Metaheuristic法に分類されるアルゴリズムについての性能比較についても報告されている^[2]。焼きなまし法 (SA: Simulated Annealing) もこのMetaheuristic法に分類されているアルゴリズムであるが、時間割問題に対してはかなり有効なアルゴリズムであることも確認されている^[3]。但し、時間割問題は多様的で、特にどのアルゴリズムが決定的に有利であるとの結論は得ることは難しく、特定分野の問題に限っての比較検討しかできないのが現状である。大学時間割問題に関しても各大学により時間割の設定条件が異なっているので比較が難しいが、一定の設定条件下で作成された問題がCompetitionの形式で公開されており、これに各種のアルゴリズムを適用して、一定の計算時間内で解くことにより、その優劣を評価検討しようという試みが行われている。本報告でも、この公開されている問題の中からいくつかを選択し、SAと局所探索法 (LS: Local Search) を組み合わせたアルゴリズムを適用した結果を、SAの再加熱 (RH: Reheating) の効果も含めて報告する。

2. 時間割モデル

ここで採用する大学時間割問題は、ITC^[4] (International Timetabling Competition) で公開されている問題の一部を採用する。このITCにおける時間割編成条件は、時間割が実行可能な条件 Hard Constraints (HC) と、時間割をより有効にする条件 Soft Constraints (SC) に分類して提示されている。HCは以下のように定義されている (はコスト名でHC, SCの各条件の違反数を表す)。

* 島根大学教育学部人間生活環境教育講座

1. 学生は、同時に2個以上の授業 (Event) に出席できない. EC
2. 教室 (Room) は授業の収容人数 RC, 授業で要求される特性 (Feature) を満足しなければならない. FC
3. それぞれの教室には同時に1つの授業しか割り当てられない.

また, SC は,

1. その日の最終時限に授業がある. LC
2. 1日に連続して3個以上の授業がある. TC
3. 1日に1個の授業しかない. OC

という3条件で、これらの条件を回避することがより良い時間割の条件とされている。HCは必ず満足することが最低条件で、SCの違反数をどれだけ制限された計算時間内で低下させることができるかでそのアルゴリズムが評価される。例題 (Instance) は、20題+1題 (Medium) がダウンロード可能で、それぞれ、Event, Room, Featureの数が異なるが、Timeslot数は1日9時間、週5日の45で固定されている。また、仕様の異なるコンピュータでの計算時間を評価するためのベンチマークが用意されている。

上記のHC, SCを評価するために、時間割編成の最低条件であるHCの違反数をコスト (Cost) C_H , 時間割の最適化のためのSCの違反数をコスト C_S として、それぞれコスト値として表す。 C_H では、HCの条件3であるEventのTimeslotへの配置の衝突を回避するのは容易なので、これはコストとして考慮しない。EventのTimeslot配置による学生の授業時間の衝突コストをEC, 教室のサイズの不適合によるコストをRC, Eventの教室への要求仕様の不適合によるコストをFCとして設定し、これらの条件に違反する毎にコスト $C_H (=EC+RC+FC)$ に1を加える。即ち、 C_H をゼロにすることがHCの満足条件となる。同様に、SCの3条件のコスト値をそれぞれ、LC, TC, OCとして、その合計値を C_S とする。アルゴリズムの評価は、HCを満足させるのに必要な計算時間 (Run Time) と、SCのコスト値の合計 C_S の値で評価する。

3. 初期配置とSAによる実行可能な時間割の編成

3.1 初期配置

Eventを各教室のTimeslotにランダムに配置するのではなく、FCとRCをゼロにするように配置する。これにはEventの人数と、Featureを調べ、これを各教室の収容人数と、Featureに適合するように配置しなければならない。そこで、Eventに適合する複数の教室をまず記録し、その中から人数の余裕度を調べて余裕度の少ない順にEventを配置する。この初期配置でFCとRCをゼロにしておいて、以下に示すSA (SA_HARD)を実行可能な時間割を編成する ($C_H=0$) ために適用する。この初期配置に要するRun Timeは約0.3秒程度で、10回の実行結果の中央値を3.3のSA_HARDの実験結果とともに後の表1に示す。

3.2 SAと局所探索によるEventの配置 (SA_HARD)

初期配置後に適用するSAはECのみをゼロにすることを目的とするが、この場合FCとRCが1以上にならないようにSAのアルゴリズムを修正する必要がある。このため、LSを適用する場合を除いて同一の教室内でEventの入れ替えをランダムに行う。また、SAの基本的な処理にはAbramson^[3]らの提案したアルゴリズムを採用する。これをSA_HARD/SOFT1/SOFT2として以下に示す。(SA_SOFT1, SA_SOFT2はこの後の4.で C_S 低減のために使用する場合のSAで、それぞれ、LSとRHの適用が異なる)

