

看護師勤務表作成問題への進化型多目的最適化の応用

奥寺将至*

渡邊真也†

(室蘭工業大学大学院 情報電子工学系専攻)[‡] (室蘭工業大学大学院 しくみ情報系領域)

1 はじめに

看護師勤務表作成問題 (Nurse Scheduling Problem, NSP) は、病院などの医療施設における1ヶ月間の看護師勤務表を様々な制約条件の下で作成する問題である [1]. その特徴は、膨大かつ複雑に関係しあう制約条件下における最適化の困難さであり、文献 [2] における調査では、1部署1ヶ月分の勤務表作成に平均約11時間が費やされているとの結果が報告されている。

このような背景から勤務表作成の自動化に対するニーズは高く、これまで、遺伝的アルゴリズム (Genetic Algorithm, GA) や差分進化法 (Differential Evolution, DE) などのヒューリスティックアルゴリズム、および数理計画法などを適用した研究が報告され [1, 3, 4, 5, 6], これらの研究成果に基づく商用の看護師勤務表作成システムが発売され広く利用されている [7, 8, 9].

しかしながら商用の作成システムでは、幾つかの問題において最低限満たすべき制約を満足した解が導出できなかったり、仮にそのような解が導出できたとしてもその候補が1つのみで選択肢に乏しいといった問題点が指摘されており、最適化エンジンの改善が強く求められている [7].

そこで本研究では、看護師勤務表作成システム「セルヴィス」¹ を開発するシステムバンク株式会社との共同研究の下、上記の問題点を解決する新たなNSPに対する最適化アプローチの開発を試みた。具体的には、進化型多目的最適化 (Evolutionary Multi-criterion Optimization, EMO) に基づいた枠組みを利用した、効率的な解探索のためにNSPの問題性質に特化した複数の独自の探索メカニズムの実装を行った。

提案アプローチの有効性の検証には、共同研究先であるシステムバンク社内においてベンチマークとして利用されている3種類のテストデータを利用し、既存エンジンとの比較実験を行った。

本稿の構成を以下に述べる。まず、2章においてNSPについて概説し、3章においてNSPの定式化および提案手法の概要を説明する。4章において数値実験を行い提案手法の有効性について検証し、最後にまとめを述べる。

2 看護師勤務表作成問題

一般的に看護師勤務表作成問題 (Nurse Scheduling Problem, NSP) は、Fig. 1に示すような m 人の看護師の1ヶ月 (ここでは日数を n とする) の勤務表をできる限り制約を満たすように作成することが目的であり [1, 2, 3, 4, 5, 6], 勤務形態を要素として持つ $m \times n$ 行列を決定する問題として捉えることができる [6]. 看護師は管理職である師長や副師長だけではなく、キャリアに応じてグループ化されることが多く [2], 勤務形態も2交代制もしくは3交代制で分類されることが多い。

NSPにおける制約条件は様々なものがあるが、看護の質に関するもの (勤務制約条件 [10]) と看護師の生活の質に関するもの (看護師制約条件 [10]) の2種類に大別することができる。前者の多くが勤務表における縦方向 (日付ごとの制約) に関する制約であり、後者の多くが横方向 (スタッフごとの制約) に関する制約であることから、それぞれ縦方向制約と横方向制約として捉えることができる。また制約の強さには、その制約を充足しないと勤務表として成り立たないハード制約とできる限り満たして欲しい (必ずしも満たす必要のない) ソフト制約の大きく2種類あり、勤務表を作成する際には全てのハード制約を満たすことが第一の目標となる。

NSPにおける代表的な制約を以下に示す。

- 縦方向制約 (看護の質に関する制約, 勤務制約条件 [10])
 - 各日の各勤務形態に必要な人数
 - グループ人数 など
- 横方向制約 (看護師の生活の質に関する制約, 看護師制約条件 [10])
 - 各看護師に割り当てる勤務回数
 - 勤務パターン など

