An Evaluation of Certain Heuristic Optimization Algorithms in Scheduling Medical Doctors and Medical Students

Christine A. White[1], Emilina Nano[2], Diem-Hang Nguyen-Ngoc[2], and George M. White[2]

[1] Dept. of Medicine, Div. of Nephrology,
Queen's University, Kingston K7L 2V6, Canada
cw38@post.queensu.ca
[2] School of Information Technology and Engineering,
University of Ottawa,
Ottawa K1N 6N5, Canada
white@site.uottawa.ca

Abstract. Four heuristic algorithms based on or inspired by the well-known Tabu Search method have been used to cast heuristically optimized schedules for a clinical training unit of a hospital. It has been found experimentally that the algorithm of choice for this problem depends on the exact goal being sought where the execution time is one of the components of the goal. If only one run is allowed, then classical Tabu Search with a tenure of 5 gave the schedule with the lowest average (and fixed) penalty. If time is not of concern and many runs are allowed then the Great Deluge algorithm may generate the schedule with the lowest penalty.

1 Introduction

In work described in a previous PATAT conference [1] a stand-alone system for casting schedules of medical staff in the Internal Medicine Clinical Teaching Unit of the Ottawa Hospital was built using the Java programming language. The algorithm constructed an initial feasible schedule and then heuristically optimized it to reduce its perceived "badness". The algorithm used was a simple version of the tabu search (TS) algorithm introduced by Glover [2] and used many times since.

The requirement was to produce duty rosters (locally referred to as *call schedules*) for medical trainees (residents and medical students) in the Clinical Teaching Unit to man the overnight shift. The duties of a shift consist in rendering medical assistance to patients in need of it during the night when the majority of the medical trainees are no longer on duty. For each night in a 28 night cycle, a shift of (ideally) 5 persons consisting of a senior resident, 2 junior residents and 2 medical students has to be scheduled. Because of chronic understaffing the shift often consists of fewer than 5 persons. Sometimes 4 and sometimes 3 persons

E. K. Burke, H. Rudová (Eds.): PATAT 2006, pp. 320–328. ISBN 80-210-3726-1.

are used if there are not enough staff available. The staff chosen for these shifts have various "ranks" and may belong to one of two teams. Since these evening rounds are in addition to regular day shifts that the medical trainees must work, there are very stringent requirements that prohibit the personnel from being overworked beyond a certain point. These numerous requirements are formulated as *soft constraints* and each violation of a constraint is associated with an integer penalty whose magnitude is a measure of the undesirability of relaxing that constraint. The sum of these integers is the measure of the "badness" of the schedule. The TS algorithm is used to minimize this badness. The constraints are examined in detail and the penalties associated with them are discussed in the earlier paper [1]. An example of a call schedule is shown in figure 1.

		Team A		Team B	
	Senior	1st Call	2nd Call	1st Call	2nd Call
Tue:	Zaidi	Shefrin	Carrier	Puglia	------
Wed:	ElFirjani	Mongiardi	------	Rajput	Mufti
Thu:	Jolicoeur	Marwaha	Payne	Oliveira	Cohn
Fri:	Ellen	Mongiardi	Carrier	Oliveira	Bal
Sat:	Zaidi	Taylor	Radke	Rajput	------
Sun:	Ellen	Mongiardi	Carrier	Oliveira	Bal
Mon:	ElFirjani	Taylor	------	Puglia	Mufti
Tue:	Treki	Shefrin	Payne	Puglia	------
Wed:	Stewart	Taylor	Carrier	Bal	------
Thu:	ElFirjani	Mongiardi	Radke	Rajput	Cohn
Fri:	Jolicoeur	Taylor	------	Puglia	Mufti
Sat:	Stewart	Mongiardi	Payne	Oliveira	Bal
Sun:	Jolicoeur	Taylor	------	Puglia	Mufti
Mon:	Treki	Marwaha	Radke	Rajput	Cohn
Tue:	Stewart	Taylor	Payne	Bal	------
Wed:	Jolicoeur	Shefrin	Carrier	Oliveira	------
Thu:	Ellen	Marwaha	------	Oliveira	Mufti
Fri:	Zaidi	Shefrin	Radke	Rajput	------
Sat:	Jolicoeur	Marwaha	Carrier	Cohn	------
Sun:	Zaidi	Shefrin	Radke	Rajput	------
Mon:	Ellen	Mongiardi	------	Rajput	Mufti
Tue:	ElFirjani	Marwaha	Payne	Puglia	Cohn
Wed:	Zaidi	Shefrin	Radke	Oliveira	------
Thu:	Treki	Taylor	------	Bal	------
Fri:	Stewart	Marwaha	Payne	Cohn	------
Sat:	Ellen	Shefrin	------	Puglia	Mufti
Sun:	Stewart	Marwaha	Payne	Cohn	------
Mon:	ElFirjani	Mongiardi	Radke	Bal	------

Fig. 1. An example of a call schedule

The senior resident is the doctor in charge. The two teams, A and B, consist of a "First Call", *i.e.* the first person to call if required, and a "Second Call", the next person to call (if there is one).