SA_HARD/SOFT1/SOFT2:

```
Set Cooling Rate    =0.99
Set initial cost    C0
Set initial temperature T0
n=0
```

```
Outer_Loop: Do Until Outer_Loop_Termination_Condition
MOVE_COUNT=0
```

```
Inner_Loop: Do While MOVE_COUNT < MAX_MOVE
MOVE_COUNT =MOVE_COUNT +1
Move Events( see Table 4 )
```

```

Calculate Cost  $C_n$ 
 $C=C_n - C_{n-1}$ 
IF  $C \leq 0$  THEN
    Exit Inner_Loop
ELSE
    Calculate probability:  $P( C ) = \exp( - C/T_n )$ 
    Generate random number:  $RND ( 0 \leq RND < 1 )$ 
    IF  $RND \leq P( C )$  THEN
        Exit Inner_Loop
    ELSE
        Restore
    END IF
END IF
Inner_Loop_End
IF Reheating_Option=True THEN REHEATING_HARD/SOFT2
IF Local_Search_Option=True THEN LOCAL_SEARCH_HARD/SOFT2
 $T_{n+1} = \alpha \cdot T_n$  ( GEO_COO:Geometric Cooling )
n=n+1
Outer_Loop_END
Outer_Loop_Termination_Condition:
SA_HARD :  $C_H=0$ , SA_SOFT1 :  $n \geq 100$ , SA_SOFT2 : Run Time  $\geq$  MAX_RUN_TIME

```

ここで、 α は冷却率、 T_0 は初期温度、MAX_RUN_TIME は Run Time の制限値、MAX_MOVE は Event の最大交換回数で、それぞれ $\alpha=0.99$ 、 $T_0=1$ 、MAX_RUN_TIME=900秒、MAX_MOVE=100に設定した。通常、SA の初期温度は文献[5]にあるように決定するが、ここでは初期配置で FC、RC を既にゼロにしていること、計算時間が制約されていることなどから低く設定した。 C はコストの変化量で、この値を基に冷却スケジュールを決定する。即ち、 $C > 0$ の場合には $P(C) = \exp(- C/T)$ の確率に依存して入れ替えを受け入れるかどうか決定する。MAX_RUN_TIME の900秒は、ITC のベンチマークプログラムの測定から最大許容 Run Time (Allowed Run Time) が942秒となったので (WINXP/SP2 with AMD OPTERON242 2CPUs at 1.6GHz)、この値を基に Run Time の制限値として設定した。Outer_Loop_Termination_Condition は、SA_HARD/SOFT1/SOFT2 における Outer_Loop の終了条件をそれぞれ示す。REHEATING_HARD/SOFT2、LOCAL_SEARCH_HARD/SOFT2 はそれぞれ、温度の再加熱、局所探索処理ルーチンで、 C_H を低減するためには x_HARD ルーチンが、4.2 の C_H 低減のための SA_SOFT2 では x_SOFT2 ルーチンがそれぞれ実行される。以下に REHEATING_HARD についての処理の概要を示すが、これらの再加熱のアルゴリズムについては文献[6]で提案されている方法を参考にし、GEO_RH、NON_MON、ENH_RH1、ENH_RH2 の4種類の再加熱スケジュールを設定した。従って、再加熱なしで単純に冷却する (GEO_COO) 場合との合計で冷却スケジュールは5種類になる。

REHEATING_HARD:

1 . GEO_RH:Geometric Reheating

```

Set  $\alpha =$ 
Cooling:  $T_{n+1} = \alpha \cdot T_n$ 
Heating: IF Local_Optima > Fixed_Number THEN  $T_{n+1} = T_n /$ 

```

2 . NON_MON:Non_Monotonic Cooling

```

Cooling:  $T_{n+1} = \alpha \cdot T_n$ 
Heating: IF Local_Optima > Fixed_Number THEN  $T_{n+1} = \text{Max}[T_r / 2, T_b]$ 

```

3. ENH_RH1:Enhanced Geometric Reheating 1

```

Set  $\epsilon =$ 
Set  $\delta = 0.05$ 
Cooling:  $T_{n+1} = \delta \cdot T_n$ 
Heating: IF Local_Optima > Fixed_Number THEN  $T_{n+1} = T_n / \delta$  :  $m_{n+1} = m_n \cdot (\delta >> \epsilon, m_n \geq \epsilon)$ 
Reset Heating Condition: IF  $T_{n+1} \geq T_0$  THEN  $T_0, T_n = T_0$ 
IF  $m_{n+1} < 0$  THEN  $m_{n+1} =$ 