各日の各勤務形態に必要な人数は、病院を運営する上で最低限必要な制約条件であり、1日に必要な各勤務形態に割り当てる勤務人数を設定する。また、グループ人数は1日あたりに各勤務形態に必要な各グループ人数に関する制約条件であり、病院が提供する医療サービスの質を維持するためのものである。例えば、「準夜勤はランクA, ランクBから1人以上」のように設定される。

各看護師に割り当てる勤務回数は、各看護師が1ヶ月あたりに割り当てられる各勤務回数に関する制約条件であり、例えば「看護師Dは1ヶ月あたり休日は10回、深夜勤は5回」などのように設定される。また、勤務パターンは、「日勤→休日→準夜勤→深夜勤」のように、連続する

*10024039@mmm.muroran-it.ac.jp

†in@csse.muroran-it.ac.jp

‡050-8585 北海道室蘭市水元町27番1号 情報工学科事務室 (渡邊 真也研)

¹セルヴィスは、全国約800病棟の導入実績を持つ市販ソフトウェアである。

	1	2	3	4	5	6	...	30	31	休	深	準	日
	火	水	木	金	土	日		水	木				
看護師A	深	準	休	日	日	深		準	休	8	5	5	13
看護師B	日	深	準	日	休	休		日	休	8	6	6	11
看護師C	休	深	準	深	準	休	...	休	日	8	5	5	13
看護師D	深	準	日	休	深	準		準	深	8	5	5	13
看護師E	日	日	休	休	日	日		深	準	8	6	6	11
看護師F	深	準	深	準	休	深		日	深	8	6	6	11
⋮													
看護師O	準	休	深	準	深	準		日	休	8	5	5	13
看護師P	日	日	日	休	深	準	...	準	深	8	4	4	15
日	10	11	10	9	6	6		12	11				
深	3	3	3	3	3	3	...	3	3				
準	3	3	3	3	3	3		3	3				

Fig. 1 Example of attendance sheet.

▼夜勤の後は休日とする

▼夜勤はベテランか中堅から少なくとも1人

	1	2	3	4	5
	月	火	水	木	金
ベテランA	夜	休	日	日	日
ベテランB	日	日	夜	休	夜
中 堅 A	夜	休	日	夜	休
中 堅 B	日	日	夜	休	夜
新 人 A	休	夜	休	夜	休
新 人 B	休	夜	休	日	日

(a) グループ人数条件を違反

	1	2	3	4	5
	月	火	水	木	金
ベテランA	夜	休	日	日	日
ベテランB	日	夜	夜	休	夜
中 堅 A	夜	休	日	夜	休
中 堅 B	日	日	夜	休	夜
新 人 A	休	日	休	夜	休
新 人 B	休	夜	休	日	日

(b) 勤務パターン条件を新たに違反

Fig. 2 Example of conflict between constraints.

勤務形態の系列に関する制約条件である。この系列は勤務時間帯や勤務負荷を考慮して設定され、「日勤→日勤→日勤のパターンは禁止」のように、禁止する系列が設定されることもある。

NSP における最大の特徴は制約条件の複雑さとその充足の難しさである。一般的に縦方向制約と横方向制約は性質が大きく異なるため、両者の間にはトレードオフの

関係が存在することが知られている。そのため制約同士が強くバッティングし、制約の充足が困難になることが多い。制約同士の干渉に関する具体的な例を Fig. 2 に示す。まず Fig. 2-(a) では、2 日において新人 A, B が縦制約であるグループ人数制約を違反している。そこでベテラン B と新人 A の 2 日の勤務形態を入れ替えることにより、制約を満たすことが出来るが、この操作によってベテラン B の 2, 3 日の勤務パターン制約を新たに違反してしまうことがわかる (Fig. 2-(b))。

遺伝的アルゴリズム (Genetic Algorithm, GA) を用いて勤務表を作成する際には、常に制約同士がバッティングしているため、これらを解消するように勤務形態を操作することになる。しかしながら、どの制約とどの制約が強くバッティングをするのか事前に予測することが困難であるため、1 つの制約を充足することにより、他の制約を複数違反してしまうことも少なくない。またバッティングはいわゆる局所解のような状態として現れ、制約充足の進んだ違反の少ない勤務表ほどバッティングが強固に (局所解の谷が深く) なるため、探索が進むほど探索が停滞しやすくなり、ハード制約を全て満たす勤務表の導出を困難にしている。

