



Journal of Applied Sciences

ISSN 1812-5654

science
alert

ANSI*net*
an open access publisher
<http://ansinet.com>

Adaptive Neighbourhoods Structure Selection Mechanism in Simulated Annealing for Solving University Course Timetabling Problems

H. Y. Tarawneh and Masri Ayob

Data Mining and Optimization Research Group (DMO), Center of Artificial Intelligence Technology, Universiti Kebangsaan Malaysia, 43600 Bangi, Selangor, Malaysia

Abstract: Usually, meta-heuristic approaches that use several neighbourhood structures can perform better than single neighbourhood structure. However, choosing a suitable neighbourhood structure to be applied during the search process is also a crucial decision. Therefore, this study proposes an adaptive neighbourhoods structure selection (AD-NS) mechanism that adaptively memorised the improvement strengths for each neighbourhood structure. The neighbourhood structure with the best improvement history will be employed to generate neighbour(s) for the current iteration. The hypothesis is, “if the improvement history of neighbourhood structure can affect the performance of neighbourhood structures selection mechanism and subsequently, the performance the applied meta-heuristic, then the meta-heuristic (i.e., Simulated Annealing (SA) in this case) with AD-NS will outperform the meta-heuristic (i.e., SA) with other neighbourhood structure selection mechanisms”. To prove this, the experiment is conducted by applying SA with AD-NS, SA with Token ring and SA with Union neighbourhood structure selection mechanisms; tested on Curriculum-Based Course Timetabling problem for the ITC-2007 track3 benchmark datasets. Results based on the average ranked, shows that SA with AD-NS approach obtained the fourth rank compared with other approaches reported in the literature. Statistical analysis on SA with AD-NS against SA with other neighbourhood structure selection mechanisms proved that the performance of SA with AD-NS is significantly better than SA with other neighbourhood structures selection mechanisms tested in this work. This indicates that the improvement history of neighbourhood structure can affect the performance of neighbourhood structures selection mechanism and subsequently, the performance the applied meta-heuristic.

Key words: Simulated annealing, neighbourhood structure selection mechanism, course timetabling problem, meta-heuristic

INTRODUCTION

The definition of the neighbourhood structure plays a crucial decision in local search algorithm (Lu and Hao, 2010). For example, a good neighbourhood structure can positively influence the performance of Simulated Annealing (SA) (Moscato, 1993). Choosing a suitable neighbourhood structure or the size of neighbourhood are critical decisions, in improving the performance of SA (Eglese, 1990; Fleischer and Jacobson, 1999). Some researchers reported that SA with small neighbourhood size had performed better than large neighbourhood size (Cheh *et al.*, 1991). Whilst, others proved that SA with large neighbourhood size had performed better than the smaller neighbourhood size (Ogbu and Smith, 1990). In addition, combining two or more different

neighbourhoods is a powerful mechanism to increase the performance of searching algorithm (Xinchao, 2011). Thus, employing several neighbourhood structures may produce better results compared to single neighbourhood structure. However, the computational time might be increased due to the number of neighbours that need to be visited and perhaps disconnected neighbourhoods. Indeed, the question is, “how to select a suitable neighbourhood structure to be applied at each iteration?” There are many ways to select the neighbourhood structure, such as union neighbourhood structures (Di Gaspero and Schaerf, 2006) and token-ring search (Glover, 1989) selection mechanisms. Many works have investigated on the effect of neighbourhood selection mechanism to the performance of meta-heuristic search. For example Lu and Hao (2010) compared between union