```

4. ENH_RH2:Enhanced Geometric Reheating 2

```

Cooling:  $T_{n+1} = \delta \cdot T_n$ 
Heating: IF Local_Optima > Fixed_Number THEN
    IF  $T_n < 0.01$  THEN
        IF  $C_n > 10$  THEN  $T_{n+1} = T_0$ 
        IF  $10 \geq C_n > 5$  THEN  $T_{n+1} = T_0 / 2$ 
        IF  $C_n \leq 5$  THEN  $T_{n+1} = T_0 / 4$ 
    END IF
END IF

```

Termination Condition:

```

IF  $C <> 0$  OR Local_Optima > Fixed_Number THEN Local_Optima = 0

```

ここで、Local_Optimaは $C=0$ の連続した回数で、閾値は実験より Fixed_Number=30 とした。即ち、連続して30回の入れ替えでもコスト値が変化しなければ、局所解に陥ったと判定する。ここで、Termination Conditionは局所解が終了したと判断する条件で、コスト変化が $C <> 0$ になった場合と再加熱後に設定する。NON_MONの再加熱スケジュールにおいて、 T_r は直前の温度のリセット値で初期値は T_0 、 T_b はその時点でのベストのコスト値が得られた温度である。従って、 $T_0 > T_b > T_r / 2$ ならば T_{n+1} は T_b となり、この T_{n+1} がリセット値 T_r となる。また、ENH_RH2は新たに追加した再加熱法である。次に局所探索(LS)のアルゴリズムLOCAL_SEARCH_HARDを以下に示す。

LOCAL_SEARCH_HARD:

```

IF Local_Optima > Fixed_Number THEN
    Move every EC-clashed Event Other Events
    Calculate cost change C
    IF RC=0 and FC =0 and C < 0 THEN
        Accept
    ELSE
        Restore
    END IF
END IF

```

3.3 実行可能な時間割編成の実験結果

ITCのInstance 1と4について、HC条件を満足し実行可能な結果が得られるまでに必要としたSA_HARDによるRun Time結果を図1(a),(b)に示す。実験ではプログラム言語にVisualBasic5を使用し、コンパイル時の処理速度の最適化オプションはAlias以外は全て採用した。図1では再加熱条件別に、LOCAL_SEARCH_HARDを適用した場合(WI_LS)と、適用しない場合(WO_LS)について示している。再加熱をしない場合はGEO_COOで表している。実

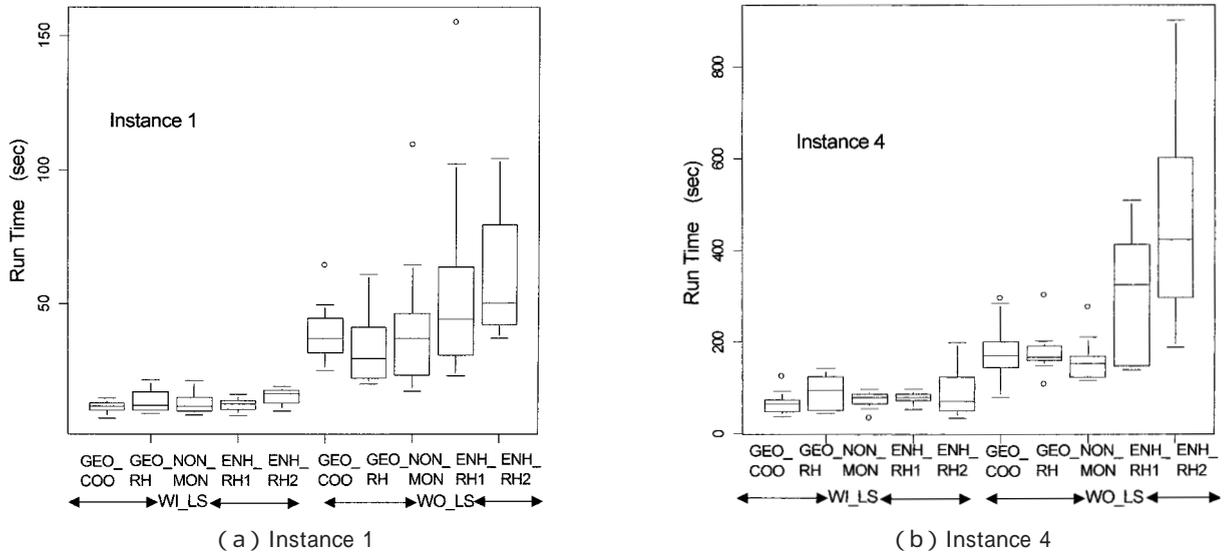


図1. SA_HARDにより実行可能な時間割を得るために必要とした計算時間(Run Time)の結果. ITCのInstance1と4について、冷却スケジュール別にLocal Searchあり(WI_LS)となし(WO_LS)の場合において、それぞれ10回の実験を行った結果をBoxplotで示す. 図中のBoxはデータの4分位点の第1と第3番目の範囲を示し、WhiskerはBox範囲の上限と下限からBox範囲の1.5倍以内での最大、最小のデータポイントを示す. Box内の水平線は中央値、外れ値は円で示す. (a) Instance 1. (b) Instance 4.