3 提案手法

本章では、提案するアプローチの概要について述べる。

3.1 方針

本研究で提案するアプローチの目標は、1) より短い時間でのハード制約を満たす解の導出 (高い探索性能) と、2) ハード制約を満たした複数の解候補の導出の大きく 2 つであり、特に前者を実現するためには、前章で述べた NSP の持つ問題特性を十分に考慮したメカニズムを組み込む必要がある。そのため、下記に示す 3 つの方針に基づいたアルゴリズム設計を行い、効率的な解探索の実現を試みた。

- 1) 特定の制約条件に対する充足
- 2) (その時点の) 優良個体に対する徹底的な局所探索
- 3) 探索が局所解に陥った場合の脱出

1 つ目の制約充足は、特定の制約については必ず充足するように配慮することで実質的な探索領域の削減を意図したものである。提案するアプローチでは、初期個体生成時に特定の制約については必ず充足させ、各遺伝的操作時には充足した制約をできるだけ違反しないように解操作を行っている。

また、2 つ目の優良個体に対する局所探索は、先鋭的に探索を進めるための方針であり、より短期解でハード制約を満たす解を導出することを目的としている。提案アプローチでは、各遺伝的操作および局所探索においてその方針を徹底しており、必ず現時点での最良と思われる個体が各操作の適用対象に選択される方針をとっている。

Table 1 Priority and weights of constraint.

Priority	Weights
絶対	10000
出来れば	100
可能なら	1

3つ目の局所解からの脱出は、探索終盤（ある程度の制約充足が進んだ違反の少ない状態）において局所解からの脱出が困難になるという問題特性に対応したメカニズムであり、本アプローチでは局所探索のフェーズにおいてこのメカニズムを陽に扱っている。

また後者を実現させるために、問題を多目的最適化問題として定式化し、進化型多目的最適化 (Evolutionary Multi-criterion Optimization, EMO) に基づいた枠組みを適用した。EMO を採用した理由は大きく2つあり、1) 前章で述べたように、NSP における制約は2種類に大別され両者の間にはトレードオフの関係があり、多目的化との親和性が高いこと、また2) 特徴の異なる複数の良質な解候補の導出が期待できるためである。

3.2 定式化

本研究における NSP は多くの先行研究と同様に、1ヶ月の勤務表を最適化パラメータとして扱った。そのため解は看護師の人数を m 、日数を n とした $m \times n$ 行列により表現され、行列の各要素には k 種類の勤務形態のうち1つを必ず割り当てる。

また、制約条件の強さに関して、本研究ではセルヴィスにおける条件設定に基づき Table 1 に示す3種類の優先度を設定した。Table 1 における「絶対」は必ず満たす必要のあるハード制約であり、「出来れば」および「可能なら」はそれぞれソフト制約であるものの重要度の違いから異なる重み付けを行った。その際、勤務表作成時の最低条件は全てのハード制約の充足であるため、「出来れば」および「可能なら」に対して「絶対」の重み値は極端に大きな値を設定している。なお、本研究では絶対条件を全て満たした状態の解が得られた時点で勤務表は完成したものと扱った。

解の評価値は Table 1 に示す重み値を用いた制約条件違反値の総和から算出する。このとき制約条件を前章で述べた縦方向制約 (勤務制約条件) と横方向制約 (看護師制約条件) の2種類に大別することにより、2目的最小化問題として定式化した。評価式を式 (1) および式 (2) に示す。

$$f^{\text{people}}(x) = \sum_{i=1}^k \omega_i x_i^{\text{people}} \quad (1)$$

$$f^{\text{duty}}(x) = \sum_{j=1}^l \omega_j x_j^{\text{duty}} \quad (2)$$

式 (1) は横方向制約 (看護師制約条件)、式 (2) は縦方向制約 (勤務制約条件) に該当する評価値である。式中における i, j は制約条件、 k, l は制約条件数、 x_i, x_j は制約条件 i, j の違反数、 ω_i, ω_j は Table 1 に基づく制約条件 i, j の重みである。