The penalties can be broadly classified as horizontal or vertical penalties. Some nights *e.g.* the third and fourth lines on the schedule, corresponding to Thursday and Friday of the first week, are fully staffed with appropriate members from each team. Other nights *e.g.* the final Thursday are very short staffed, with only three medical trainees on duty. The first of these examples attracts a penalty of 0 while the second example attracts a penalty of 300. These are examples of horizontal penalties which are ones that can be evaluated simply by scanning each line separately.

Vertical penalties are those that arise from consecutive days. The first example of two nights referred to above has one trainee, Oliveira, working for two consecutive nights. This attracts a penalty of 100. The weekends, defined to consist of Friday, Saturday and Sunday, are very sensitive. A pattern consisting of working on Friday and Sunday (but *not* Saturday) or its converse are greatly desired. Failure to achieve this attracts a high penalty. The example of table 1 manages to achieve this goal at the expense of attracting horizontal penalties. As a rule, horizontal and vertical penalties play against each other, reducing one usually implies increasing the other. The one that "wins" is the one having the higher value. Table 1 lists the various defects that each night's shift might have and the corresponding penalties.

Table 1. Conditions and their penalties

Condition	**Penalty**
one student missing	5
student replaces missing junior - senior is on student's team	10
two student missing	20
junior and student missing - student takes junior's place - senior is on student's team	40
student replaces missing junior - senior is not on student's team	80
junior and student missing - student takes junior's place - senior is not on student's team	100
two juniors missing - replaced by two students	300
anything else	500

In the quest to cast the best schedule in a reasonable time, four different heuristic algorithms based on the local search strategy were implemented and tested with real data obtained from the hospital. A number of different data sets were used. For real data, the results all exhibited the same overall behaviour. We believe this is so because the staff mix and numbers were chosen by a supervisor having much experience casting schedules by hand, thus they all exhibit a certain homogeneity. This paper summarizes the four algorithms used and discusses the results obtained from each one.

2 Algorithms Investigated

The algorithms investigated were:

– Tabu Search with Fixed Tenure
– Tabu Search with Random Tenure
– Great Deluge
– IDWalk

The first of these is deterministic. The algorithm always yields the same answer when given the same data. Thus the Tabu Search with fixed tenure was executed once for each tenure value. The other algorithms are not deterministic. A pseudo-random generator is used and therefore each run may yield a different result. For these cases, 20 runs were made and the values shown in the tables are based on these 20 runs. For the Great Deluge algorithm, an additional 1800 runs were made and analyzed more thoroughly.

The results were all obtained from code written in C# on the Microsoft Visual Studio .Net IDE and executed on a PC under Windows XP with a Celeron chip at 2.8 GHz. with 192 MBytes of RAM.

2.1 Tabu Search with Fixed Tenure

The classical Tabu Search algorithm (TS) whose entries in the single tabu list have a fixed tenure was the original algorithm implemented in the project [1]. This algorithm was rerun with actual hospital data keeping the tenure fixed but varying its value in different runs. This algorithm is deterministic. A run always returns the same schedule having the same penalty. The aspiration function used by TS was set to return a value equal to the best value found so far. The algorithm was run using fixed tenures of 5, 10, 20 and 40. The values of the penalties and execution times of the schedules obtained are shown in table 2.

Table 2. Values of penalties and execution times

tenure (fixed)	penalty	execution time (s.)
40	1520	27.6
20	1495	27.0
10	1415	25.8
5	1270	25.9

2.2 Tabu Search with Random Tenure

This variation of the classical TS differs only in that the tenure of an entry in the tabu list is drawn from a series of pseudo-random integers whose value is determined when the entry is inserted into the list. This procedure is the same as that discussed by Di Gaspero and Schaerf in [3]. The distribution of these tenures was uniform discrete on the interval [10, 40]. This algorithm is not deterministic and successive runs with the same program and the same data usually give different schedules having different penalties. Statistics obtained from the 20 runs are summarized in table 3.