Fig. 1 Results of the run time to obtain a feasible timetable for ITC instances 1 and 4 based on 10 runs using SA_HARD for each cooling schedule. The box shows the range between the 25% and the 75% quantile of the data. The whiskers extend to the most extreme data point which is no more than 1.5 times of the interquartile range from the box. A bar in the box is the median and the circles are outliers. (a) Instance 1. (b) Instance 4.

験は各設定条件について10回の実行を行い、結果の中央値を横線で、第1と第3の4分位数の範囲をBoxで示す. WhiskerはBox範囲の上限と下限からBox範囲の1.5倍の範囲内での最大、最小のデータ値を示している. 結果から明らかなのは、LSの適用により確実に解に到達する計算時間が減少することである. 再加熱の効果はInstance 1, 4ともLSを適用した場合は再加熱による差異が少ないことがわかる. また、LSを適用しない場合は、ENH_RH2が結果が良くないことがわかる. この原因は再加熱の条件として、温度 T_n が0.01を下回った時に $C_n > 10$ の場合には再加熱して、初期温度 T_0 まで急上昇させるためと考えられる. これによりコスト値が上昇し、再度減少させるのに多くの計算時間を必要とするためと考えられる.

Instance 1から10までについて、3.1の初期配置のRun Time、及び再加熱にENH_RH2を使用しLSを適用した場合のSA_HARDのRun Timeについて、それぞれ10回の実験結果の中央値を表1に示す. ITC参加者のうち、実行可能な時間割を獲得するためのRun Timeが報告されている1位のKostuch^[7]の結果と比較すると、筆者らの結果はベンチマークで測定されたAllowed Run Timeの違いを考慮すると良い結果の様に見える. しかし、Kostuchの場合はSCの減少まで考慮したアルゴリズムを適用しているので単純な比較はできない.

表1 初期配置とSA_HARDによりInstance1-10の実行可能な時間割を得るまでのRun Timeの結果. 値は10回の実験結果の中央値で示し、SA_HARDの再加熱スケジュールはENH_RH2でLS(WI_LS)も適用している. Allowed Run Timeはベンチマークにより計測されたRun Timeの最大許容時間(秒).

Table1 Results of the run time for instances 1 to 10 for obtaining a feasible timetable. The results are the medians of 10 runs for both initial assignment and SA_HARD with reheating (ENH_RH2) and local search (WI_LS). Allowed Run Time means the values of run time in seconds measured with the benchmark program.

| Instance | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | Allowed Run Time (seconds) |
|---------------------|------|------|------|------|------|------|------|------|------|------|----------------------------|
| Initial Assignment | 0.2 | 0.2 | 0.3 | 0.3 | 0.2 | 0.2 | 0.3 | 0.3 | 0.2 | 0.2 | 942 |
| SA_HARD +ENH_RH2+LS | 16.6 | 11.3 | 20.2 | 71.1 | 60.9 | 60.1 | 30.7 | 15.0 | 15.2 | 16.5 | |

4. 時間割編成の最適化

4.1 Timeslot 単位の並び替え (SA_SOFT1)

HCを満足した時間割は実行可能な時間割であるが、時間割編成のSCを満足する最適な時間割ではない。SCの3条件を満足させるためには更にEventの並び替えが必要になる。ここでは上記の表1に示したSA_HARD+ENH_RH2+LSで得られた実行可能な時間割編成結果に対して、常にHCを満足した状態を保ったままでこの並び替えを行うことにする。これには時間割の同一Timeslot内の全Eventを、教室の配置を保ったまま異なるTimeslotの全Eventと置換する必要がある(Sequencing)。これを図2に示す。このTimeslot単位の入れ替え操作は3.2で示したSA(SA_SOFT1)によって実行するが、Move Events部分をTimeslot単位の操作に変更する必要がある。但し、この方法によるCsの改善効果は限定されることが予想されるので、Run Timeの消費を抑制するためにSAの温度更新回数nは、3.2のOuter_Loop_Termination_Conditionに示すように100回を上限に設定した。また、LSとRHは適用しないので冷却スケジュールはGEO_COOのみである。SAのパラメータMAX_MOVEは5000とした。このSA_SOFT1によるCsの改善効果を表2に示す。この表より、SA_SOFT1によるコストの改善率は約40%程度であることがわかる。

4.2 SAと局所探索による最適化(SA_SOFT2)

SA_SOFT1で満足されずに残ったSCを対象にして、各Eventを同じ教室内で入れ替えてCsを改善するために、SA(SA_SOFT2)とLS(LOCAL_SEARCH_SOFT2)を実行する。これは、基本的に3.2のSA_HARDと同じであるが、3.3