Corresponding Author: Masri Ayob, Data Mining and Optimization Research Group (DMO), Center of Artificial Intelligence Technology, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, 43600 Bangi, Selangor, Malaysia
Tel: +603 8921 6741 Fax: +603 8921 6184

and token ring combinations (operators/selections) within tabu search meta-heuristic. They found that token ring selection mechanism outperformed union combination for all instances. Avanthay *et al.* (2003) have combined six neighbourhood structures to enhance their proposed Variable Neighbourhood Search (VNS). They switched from one neighbourhood structure to another one, when the current neighbourhood structure cannot improve the solution in several consecutive non improvement iterations. The neighbourhood structure was selected according to the solution quality, which was produced by the VNS algorithm for each neighbourhood structure. Abdullah *et al.* (2010) had applied a Round Robin (RR) algorithm in SA to control the selection of neighbourhood structures called dual-sequence SA (DSA) to solve post enrolment course timetabling problems. They showed that the performance of DSA is comparable with the state-of-the-art techniques. Triki *et al.* (2005) had presented an empirical study on the effect of neighbourhood structure selection to the performance of SA algorithm. Results showed that choosing a suitable neighbourhood is important to the performance of SA. In other work, Zhang *et al.* (2010) have applied SA with a new neighbourhood structure for high school timetabling problems. In order to search for the best neighbour, they performed a sequence of swaps between pairs of time slots. Recently, Kalender *et al.* (2012) have hybridised SA with an improvement oriented heuristic selection mechanism to solve a curriculum based course timetabling problem at Yeditepe University.

This shows that the decision to choose a suitable neighbourhood structure to be applied can affect the performance of applied meta-heuristic. Therefore, this work proposes an adaptive neighbourhood structure selection mechanism in SA algorithm (SA with AD-NS). SA with AD-NS will adaptively switches among the neighbourhood structures, guided by the improvement history of each neighbourhood structure. The applied neighbourhood structure will be rewarded if it can improve the quality of the current solution. Otherwise, it will be penalised to give better chance to the other neighbourhood structures to be selected. The hypothesis for this work is, “if the improvement history of neighbourhood structure can affect the performance of neighbourhood structures selection mechanism and subsequently, the performance of the applied meta-heuristic, then the meta-heuristic (e.g., SA in this case) with AD-NS will outperform the meta-heuristic (i.e., SA) with other neighbourhood structure selection mechanisms”.

PROBLEM DESCRIPTION

This work use the curriculum based course timetabling problem (CB-CTT) that is a variant of an educational timetabling problem, presented in track 3 of the Second International Timetabling Competition-ITC2007 track³ (Di Gaspero *et al.*, 2007), (<http://www.cs.qub.ac.uk/itc2007/>). CB-CTT consisted of scheduling all lectures into weekly timetable, where each course lecture must be assigned to a period and a room in accordance to a given set of constraints, by satisfying the hard constraints and minimising the violation of the soft constraints. Hard constraint is mandatory in order to have a feasible timetable. Whilst, violation of soft constraints should be minimised in order to have good quality timetable.

The Constraints for CB-CTT ITC2007 track³ datasets:

This problem has four hard constraints (H1-H4) and four soft constraints (S1-S4) presented by Di Gaspero *et al.* (2007) as follows:

- **Hard constraints:**

- H1:** Lectures. All lectures of a course must be scheduled to a distinct periods
- H2:** Room Occupancy. Any two lectures cannot be assigned into the same room at the same period
- H3:** Conflicts. Lectures of courses in the same curriculum or taught by the same teacher cannot be schedule into the same period
- H4:** Availability. If the teacher of a course is not available at a given period, then the lectures of the course cannot be assigned to that period

- **Soft constraints:**

- S1:** Room Capacity. For each lecture, the number of students attending the course should not be greater than the capacity of the room hosting the lecture
- S2:** Minimum Working Days. The lectures of a course should be spread into the given minimum number of days
- S3:** Room Stability. All lectures of a course should be scheduled at the same room. If this is impossible, the number of occupied rooms should be as few as possible
- S4:** Curriculum Compactness. For a given curriculum, a violation is counted if there is one lecture not adjacent to any other lecture belonging to the same curriculum within the same day, which means the agenda of students should be as compact as possible