Table 3. Statistics obtained from runs of TS with random tenure

	penalty	execution time (s.)
min	1235	26.4
max	1495	28.5
mean	1342	27.2
std dev	82.8	0.6

2.3 Great Deluge

This method was introduced by Gunter Dueck in 1993 [4]. The name "Great Deluge" was chosen by Dueck to illustrate the progress of the algorithm in an analogy where a person is trying to keep his feet dry by climbing in mountainous terrain during a great deluge. The person moves by taking a step in some randomly chosen direction while the water level continues to rise. He stays in the new position only if he can keep his feet dry. If this is not possible he moves randomly again. Eventually all moves result in wet feet or the time has run out and the algorithm stops. This algorithm has the desirable property that there is only one adjustable parameter, the rate of rise of the water level. As above, this method is not deterministic. The results are summarized in table 4.

Table 4. Statistics obtained from runs of the Great Deluge algorithm

	penalty	execution time (s.)
min	1210	7.4
max	2840	9.6
mean	1720	7.8
std dev	353.9	0.6

2.4 IDWalk

This method, called Intensification/Diversification Walk (or IDWalk), was introduced by B. Neveu *et al.* [5] and is related to the TS method. There are three parameters, S, the number of moves, Max, the number of potential neighbours studied in each move and $SpareNeighbour$, the diversification strategy.

This algorithm performs S moves and returns the best solution it found. In choosing the next move to make, it examines at most Max candidate neighbours, selecting them randomly. If the penalty of this candidate, x', is less than or equal to the penalty of the current solution, then the solution corresponding to x' is chosen for the move. If no neighbour has been selected from among the Max examined, then one of the rejected candidates is chosen for the next move. If $SpareNeighbour$ was set equal to "best", then the least bad of the rejected neighbours is chosen for the next move. Otherwise, $SpareNeighbour$ was set to "any" and any one of the rejected neighbours is chosen randomly for the next move.

In this investigation, $S = 1000$ and $Max = 378$. The two choices for $SpareNeigh$ were tried and it was found that the value, "best" gave the superior results. As before a number of runs were made using the same input data. The results are summarized in table 5.

Table 5. Statistics obtained from runs of the IDWalk algorithm

	penalty	execution time (s.)
min	1160	57.4
max	1810	77.5
mean	1394	68.1
std dev	225.1	7.7

3 Comparison of penalties obtained by the four methods

These results are shown graphically in figure 2.

The scale of the entire figure is shown by the top line. The left end of the line has the value 1100 and the right end has the value 2100. The second line of the figure shows the penalties when TS is used with fixed tenures of 5, 10, 20 and 40. The best call schedule (with the smallest penalty) is found when the tenure was fixed at 5. This schedule has a penalty of 1270.

When random tenures in the interval [10, 40] are used, the schedules obtained are generally better than those obtained with a fixed tenure set to any value within this interval. Recall that about 66% of the values obtained are within one standard deviation on either side of the mean (which means that about 33% are not). The best value obtained was 1235 which is better than anything that was

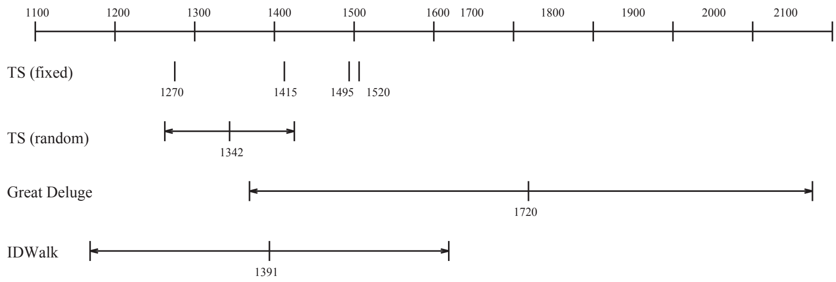

Fig. 2. Comparison of results

obtained using fixed tenure. The worst value was 1495 which is about the same value as that obtained using a fixed tenure of 20 and better than the results with a value of 40. The mean value and the interval delineated by plus or minus one standard deviation (containing about 66% of the values) is shown on figure 2 immediately below the results for TS(fixed).

As expected, the values obtained with the Great Deluge algorithm were rather poor. The average penalty value of 1720 was the worst of the four methods and the standard deviation was the largest obtained. The spectrum of values obtained was such that the worst was worse than any of the TS results but the best was also better than anything found by TS. The algorithm executed rapidly and its mean time to completion, at 7.8 seconds was better than three times faster than its fastest rival. The mean value and the one standard deviation interval are shown on figure 2.