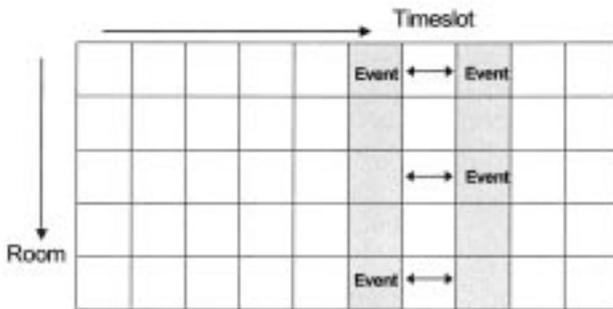


図2 Timeslotの並び替え(Sequencing)。Timeslot(陰影部分)内のEventを別のTimeslotのEventと、同一Room内で入れ替えるSA_SOFT1によるCsの低減。

Fig.2 Sequencing of the timeslots. All events are moved or exchanged between the two timeslots shown as the shaded areas in the figure for the reduction of Cs using SA_SOFT1. The room assignments of the events are not changed.

表2 表1のSA_HARD+ENH_RH2+LSの結果に対してSA_SOFT1適用後のCsのコストの改善結果(中央値)。各Instanceの下段が適用後の結果を示す。

Table2 Improved results of cost Cs after applying SA_SOFT1 for the results of SA_HARD+ENH_RH2+LS in Table1 are shown at the lower rows for each instance. The values are the medians of 10 runs.

| Cost Instance | LC | TC | OC | Cs |
|---------------|-------|-------|------|--------|
| 1 | 392.0 | 392.0 | 52.0 | 836.0 |
| | 328.0 | 195.5 | 22.5 | 546.0 |
| 2 | 391.0 | 367.0 | 62.0 | 820.0 |
| | 314.0 | 169.0 | 28.5 | 511.5 |
| 3 | 393.0 | 391.0 | 46.0 | 830.0 |
| | 322.0 | 198.5 | 19.0 | 539.5 |
| 4 | 567.5 | 538.5 | 78.5 | 1184.5 |
| | 478.5 | 270.0 | 34.0 | 782.5 |
| 5 | 593.5 | 618.5 | 77.0 | 1289.0 |
| | 477.5 | 264.5 | 31.5 | 773.5 |
| 6 | 593.5 | 587.0 | 75.0 | 1255.5 |
| | 466.0 | 273.5 | 35.0 | 774.5 |
| 7 | 681.0 | 666.0 | 96.5 | 1443.5 |
| | 492.0 | 291.5 | 42.0 | 825.5 |
| 8 | 497.5 | 471.5 | 65.0 | 1034.0 |
| | 395.0 | 217.5 | 25.0 | 637.5 |
| 9 | 424.0 | 400.5 | 65.5 | 890.0 |
| | 345.5 | 195.0 | 24.0 | 564.5 |
| 10 | 399.5 | 390.5 | 51.0 | 841.0 |
| | 323.5 | 190.5 | 23.5 | 537.5 |

のHCの結果より再加熱の方法による差異はあまり認められないことから、再加熱の方法は REHEATING_SOFT2 1種類を、LSはLCとTCを対象にした LOCAL_SEARCH_SOFT2 を行う。OCに関するLSを設定しない理由は、1日1個の授業という条件は比較的削減しやすいためである。また、 C_H の場合と異なり、 C_S をゼロにすることはこれまでの報告例からかなり困難なので、制限時間 (MAX_RUN_TIME) 内で到達できた C_S の最小値の値で結果を評価する。また、SAのパラメータは、MAX_MOVE=10000, Fixed_Number=100に設定した。再加熱のアルゴリズム REHEATING_SOFT2 と、LSのアルゴリズム LOCAL_SEARCH_SOFT2 を以下に示す。

REHEATING_SOFT2 :

```

IF LOCAL_OPTIMA > FIXED_NUMBER THEN
  IF  $C_S > C_B$  THEN
     $T_{n+1} = T_0$  (  $=4$  for  $C_B > 300$ ,  $=2$  for  $C_B \leq 300$  )
  END IF
  LOCAL_OPTIMA=0
ELSE
  LOCAL_OPTIMA=LOCAL_OPTIMA+1
END IF

```

ここで C_B はその時点での最良のコスト値である。即ち、局所解に陥った場合にその時点での最良のSCのコスト値 C_B と現在のSCのコスト値 C_S を比較し、 $C_S > C_B$ ならば C_B に依存して再加熱する温度 ($2T_0$ または $4T_0$) を決定する。LOCAL_SEARCH_SOFT2 はLS_LC とLS_TC の2種類で、以下のように、コストLC, TCのカレント値の比 LC_n / TC_n に対応して排他的に適用する。なお、LS_LC, LS_TCともEventの移動は同一Room内に限定している。