3.3 提案手法の流れ

提案手法の流れを以下に示す。なお、文中における最良個体 (Best) は、各目的関数値の総和が最小となる個体を指し、最悪個体 (Worst) は、各目的関数値の総和が最大となる個体を指す²。

- Step 0) 特定の制約条件を充足させた初期個体を N 個生成し、アーカイブに保存する。
- Step 1) アーカイブにおける最良個体 A_{Best} を P_1 、ランダムに選択した個体を P_2 とし、親個体とする。
- Step 2) 親個体 P_1, P_2 の個体間距離³ $D(P_1, P_2)$ を算出し、 $T \leq D(P_1, P_2)$ ならば交叉、 $T > D(P_1, P_2)$ ならば突然変異を行う。遺伝的操作は dMSXF や dMSMF[11] のような多段階の枠組みを利用している。
- Step 3) 生成した子個体の中から最良個体 C_{Best} を選択し、アーカイブに保存する。アーカイブサイズ A を超過している場合は、親体 P_1, P_2 またはアーカイブにおける最悪個体 A_{Worst} と置き換える。
- Step 4) アーカイブの個体に対して局所探索を行う。対象となる個体は、Step 3) においてアーカイブが更新されなかった場合は A_{Best} であり、更新された場合はランダムに選択した非劣解1個体とする。
- Step 5) アーカイブの個体が1つでも「絶対」の制約条件 (ハード制約) を全て満たしていれば勤務表が完成したと見なして終了する。そうでなければ Step 2) に戻る。

上記のように、提案手法では遺伝的操作と局所探索を組み合わせた枠組みとなっており、これら各メカニズムの有機的な連携により効率的な解探索を試みている。また、遺伝的操作および局所探索の対象として必ずその時点の最良個体 (A_{Best}) を選択し、優良個体に対する改善を最優先する仕組みを採用していることが分かる。

4 数値実験

提案手法の有効性を検証するために、提案手法と現在のセルヴィスに搭載されているエンジンとの比較実験を下記の2種類について行った。

² 提案手法では、制約条件を2種類に分けた2目的最適化問題として扱っているが、NSP において最も重要なのはハード制約違反を全て充足することであり、また目的関数値が小さいほど制約違反が少ない良い個体であると判断できる。そのため、各目的関数値の総和による個体の優越を行うことで、 $f^{\text{people}}(x)$ および $f^{\text{duty}}(x)$ のハード制約充足を最優先する戦略を採用した。

³ ハミング距離を用いる。

Table 2 Classification of type of constraints in test problems.

	# Constraints		
	Test Problem 1	Test Problem 2	Test Problem 3
各日に必要な各勤務人数	4	5	5
グループ人数	13	10	12
その他 (縦方向制約)	3	9	7
各看護師に割当てする勤務回数	2	5	7
勤務パターン	1	7	6
禁止勤務パターン	8	8	18
その他 (横方向制約)	5	8	9

Table 3 Detail of test problems.

	Test problem		
	1	2	3
シフト	2交代制	2交代制	3交代制
勤務形態数	4	5	5
看護師数	30	25	26
固定箇所数	0	99	136
縦方向制約数	20	24	24
横方向制約数	16	28	40

- 1) 一定時間内での満足解導出割合
- 2) 目的関数値の推移

前者はアルゴリズムの安定性の検証を目的としており、割合が高いほど安定して勤務表を導出できると考えられる。また後者は探索効率の分析を目的としており、具体的には同程度の評価回数ごとに母集団における個体評価値の最良 (各目的関数値の総和の最小) 値をプロットし、得られる推移の比較を通して探索効率の分析を行う。

対象問題にはシステムバンク社においてセルヴィスの開発時に利用されている3種類のベンチマークテスト問題を用いた。各問題は、実際の病棟事例に基づいて作成されており、それぞれ異なる困難性や特性を有している。以下、各テスト問題の詳細について説明する。各問題の設定および制約条件の内訳は Table 3, Table 2 の通りである。