Objective function: To evaluate the quality of the timetable, an objective function as proposed in Lu and Hao (2010) Eq. 1 is used to calculate the violations of soft constraints. Since, this is the minimisation problem, smaller objective value (penalty cost) is better:

$$\min (f(X)) = F1+F2+F3+F4 \quad (1)$$

where, F1 to F4 are the penalties cost for violating the soft constraints (S1-S4). F1 is calculated by subtracting the capacity of the room from the number of students attending the lecture (if the student enrolment for the lecture is greater than the room capacity, i.e., violation of S1). F2 is the violation of the minimum working day for the lectures of a course (i.e., violation of S2). If the lectures of a course are not adequately spread, then the penalty value is calculated. This is done by subtracting the day gap between two lectures of the same course from the minimum numbers of day gap. F3 is the violation of room stability (i.e., violation of S3). If a lecture is assigned to more than one room, the violation is counted by subtracting one from the number of rooms that the lecture is assigned. F4 is the violation of curriculum compactness (i.e., violation of S4). A violation is counted if the lecture is not adjacent to any other lecture belonging to the same curriculum within the same day.

PROPOSED ALGORITHM

This work uses a standard SA but the aim is to enhance the performance of SA by employing several neighbourhood structures and proposes an adaptive neighbourhood structures selection mechanism.

Basic simulated annealing: Simulated Annealing (SA) is one of the oldest meta-heuristic algorithms that have strategy to avoid local minima, by allowing the bad quality solution to be accepted (with some probability). SA always accepts good quality neighbour solution; and probably accepts worse quality solutions using probability acceptance criteria, $P(S_i) = e^{-\Delta f/T_i}$. This probability function control the acceptance of neighbour solution s^* based on the quality of s^* , $f(s^*)$, with regard to the quality of the current solutions, $f(s)$ and the current temperature T_i , where $\Delta f = f(s^*)-f(s)$. The current temperature T_i is reduced using a cooling schedule with a given cooling rate α for each iteration or level, until the temperature reaches final minimum temperature, T_{min} (that is closed to zero). SA stops when T_{min} is reached.

Adaptive neighbourhoods structure selection: This study proposed new neighbourhood selection mechanism

(AD-NS) by adaptively switch among the neighbourhood structures during the search process. The objective of the proposed neighbourhood structure selector is to improve the performance of the search algorithm (i.e., SA in this work), by applying a suitable neighbourhood structure at a given time. Using the AD-NS, the suitable neighbourhood structure with the best improvement history will be employed. To test the idea, this study used two neighbourhoods structures denoted as NS_1 and NS_2 .

NS₁: Move one lecture from the current period to another clash free position period

NS₂: Randomly swaps two different lectures (without violating the hard constraints)

The improvement strength (fitness) (after the n^{th} iteration) of the employed and unemployed neighbourhood structure (at the n^{th} iteration) are calculated using Eq. 2 and 3, respectively:

$$F(N_i)_n = \begin{cases} f(Sol) & \text{if } n = 0 \text{ and } \Delta \neq 0 \\ \sum_{i=0}^{n-1} F(N_i)_0 + \Delta & \text{if } n > 0 \text{ and } \Delta \neq 0 \end{cases} \quad (2)$$

where, n is the number of the employed iterations; i is the i^{th} neighbourhood structure that is employed at the n^{th} iteration; $f(Sol)$ is the quality of the incumbent solution; and Δ is the differences between the quality of the current solution $f(Sol)$ and the trial solution $f(Sol')$, $\Delta = f(Sol')-f(Sol)$.

In order to update the fitness of the unemployed neighbourhood structures and give them fair chance to be selected later, this work proposes Eq. 3 that subtracts Δ from it. In the minimization problem, the value of Δ is negative when the quality of trial solution is better than the quality of the current solution. Therefore, the fitness of the applied neighbourhood structure (N_i) will be stronger (lower value of $F(N_i)_n$) if it managed to generate good quality neighbour (note: The lowest fitness value indicates better fitness), whilst the fitness of the unemployed neighbourhood become weaker. However, the fitness of the unemployed neighbourhood will become stronger if the applied neighbourhood structure failed to generate a good quality neighbour. Therefore, the unemployed neighbourhoods will have better chances to be applied at the next iterations, when the current employed neighbourhood failed to improve the quality of the current solution:

$$F(N_j)_n = \begin{cases} f(Sol) & \text{if } n = 0 \text{ and } \Delta \neq 0 \\ F(N_j)_{n-1} - \Delta & \text{if } n > 0 \text{ and } \Delta \neq 0 \\ F(N_j)_{n-1} + \rho & \text{if } n > 0 \text{ and } \Delta = 0 \end{cases} \quad (3)$$

```

Let :
Sol be an initial solution; //generated using any constructive heuristic
f(Sol) as the quality of Sol;
Sol* as the best found solution;
n as the number of iterations;
T0 as the initial temperature; //is determined using a dynamic initial temperature
Ti as the current temperature;

Given an initial solution Sol as an input;
1. Initialisation:
2. Set n = 0, Sol* = Sol, f(Sol*) = f(Sol), Ti = T0;
3. Generate k neighbours from Ni neighbourhood structures (NS1 and NS2);
4. Select the best neighbour generated neighbour from NS1, Sol1 and set F(N1)0 = f(Sol1);
5. Select the best neighbour generated neighbour from NS2, Sol2 and set F(N2)0 = f(Sol2);
6. Do while (termination criteria does not met)
7. Select suitable neighbourhood structure, Ni that has the best fitness // (calculated using equations 1 and 2)
8. Generate k neighbours from Ni neighbourhood structures and choose the best neighbour, Sol1;
9. Calculate  $\Delta = f(\text{Sol}^1) - f(\text{Sol})$ ;
10. Update the fitness (improvement strength for all neighbourhood structures using Eq.1 and 2)
11. If  $\Delta = 0$  then // Improved solution.
12. Sol = Sol1 // Update the current solution
13. If f(Sol1) < f(Sol*) // Sol1 has better quality than Sol*.
14. Sol* = Sol1 // Update the best solution so far.
15. End if
16. Else
17. Generate a random number  $\zeta$  between [0, 1];
18. If ( $e^{-\Delta/\zeta T_i} > \eta$ ) then //The solution Sol1 is accepted
    Sol = Sol1
19. End if
20. End Else
21. Update the temperature, Ti; //Using adaptive cooling schedule as in (Tarawneh et al., 2013)
22. n = n+1
23. End while;
24. Output the final solution, Sol*;
    
```

Fig. 1: The pseudo code of the SA with AD-NS selection mechanism

where, $i \neq j$ and j is the unemployed neighbourhood structure at the n^{th} iteration and ρ is a random number between 0 and 1.

When Δ is equal to zero, the unemployed neighbourhood structure improvement history will be updated by subtracting a random number between 0 and 1 from its fitness value, in order to give the unemployed neighbourhood structure a higher chance to be selected at the next iteration. Whilst, the fitness of the employed neighbourhood structure (that was not performed well), is not changed. AD-NS applies one neighbourhood per iteration. Thus, the neighbourhood structure with the best fitness (i.e., the best improvement history/smallest fitness value) will be applied at the current iteration.

Simulated annealing with adaptive neighbourhood structures selection: Figure 1 shows the pseudo-code for SA that implemented with AD-NS selection mechanism. In this study, the initial temperature is initialized dynamically using a dynamic initial temperature by applying several iterations (at the early stage) and calculate the deviation average to determine a suitable initial temperature for each instance. For the cooling schedule, the amount of

reducing the temperature value is adaptively adjusted according to the total number of iterations and the initial temperature in order to decide the decrement amount during the search process as in Tarawneh *et al.* (2013).

SA starts once the initial solution Sol is generated (randomly or using constructive technique). For the first iteration, both NS₁ and NS₂ neighbourhood structures are employed to equally generate k neighbour solutions of Sol. Then the best quality neighbours of each neighbourhood are selected to update the improvement strength (fitness) of each neighbourhood structure, respectively, using Eq. 1. After the first iteration, the neighbourhood structure with the best (minimum) fitness (calculated using Eq. 1 and 2) is selected to generate k neighbour solutions of Sol. The best neighbour, Sol¹ that is generated by the selected neighbourhood structure is then selected for calculating the Δ (where $\Delta = f(\text{Sol}^1) - f(\text{Sol})$). The Δ is used to calculate the fitness of all the neighbourhood structures and acceptance criterion of the SA. The fitness of all neighbourhood structures (employed and non-employed) are updated using Eq. 1 and 2, accordingly. From line 11 until 24 in Fig. 1, the SA with AD-NS works similar as standard SA algorithm.