LOCAL_SEARCH_SOFT2 :

```

IF  $LC_n / TC_n > 1.1$  THEN
  LS_LC
ELSE
  LS_TC
END IF

```

LS_LC :

```

Move every Event at last timeslot of day      Other Events not at last timeslot of day
IF  $LC_{n+1} \leq LC_n$  AND EC=0 THEN
  Accept
ELSE
  Restore
END IF

```

LS_TC :

```

Move every TC-violated Event      Other Events
IF  $TC_{n+1} \leq TC_n$  AND  $LC_{n+1} \leq 1.1 \cdot LC_n$  AND EC =0 THEN
  Accept
ELSE
  Restore
END IF

```

ここで、 LC_n , TC_n はEventの移動前、 LC_{n+1} , TC_{n+1} は移動後のLC, TCのコスト値を示す。また、係数1.1は、LS_TC適用時にLCの増加許容度の上限を設定するためである。この値は実験により決定した。

4.3 SCの実験結果

上記のアルゴリズムにより, Instance1~10 について各10回実験を行った結果を図3, 図4に示す. 図3は10個の Instance について, 各実験で得られたコスト C_s の値について図1と同様な Boxplot 表示で示す. 図3から指摘できることは, Instance4, 5, 6 のように比較的難しいと思われる問題では得られた解が分散しやすい傾向にあることがわかる. また, 図4は Instance2 の C_s について, 10回の実験中の最良値 ($C_s = 92$) が得られた場合の, C_s と温度の変化の様子を SA の反復回数 (MOVE_COUNT の合計) に対して示す. この図では, SA の再加熱による温度変化と, コスト C_s の変化の様子を示しているが, 再加熱によって温度を 1 以上に急激に上昇させても, LOCAL_SEARCH_SOFT2 を併用しているため C_s の上昇は過度にならず抑制されていることがわかる. 次に, ITC の上位 7 位までの結果^[7]と比較した表を表3に示す. この表では, 筆者らの結果は10回の実験結果における最良値 (Best) と中央値 (Median) の両方の結果と比較した. ここで, ITC の結果の 'Verified' は ITC で公式に確認された結果を, 'Claimed' は参加者の報告結果である. この表より, 筆者らの結果は最良値で 6 位とまずまずの結果であるが, 中央値では12位とあまり良くない. この理由としては, SA による解法では比較的結果の値が変動することが考えられる. また, 表3よりわかることは, 上位 5~12 位はあまり差がないが, 上位 1~4 は 5~12 位に比較して明確な差をつけて優位であることである. 即ち, Instance 9 の 1 例を除いて, 上位 1~4 は全ての Instance において下位の結果よりも良いことがわかる. この理由については明確にはわからないが, ITC 公式 1 位の Kostuch の報告では C_s の低減にやはり SA を採用しているものの, 入れ替えはランダムではなく同一 Room 内の他の全 Events と入れ替える方法を採用している. さらにその後, Greedy アルゴリズムに近いような SA パラメータを使用して, より貪欲に C_s の低減を試みていることが成功したと考えられる. 従って, SA とか Tabu Search のような汎用的なアルゴリズムをそのまま使用するのではなく, 積極的に様々なパラメータの修正を行い, さらに局所探索アルゴリズム等を併用して適用するのが効果的であると考えられる. なお表3の公式結果以外にも, 公式参加とは認められていないが, 公式 1 位よりも良い結果が得られたという報告もある^[8]. 表4は SA_HARD, SA_SOFT1, SA_SOFT2 で採用した SA のパラメータと Event の移動規則の一覧表である.