テスト問題 1

2交代制の病棟を想定しており、看護師は30名、勤務形態は4種類 (夜勤, 明勤, 日勤, 休日) 存在する。看護師数が最も多いため、勤務表のサイズが最大の問題である。また明勤は夜勤の午前12時以降の勤務時間を指す勤務形態であり、夜勤と明勤を一連の勤務系列として扱うことによって日勤と夜勤から成る2交代制の病棟の勤務表を作成することができる。

テスト問題 2

テスト問題1と同様に2交代制の病棟を想定しており、看護師は25名、勤務形態は5種類 (夜勤, 明勤, 日勤, 遅出, 休日) 存在する。しかし日勤が日勤と遅出の2種類に分かれており、また勤務表中にあらかじめ勤務形態が決定された変更不可な固定箇所が複数存在するため、探索空間が複雑化していることが考えられる。

テスト問題 3

テスト問題3では3交代制の病棟を想定しており、看護師は26名、勤務形態は5種類 (深夜勤, 準夜勤, 日勤, 遅出, 休日) 存在する。勤務形態数はテスト問題2と同じだが、3交代制であるため、夜勤が準夜勤と深夜勤に分けられるところが大きな特徴である。また固定箇所が多く制約条件の数が最大であるため、探索空間の複雑性の高さが予想される。更に縦方向条件に対して横方向条件の割合が大きいため、探索の際には横方向制約の特性を強く配慮する必要があることが推察される。

4.1 各種設定パラメータ

アーカイブサイズ A を64, 初期個体数 N を2とし、遺伝的操作を切り替える閾値 T は50と設定した。また実験に使用した端末はCPUがIntel®Core i7-3770であり、メインメモリは16GB搭載している。

4.2 実験結果と考察

本節では、4章で記した数値実験の結果とその考察を述べる。

一定時間内の満足解導出割合

各テスト問題に対して提案手法および既存エンジンを30試行ずつ適用し、一定時間内に勤務表が完成⁴した割合を Table 6, 探索に費やした計算時間および評価回数を Table 4, Table 5に示す。ここで一定時間の上限は10分間と設定し、10分以内に勤務表が完成しなかった場合は強制的に探索を終了するものとした。

Table 6から、既存エンジンではテスト問題3でしか勤務表を導出できなかったが、提案手法では全ての問題に対して100%勤務表を導出できた。また Table 4, Table 5から、既存エンジンでは勤務表が導出できなかったテスト問題1, 2において、提案手法ではわずか1, 2分程度で導出できていることがわかる。テスト問題3についても、計算時間が半分以下に削減できていることが確認できる。そのため、提案手法では全ての問題に対して既存エンジンよりも少ない計算資源かつ短時間で確実にハード制約を満たす勤務表の導出を実現できていることがわかる。

⁴ 母集団に保存している個体のうち、どれか1つでも絶対条件の制約違反数が0になれば勤務表が完成したと見なす。

Table 4 Computing times and number of iterations of original method in each test problem.

	Test Problem 1			Test Problem 2			Test Problem 3		
	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max
Computing time (sec)	N/A	N/A	N/A	N/A	N/A	N/A	266.028	394.988	544.407
# Iterations	N/A	N/A	N/A	N/A	N/A	N/A	1,191,204	2,439,949	2,866,499

Table 5 Computing times and number of iterations of proposal method in each test problem.

	Test Problem 1			Test Problem 2			Test Problem 3		
	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max
Computing time (sec)	33.738	112.808	322.789	25.984	84.321	182.155	74.186	152.364	323.345
# Iterations	399,375	1,376,266	4,039,736	243,286	830,809	1,833,295	639,494	1,306,054	2,812,701

目的関数値の推移

上記の実験結果を踏まえ、勤務表の導出が可能な評価回数として上限を 100 万回に設定し、評価回数が上限に達するまでの目的関数値の推移について比較を行った。ここで目的関数値は 4 章で述べたように、母集団の個体評価値の最良 (各目的関数値の総和が最小) 値を指すことに注意されたい。試行回数を 30 回とし、評価回数 100 万回時点の目的関数値が全試行中の中央値に最も近い試行の結果を比較データとして扱った。評価回数 2 万回毎に目的関数値をプロットした結果を Fig. 3 に示す。ただし手法ごとに Table 1 に示した制約条件の重み値が異なるため、プロットする際には既存エンジンで用いられている値⁵ に統一した。