Statistical method: Since SA is a stochastic algorithm, the quality of the generated solutions are not normally distributed. This was tested using Kolmogorov-Smirnov. Therefore, the Wilcoxon Signed-Rank Test (i.e., a non-parametric test) with a 95% confidence level, is used for a statistical analysis on the results obtained by the tested approaches. This is performed to validate the Alternative Hypothesis, which is:

“If the improvement history of neighbourhood structure can affect the performance of neighbourhood structures selection mechanism and subsequently, the performance the applied meta-heuristic, then the meta-heuristic (i.e., SA in this case) with AD-NS will outperform the meta-heuristic (i.e., SA) with other neighbourhood structure selection mechanisms.”

EXPERIMENTAL SETUP AND RESULTS

Table 1 shows the parameters setting for the SA algorithms tested in this work. These parameter values were determined based on preliminary test. The

Table 1: Parameters values used for SA algorithms

Parameters	Method/value
Temperature	Dynamic initial temperature
Cooling schedule	Adaptive as in (Tarawneh <i>et al.</i> , 2013)
Termination condition	460 sec
Number of neighbours to be generated for each iteration, k	*P/8 neighbours, where P is the number of the free position or the total number of lectures.

The neighbours size is determined based on the primarily experiments

experiments have been carried out (with 31 independent runs) to investigate the search capability for AD-NS in SA algorithm. For comparison, this study used the standard SA as in Abramson (1991) with dynamic initial temperature and adaptive cooling schedule.

Table 2 shows a comparison results among SA with AD-NS ($\Delta = f(\text{Sol}')-f(\text{Sol})$), SA with token-ring (NS1→NS2) and SA with union (NS1 U NS2) neighbourhood structures selections under the ITC2007 track³ timeout condition (460 sec in this machine). It shows that AD-NS with SA outperformed the other neighbourhood structures selections across in all instances, (for all best, worst, mean and the standard deviation). A statistical analysis on the results (Table 3) is performed using Wilcoxon Signed-Rank Test in order to support the Alternative Hypothesis, which is:

“If the improvement history of neighbourhood structure can affect the performance of neighbourhood structures selection mechanism and subsequently, the performance the applied meta-heuristic, then the meta-heuristic (i.e., SA in this case) with AD-NS will outperform the meta-heuristic (i.e., SA) with other neighbourhood structure selection mechanisms.”

Table 3 showed that (in general) the performance of SA with AD-NS is statistically significant better than the performance of SA with other two neighbourhood selection mechanisms (token-ring and union). SA with AD-NS obtained better quality of solutions almost in all instances (except comp01 and comp11). The significance

Table 2: Experimental results of AD-NS, token ring and union neighbourhood structures combinations in SA over 31 independent runs on ITC2007 track³ datasets