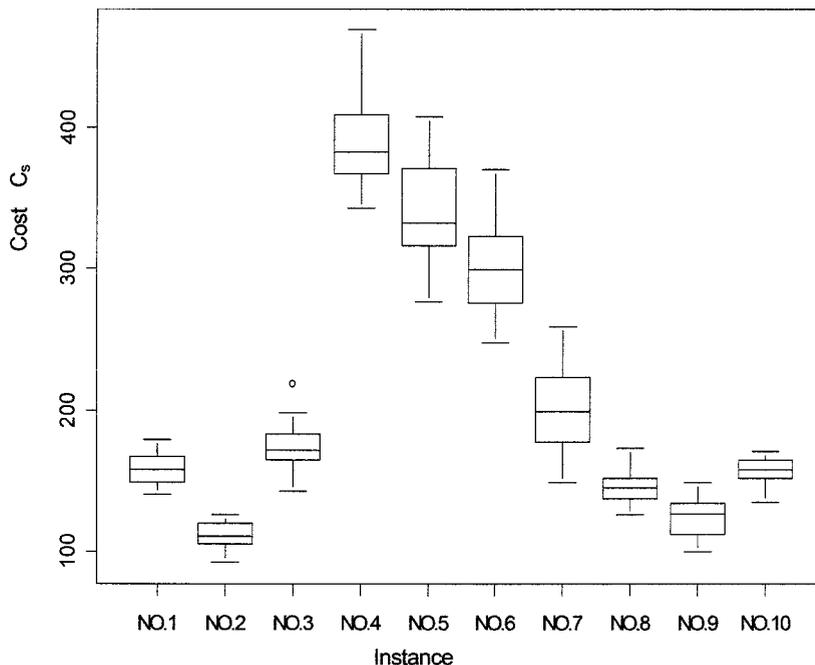


図3. SA_SOFT2により, ITC の例題 1 から 10 をそれぞれ 10 回実行した場合の, 得られた 10 個のコスト C_s の分布を Boxplot で示す. 図中の Box はデータの 4 分位点の第 1 と第 3 番目の範囲を示し, Whisker は Box 範囲の上限と下限から Box 範囲の 1.5 倍以内での最大, 最小のデータポイントを示す. Box 内の水平線は中央値, 外れ値は円で示す.

Fig. 3 Boxplot shows the distribution of cost C_s obtained from 10 runs using SA_SOFT2 for each ITC instance 1 to 10. The box shows the range between the 25% and the 75% quantile of the data. The whiskers extend to the most extreme data point which is no more than 1.5 times of the interquartile range from the box. A bar in the box is the median and the circles are outliers.

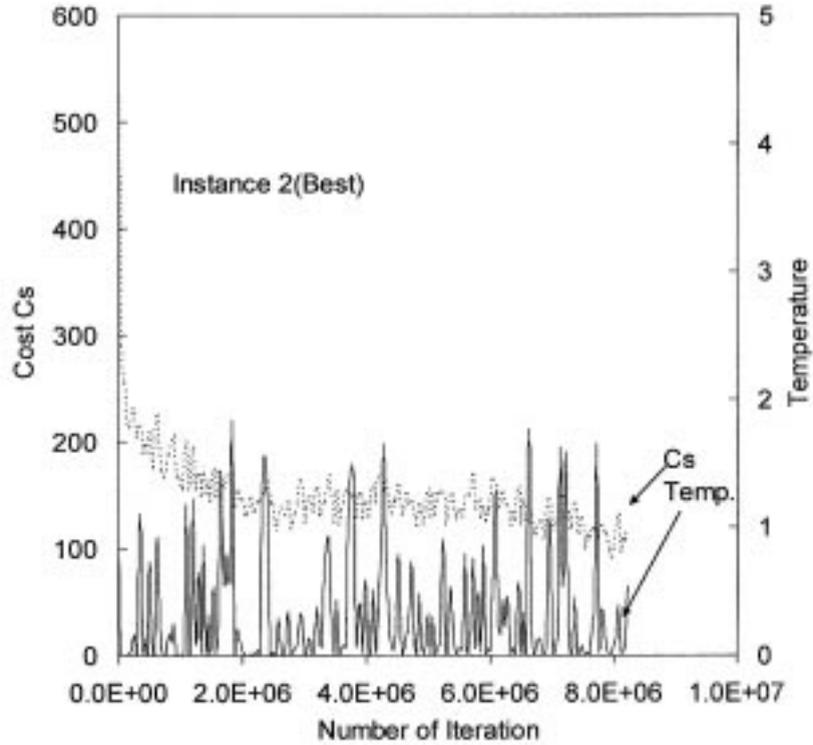


図4 Instance2のコスト C_s の最良値が得られた実験の SA_SOFT2 の反復回数に対する C_s と温度(Temp.)変化。
 Fig. 4 Cost and temperature versus number of the iteration of SA_SOFT2, where the best cost of C_s for instance 2 is obtained.

表3 SA_SOFT2 適用後の Instance 1-10 の Soft Constraints の違反値 (コスト値) の実験結果を, ITC2002 参加者の上位 7 番目までの結果と比較した表。Our Results(Best)/(Median)は筆者らの 10 回の実験における最良値と中央値をそれぞれ示す。Claimed は ITC 参加者自身の報告値, Verified は ITC 委員会の評価値を示す。