Fig. 3 から、提案手法は全ての問題に対して常に既存エンジンよりも目的関数値が勝っており、収束性が非常に高いことがわかる。特に探索開始時 (評価回数 0 回) に注目すると、この時点で目的関数値に大きな差があり、初期個体生成時に特定の制約条件を充足する効果が表れていると考えられる。実際、このときのハード制約違反数は Table 7 の通りであり、充足できているハード制約数にも大きな差があることがわかる。

その後も提案手法は既存エンジンに比べて解が高速に収束しており、特にテスト問題 1 ではそれが顕著に現れている (Fig. 3-(a))。またテスト問題 2 の結果 (Fig. 3-(b)) に注目すると、既存エンジンで評価回数を 100 万回費やして得られる値が、提案手法では評価回数 5 万回程度で導出できており、計算資源が大幅に削減されていることがわかる。テスト問題 3 では他の問題ほどの有意差は得られていないものの、やはり提案手法が勝っていることが確認できる。この結果から、提案手法は既存エンジンよりも効率的な探索が実現できていることは明らかである。

以上 2 つの実験結果より、提案手法は既存エンジンよりも明らかに探索性能が向上していると言える。

Table 6 Success rate for finding a satisfied attendance sheet in proposal and original methods.

	Proposed method	Original method
Test Problem 1	30/30	0/30
Test Problem 2	30/30	0/30
Test Problem 3	30/30	11/30

Table 7 The number of hard constraint violation of best individual of initial state in proposal and original methods.

	Proposed method	Original method
Test Problem 1	502	1083
Test Problem 2	233	1862
Test Problem 3	440	1778

5 おわりに

本稿では、看護師勤務表作成問題 (Nurse Scheduling Problem, NSP) に対して進化型多目的最適化に基づく効率的な解探索アプローチを提案し、数値実験を通してその有効性について検証を行った。実際の病棟事例に基づいた 3 種類のベンチマークテスト問題への適用を通して、提案手法の探索性能が既存エンジンよりも明らかに優れていることが確認できた。

今後の展望としては、得られる解候補の多様性について着目し、ハード制約が全て充足した時点で多様性重視の探索に切り替えるようなフレームワークの導入を予定している。

参考文献

- [1] 川中普晴, 山本康高, 吉川大弘, 篠木剛, 鶴岡信治. 遺伝的アルゴリズムを用いた看護婦勤務表の自動生成. 電気学会論文誌, Vol. 122-C, No. 6, pp. 1023-1032, 2002.
- [2] 池上敦子. ナース・スケジューリング-調査・モデル化・アルゴリズム (特集計算推論-モデリング・数理・アルゴリズム). 統計数理, Vol. 53, No. 2, pp. 231-259, 2005.
- [3] 新妻真輔, 池上敦子, 品野勇治. 列生成法を用いた