Instances	(Token ring) NS ₁ →NS ₂				(Union) NS ₁ U NS ₂				(AD-NS) NS ₁ ↔NS ₂			
	Best	Worst	Mean	STD	Best	Worst	Mean	STD	Best	Worst	Mean	STD
Comp01	5	6	5.06	0.250	5	8	5.23	0.617	5	5	5.00	0.000
Comp02	86	105	97.35	5.684	87	111	102.16	6.573	85	103	94.65	5.794
Comp03	93	121	104.81	8.023	93	134	120.19	9.537	93	114	102.09	6.753
Comp04	65	85	76.35	6.509	65	90	79.97	7.570	63	83	71.72	6.806
Comp05	330	350	340.48	5.372	334	356	346.48	6.297	328	343	335.58	5.461
Comp06	84	120	102.00	9.778	86	126	111.71	10.802	84	116	98.39	9.084
Comp07	54	71	63.65	5.251	57	83	70.52	6.757	51	67	60.32	4.956
Comp08	54	76	66.94	6.303	60	85	75.00	6.723	52	71	62.30	5.702
Comp09	143	169	157.94	7.550	149	183	164.68	8.179	140	161	150.82	6.543
Comp10	36	53	44.45	5.824	40	63	53.94	6.418	31	48	40.71	5.100
Comp11	0	1	0.16	0.374	0	2	0.45	0.624	0	1	0.09	0.288
Comp12	341	369	356.42	8.955	342	379	362.81	11.368	336	359	347.29	7.773
Comp13	95	120	106.58	7.865	96	127	110.77	10.494	91	115	101.71	6.357
Comp14	77	100	88.61	7.680	79	109	93.87	7.991	75	96	85.74	6.976
Comp15	83	107	94.74	6.831	86	116	97.39	8.277	80	102	90.48	6.683
Comp16	60	88	74.74	8.185	62	94	77.16	9.505	56	85	70.55	9.352
Comp17	84	103	94.58	6.071	88	119	98.61	8.589	82	101	91.81	5.845
Comp18	94	120	104.84	7.453	99	129	111.39	8.640	89	115	98.90	7.222
Comp19	91	121	105.87	8.385	96	127	111.06	9.405	90	117	103.19	7.543
Comp20	46	68	56.26	6.309	47	76	60.71	9.023	45	61	52.76	5.852
Comp21	103	127	114.97	6.730	102	135	117.23	8.593	102	124	111.19	6.096

Best results in bold, All the figures in column “Best”, “Worst” and “Mean” representing the quality of solutions for best, worst and mean for 31 runs

level in this test is 0.05, i.e., if p value is less than 0.05 then, the performance of SA with AD-NS is statistically significant better compared to token ring and union neighbourhood structures selection mechanisms. Results supported the hypothesis that AD-NS can enhance the

performance of SA compared to token ring and union neighbourhood structures selections mechanisms (i.e., the alternative hypothesis is accepted).

Finally, the performance of SA with AD-NS is compared against other approaches from the literature (Table 4) and the best know results under ITC2007 Track³ timeout condition. Table 4 showed that the SA with AD-NS can produce good quality solution compared with other approaches in the ITC 2007 Track³, under the time out condition which is equal to 460 seconds in this work. For example, SA with AD-NS managed to obtain the best known results in the comp1 and comp11 (small instances). For comp17 and 21, SA with AD-NS obtained the best results in comparison with other approaches compared in Table 4. Furthermore, it outperformed (Geiger, 2012; Clark *et al.*, 2008) approaches for all instances.

According to the average results (Avg) in Table 4, the approaches in Table 4 is ranked. The smaller average ranked is better. For example, Muller (2009) is ranked first for 15 instances out of 21, (Lu and Hao, 2010) ranked second and SA with AD-NS approach obtained the fourth rank compared with others. According to the average ranked in Table 4, the SA with AD-NS approach demonstrated that AD-NS selection mechanism can enhance the performance of SA.

Table 3: Statistical analysis (Wilcoxon Signed-Rank Test) for AD-NS against token ring and union neighbourhood structures selections in SA tested on ITC 2007 track³ datasets

Instances	AD-NS vs token ring	AD-NS vs union
	p-value	p-value
Comp01	0.051	0.016
Comp02	0.003	0.000
Comp03	0.001	0.000
Comp04	0.001	0.000
Comp05	0.001	0.000
Comp06	0.003	0.000
Comp07	0.002	0.000
Comp08	0.000	0.000
Comp09	0.000	0.000
Comp10	0.001	0.000
Comp11	0.059	0.016
Comp12	0.000	0.001
Comp13	0.001	0.000
Comp14	0.005	0.001
Comp15	0.001	0.000
Comp16	0.003	0.001
Comp17	0.001	0.015
Comp18	0.000	0.000
Comp19	0.005	0.000
Comp20	0.000	0.002
Comp21	0.001	0.000

Table 4: Average and best results of the SA with AD-NS in comparison with the top five competitors and best know results from ITC 2007 Track³