| Place | Author(s) | Instance | | | | | | | | | |
|-------|-------------------------------------|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | P.Kosluch | 45 | 25 | 65 | 115 | 102 | 13 | 44 | 29 | 17 | 61 |
| 2 | B.Jaumard et al. | 61 | 39 | 77 | 160 | 161 | 42 | 52 | 54 | 50 | 72 |
| 3 | E.Burke et al. | 85 | 42 | 84 | 119 | 77 | 6 | 12 | 32 | 184 | 90 |
| 4 | L.Di Gaspero and A.Shaerif | 63 | 46 | 96 | 166 | 203 | 92 | 118 | 66 | 51 | 81 |
| 5 | H.Amten and A.Lokketangen(Claimed) | 132 | 69 | 137 | 228 | 225 | 133 | 161 | 131 | 126 | 148 |
| 6 | Our results(Best) | 140 | 92 | 143 | 343 | 277 | 248 | 149 | 126 | 100 | 135 |
| 7 | A.Dubourg et al.(Claimed) | 131 | 97 | 145 | 318 | 281 | 199 | 194 | 128 | 109 | 141 |
| 8 | H.Amten and A.Lokketangen(Verified) | 132 | 92 | 170 | 265 | 257 | 133 | 177 | 134 | 139 | 148 |
| 9 | A.Dubourg et al.(Verified) | 148 | 101 | 162 | 350 | 412 | 246 | 228 | 125 | 126 | 147 |
| 10 | G.Toro and V.Parada(Claimed) | 137 | 103 | 156 | 346 | 335 | 243 | 195 | 149 | 147 | 153 |
| 11 | G.Toro and V.Parada(Verified) | 178 | 103 | 156 | 369 | 336 | 246 | 225 | 210 | 154 | 153 |
| 12 | Our results(Median) | 158 | 111 | 172 | 383 | 333 | 300 | 199 | 145 | 127 | 158 |

表4 SA_HARD, S_SOFT1, SA_SOFT2 で使用した SA のパラメータの値と Event の移動規則
 Table 4 Values of SA parameters and rule of moving the events used for SA_HARD, SA_SOFT1 and SA_SOFT2

| Parameters & Method | α | β, β_0 (GEO_RH, ENH_RH1) | ϵ_0 (ENH_ RH1) | δ (REHEATING _SOFT2) | T_0 | MAX_ MOVE | Fixed_ Number | Rule of moving events |
|---------------------|----------|--|-------------------------------|-----------------------------------|-------|--------------|------------------|-----------------------------|
| SA_HARD | 0.99 | 0.99 | 0.05 | - | 1 | 100 | 30 | Same room |
| SA_SOFT1 | 0.99 | - | - | - | 1 | 5000 | - | Timeslot sequencing |
| SA_SOFT2 | 0.99 | - | - | 4 or 2 | 1 | 10000 | 100 | Same room |

5. 結び

大学の時間割作成の例題として ITC の時間割問題の一部を対象にして、実行可能な状態を得るまでの実行時間と、最適な時間割を得るための条件に対する違反数削減について、再加熱スケジューリングを伴う焼きなまし法と、局所探索法を同時に適用した場合の効果について実験した。実験結果より、焼きなまし法と局所探索法を併用することは実行可能な時間割を得るにはかなり効果があることがわかったが、焼きなまし法の再加熱時スケジュールの差異による変化は特に認められなかった。また、実行可能な時間割を更に最適化するための結果については、ITC の報告結果と比較すると、最良値ではまずまずの結果であったが、中央値による比較では満足すべき結果は得られなかった。今後の課題は、更に効果的な局所探索のアルゴリズムを考案し、最適な時間割編成に対する違反数をさらに削減することが必要であると考えられる。また、今回採用したプログラミング言語以外の実行時間についても比較検討する必要があると思われる。

文献

- [1] E. K. Burke and S. Petrovic, "Recent research directions in automated timetabling," *European Journal of Operational Research*, Vol.140, 2, 16, pp. 266-280, 2002.
- [2] Rossi-Doria, M. Samples, M. Birattari, M. Chiarandini, M. Dorigo,, L. M. Gambardella, J. Knowles, M. Manfrin, M. Mastrolilli, B. Paechter, L. Paquete, and T. Stiitzle, "A comparison of the performance of different metaheuristics on the timetabling problem," *Lecture Notes in Computer Science*, Vol.2740, Springer Verlag KG, pp.329-351, 2003.
- [3] D. Abramson, "Constructing school timetables using simulated annealing: sequential and parallel algorithms," *Management Science*, Vol.37, 1, pp.98-113, 1991.
- [4] <http://www.idsia.ch/Files/ttcomp2002/>
- [5] S.R.White, "Concepts of scale in simulated annealing," *Proceedings of IEEE International Conference on Computer Design*, pp.646-651, 1984.
- [6] D. Abramson, M. Krishnamoorthy, and H. Dang, "Simulated annealing cooling schedules for the school timetabling problem," *Asia-Pacific Journal of Operational Research*, 16, pp.1-22, 1999.
- [7] <http://www.idsia.ch/Files/ttcomp2002/results.htm>
- [8] M. Chiarandini, K. Socha, M. Birattari and O. R. Doria, "An effective hybrid approach for the university course timetabling problem," AIDA-03-05, FG Intellektik, TU Darmstadt, March, 2003.