The IDWalk method yielded a mean penalty of 1394, better than Great Deluge but worse than TS with random tenure. The large standard deviation of about 225 illustrates the range of values obtained. Its lowest value of 1160 was the best value obtained by any of the algorithms. Its highest value was higher than any obtained by any of the TS methods but worst than Great Deluge. Its mean execution time of 68.1 s. makes it the slowest algorithm to complete execution. The lowest value was obtained by the run having the longest execution time. As before the mean value and the one standard deviation interval are shown on the last line of the figure.

A t-test on pairwise comparison of the three distributions assuming unequal variances shows that the TS (random) and IDWalk algorithms result in distributions having identical means, at the 95% level. When TS (random) and Great Deluge are compared the means were found to be different. The same was true when the IDWalk and Great Deluge distributions were compared.

Because of the wide range of penalty values found by the Great Deluge algorithm, this case was studied further. Using the same input data, 1800 runs were made. The histogram of the results obtained is shown in figure 3.

It is evident that there is a large variation in the values obtained, the smallest value being 1135 (obtained 3 times) and the largest value being 3620 (obtained

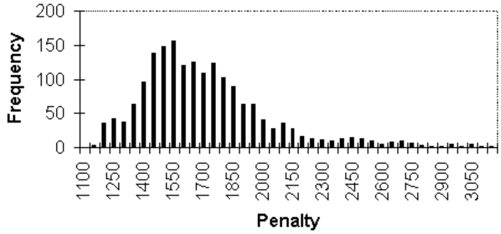

Fig. 3. Histogram of 1800 penalties obtained for the Great Deluge algorithm

once). Therefore the worst schedule has an associated penalty that is more than three times larger than the best one. The penalty of 1135 is the best obtained in any run of any algorithm studied here. Although the mean value was the worst obtained in this study, the minimum value was the best minimum value due to the high variability of the results.

4 Discussion

Using the experimental results obtained using data from this medical call schedule problem, the ranking of the four methods tested based on their mean penalty is:

1. TS - fixed tenure = 5
2. TS with random tenure [10, 40]
3. IDWalk
4. Great Deluge

For consistency of results, the method of choice is TS with random tenure for the non-deterministic algorithms and TS with fixed tenure for the deterministic ones.

However, if execution time is not critical, the method of choice may be the Great Deluge which yields results rapidly, even if most of them are not very good. One of them may be very good indeed.

If it is desired to find the *minimum minimorum* and execution time is not a concern then IDWalk may well give the best results. If time is a concern, then it may be useful to run the Great Deluge multiple times and select the schedule having the smallest penalty. For the time of one run of IDWalk, we could have about ten runs of Great Deluge. This may be the method of choice in a setting where actual schedules have to be produced and used in a real hospital. The system can be left running overnight and the best of the schedules obtained can be retrieved in the morning. With 12 hours available for repeated automatic runs and an 8 second run time, about 5400 schedules can be produced and the best one used. If desired, the best schedule could be taken as the initial solution for another metaheuristic to further improve.

It should be remembered that these results strongly reflect the specific requirements and data taken from one institution. Discussions with other units in the same hospital revealed that apart from having to use different data the algorithms would have to use different weights and different criteria in forming the corresponding penalties. This might lead to different conclusions. We are investigating this further.

5 Conclusions

Depending on the desired goals and available execution time, the algorithm of choice for this problem will vary. The lowest average penalty is obtained by TS with fixed tenure of 5. The lowest penalty found in the comparative samples was obtained by IDWalk, simultaneously taking the longest time to get it. The lowest penalty of all was obtained during the additional Great Deluge runs. If time is of little concern, multiple runs of Great Deluge may be the best strategy.

References

1. Christine A. White and George M. White. Scheduling doctors for clinical training unit rounds using tabu optimization. *PATAT IV: Lecture Notes in Computer Science*, 2740:120–128, 2003.
2. F. Glover and M. Laguna. *Tabu search.* Kluwer Academic Publishers, 1997.
3. Luca Di Gaspero and Andrea Schaerf. Multi-neighbourhood local search with application to course timetabling. *PATAT IV: Lecture Notes in Computer Science*, 2740:262–275, 2003.
4. Gunter Dueck. New optimization algorithms. *J. Comp. Phys.*, 104:86–92, 1993.
5. Bertrand Neveu, Gilles Trombettoni, and Fred Glover. Idwalk : A candidate list strategy with a simple diversification device. *CP 2004: Lecture Notes in Computer Science*, 3258:423–437, 2004.