Rank	SA with AD-NS		(Muller, 2009)		(Lu and Hao, 2010)		M1		(Geiger, 2012)		(Clark <i>et al.</i> , 2008)		Best Known*
	Avg	Best	Avg	Best	Avg	Best	Avg	Best	Avg	Best	Avg	Best	
	4		1		2		3		5		6		Best Known*
Comp01	5.00	5	5.0	5	5.0	5	5.1	5	6.7	5	27.0	10	5
Comp02	94.65	85	61.3	51	61.2	55	65.6	50	142.7	111	131.1	111	24
Comp03	102.09	93	94.8	84	84.5	71	89.1	82	160.3	128	138.4	119	66
Comp04	71.72	63	42.8	37	46.9	43	39.2	35	82.0	72	90.2	72	35
Comp05	335.58	328	343.5	330	326.0	309	334.5	312	525.4	410	811.5	426	290
Comp06	98.39	84	56.8	48	69.4	53	74.1	69	110.8	100	149.3	130	27
Comp07	60.32	51	33.9	20	41.5	28	49.8	42	76.6	57	153.4	110	6
Comp08	62.30	52	46.5	41	52.6	49	46.0	40	81.7	77	96.5	83	37
Comp09	150.82	140	113.1	109	116.5	105	113.3	110	164.1	150	148.9	139	96
Comp10	40.71	31	21.3	16	34.8	21	36.9	27	81.3	71	101.3	85	4
Comp11	0.09	0	0.0	0	0.0	0	0.0	0	0.3	0	5.7	3	0
Comp12	347.29	336	351.6	333	360.1	343	361.6	351	485.1	442	445.3	408	300
Comp13	101.71	91	73.9	66	79.2	73	76.1	68	110.4	622	122.9	113	59
Comp14	85.74	75	61.8	59	65.9	57	62.3	59	99.0	90	105.9	84	51
Comp15	90.48	80	94.8	84	84.5	71	89.1	82	160.3	128	138.0	119	66
Comp16	70.55	56	41.2	34	49.1	39	50.2	40	92.6	81	107.3	84	18
Comp17	91.81	82	86.6	83	100.7	91	107.3	102	143.4	124	166.6	152	56
Comp18	98.90	89	91.7	83	80.7	69	73.3	68	129.4	116	126.8	110	62
Comp19	103.19	90	68.8	62	69.5	65	79.6	75	132.8	107	125.4	111	57
Comp20	52.76	45	34.3	27	60.9	47	65.0	61	97.5	88	179.3	144	4
Comp21	111.19	102	108.0	103	124.7	106	138.1	123	185.3	174	185.8	169	75

Best known results in bold; while the best results among approaches in italic, Hybrid approach (Muller, 2009), TS (Lu and Hao, 2010), M1: Third place winner for the ITC2007 track3 (Note: There is no publication for this work except in the ITC2007 track3 web site) (http://www.cs.qub.ac.uk/itc2007/index_files/finalists.htm), Threshold accepting local search (Geiger, 2012), Repair-based Local Search (Clark *et al.*, 2008), Avg: Average, Best : Best, Best*: Best Known results under ITC2007 track3 timeout condition

CONCLUSION

This work proposed a new adaptive neighbourhood structures selection (AD-NS) mechanism to enhance the SA performance by adaptively select the suitable neighbourhood structure during the search process according to each neighbourhood structure improvement history. The performance of AD-NS in SA was evaluated by testing the approach on curriculum based course timetabling problem that was presented in track 3 of the Second International Timetabling Competition (ITC2007 track³). Results demonstrated that AD-NS can enhance the performance of SA compared to token ring and union neighbourhood structures selections mechanisms. This indicates that the improvement history of neighbourhood structure can affect the performance of neighbourhood structures selection mechanism and subsequently, the performance of the applied meta-heuristic. For the future work, AD-NS will be applied using more neighbourhood structures in other local search algorithms such as variable neighbourhood structure algorithm.