⁵ 絶対 : 100, 出来れば : 10, 可能なら : 1

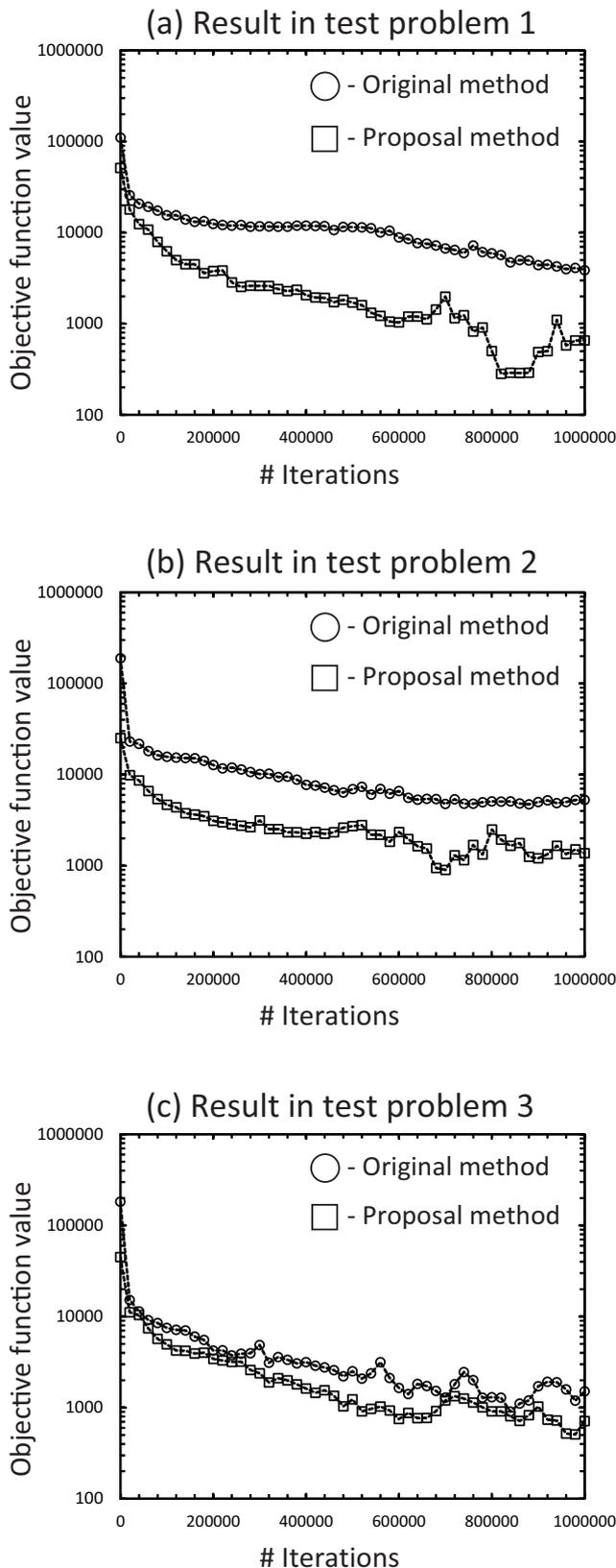


Fig. 3 The transitions of the objective function value of proposal and original methods in each test problem.

ナーススケジューリング問題の解法. 情報処理学会研究報告. MPS, 数理モデル化と問題解決研究報告, Vol. 2009, No. 19, pp. 221-224, feb 2009.

[4] 小清水誠, 荒井誠. Gaによるナーススケジューリング問題の一解法. 釧路工業高等専門学校紀要, Vol. 36, pp. 61-66, dec 2002.

[5] 小清水誠, 荒井誠. Gaによるナーススケジューリング問題の一解法: 第2報大規模問題へのアプローチ. 釧路工業高等専門学校紀要, Vol. 37, pp. 41-46, dec 2003.

[6] 串田淳一, 大場和久, 亀井且有. Differential evolutionによるナーススケジューリングアルゴリズムの提案. 進化計算シンポジウム 2011, pp. 299-306, dec 2011.

[7] システムバンク株式会社「セルヴィス」. <http://servis-net.jp/>. 参照 2014年8月1日.

[8] 富士通 看護師勤務管理支援システム「ナーススケジューラ v6」. <http://jp.fujitsu.com/group/fwest/services/nurse/>. 参照 2014年8月1日.

[9] 松岡興産株式会社「看護師スケジューリング支援システム」. <http://www.matsusakakosan.co.jp/system/package03.html>. 参照 2014年8月1日.

[10] 敦子池上. ナース・スケジューリング問題のモデル化について. 情報処理学会研究報告. MPS, 数理モデル化と問題解決研究報告, Vol. 96, No. 10, pp. 1-6, jan 1996.

[11] 花田良子, 廣安知之, 棟安実治. 内挿/外挿的な領域への遺伝的多段階探索の解探索性能の解析. 人工知能学会論文誌, Vol. 24, No. 1, pp. 136-146, 2009.