ACKNOWLEDGMENT

The authors wish to thank Ministry of Higher Education for supporting this work under the FRGS Research Grant Scheme (FRGS/1/2012/SG05/UKM/02/11).

REFERENCES

- Abdullah, S., K. Shaker, B. McCollum and P. McMullan, 2010. Dual sequence simulated annealing with round-robin approach for university course timetabling. *Evol. Comput. Comb. Optim.*, 6022: 1-10.
- Abramson, D., 1991. Constructing school timetables using simulated annealing: Sequential and parallel algorithms. *Manage. Sci.*, 37: 98-113.
- Avanthay, C., A. Hertz and N. Zufferey, 2003. A variable neighborhood search for graph coloring. *Eur. J. Oper. Res.*, 151: 379-388.
- Cheh, K.M., J.B. Goldberg and R.G. Askin, 1991. A note on the effect of neighborhood structure in simulated annealing. *Comput. Oper. Res.*, 18: 537-547.
- Clark, M., M. Henz and B. Love, 2008. Quikfix a repair-based timetable solver. *Proceedings of the 7th International Conference on the Practice and Theory of Automated Timetabling*, August 18-22, 2008, Montreal, Canada.
- Di Gaspero, L. and A. Schaerf, 2006. Neighborhood portfolio approach for local search applied to timetabling problems. *J. Math. Model. Algorith.*, 5: 65-89.
- Di Gaspero, L., B. McCollum and A. Schaerf, 2007. The second international timetabling competition (ITC-2007): Curriculum-based course timetabling (track 3). *Proceedings of the International Workshop on 7th International Conference on Automated Planning and Scheduling*, September 22-26, 2007, Rhode Island, USA.
- Eglese, R.W., 1990. Simulated annealing: A tool for operational research. *Eur. J. Oper. Res.*, 46: 271-281.
- Fleischer, M. and S.H. Jacobson, 1999. Information theory and the finite-time behavior of the simulated annealing algorithm: Experimental results. *INFORMS J. Comput.*, 11: 35-43.
- Geiger, M.J., 2012. An application of the threshold accepting metaheuristic for curriculum based course timetabling. *Ann. Oper. Res.*, 194: 189-202.
- Glover, F., 1989. Tabu search-Part I. *ORSA J. Comput.*, 1: 190-206.
- Kalender, M., A. Kheiri, E. Ozcan and E.K. Burke, 2012. A greedy gradient-simulated annealing hyper-heuristic for a curriculum-based course timetabling problem. *Proceedings of the 12th UK Workshop on Computational Intelligence*, September 5-7, 2012, Edinburgh, USA., pp: 1-8.
- Lu, Z. and J.K. Hao, 2010. Adaptive tabu search for course timetabling. *Eur. J. Oper. Res.*, 200: 235-244.
- Moscato, P., 1993. An introduction to population approaches for optimization and hierarchical objective functions: A discussion on the role of tabu search. *Ann. Oper. Res.*, 41: 85-121.
- Muller, T., 2009. ITC2007 solver description: A hybrid approach. *Ann. Oper. Res.*, 172: 429-446.
- Ogbu, F.A. and D.K. Smith, 1990. The application of the simulated annealing algorithm to the solution of the $n/m/C_{max}$ flowshop problem. *Comput. Oper. Res.*, 17: 243-253.
- Tarawneh, H.Y., M. Ayob and Z. Ahmad, 2013. A hybrid simulated annealing with solutions memory for curriculum-based course timetabling problem. *J. Applied Sci.*, 13: 262-269.
- Triki, E., Y. Collette and P. Siarry, 2005. A theoretical study on the behavior of simulated annealing leading to a new cooling schedule. *Eur. J. Oper. Res.*, 166: 77-92.
- Xinchao, Z., 2011. Simulated annealing algorithm with adaptive neighborhood. *Applied Soft Comput.*, 11: 1827-1836.
- Zhang, D., Y. Liu, R.M. Hallah and S.C.H. Leung, 2010. A simulated annealing with a new neighborhood structure based algorithm for high school timetabling problems. *Eur. J. Operational Res.*, 203: 550-558.