Investigation of simulated annealing components to solve the university course timetabling problem

Issam AlHadid*
Faculty of Computer Information Systems
University of Jordan
Jordan
issamfe@hotmail.com

Khalid Kaabneh
Faculty of Information Technology
Al-Ahliyya Amman University
Jordan
dr_kaabneh@yahoo.com

Hassan Tarawneh
Faculty of Information Technology
Al-Ahliyya Amman University
Jordan
hassan_trawneh@yahoo.com

Aysh Alhroob
Faculty of Information Technology
Isra University
Jordan
aysh@iu.edu.jo

Abstract. Simulated Annealing (SA) is a common meta-heuristic algorithm that has been widely used to solve complex optimization problems. This is due to its ease of implementation and capability to escape from local optimum. This research conducts an investigation on three of SA components: the initial temperature, cooling schedule and neighborhood structure. We observed that the high initial temperature leads SA to accept any solution (wasting more computational time), whilst the lower value leads SA to quickly trap in local optimum. Based on research findings from this phase, for each component we suggested a technique to overcome the limitations. The limitations are: (i) a dynamic initial temperature mechanism that dynamically chose the suitable initial temperature for each instance problem; (ii) adaptive cooling schedule that will adjust the temperature value during the search; and (iii) a new neighborhood structure that will improve the search ability by minimizing the random selection. In the second phase. The experimental results show that the proposed techniques and approaches in all phases have outperformed the SA and comparable to other approaches in the literature (tested on university course timetabling benchmark dataset ITC2007-Track3).

Keywords: simulated annealing, meta-heuristic, university course timetabling problem, optimization, ITC2007-Track3.

∗. Corresponding author

1. Introduction

Timetabling is considered as a unique case of scheduling problems [23]. Wren [23] defined the timetabling as follows:

> "The allocation, subject to constraints, of given resources to objects being placed in spacetime, in such a way as to satisfy as nearly as possible a set of desirable objectives".

Schaerf [18] categorized the educational field into three common timetabling problems: university course timetabling problem, examination timetabling problem and school timetabling problem. All of them may different but they still have the same fundamental characteristics. For example, the sizes of class for all school lectures are normally similar and have the same group of students in common that are associated with the same courses, meanwhile the university courses have different number of students' enrolments. Table (1) summarizes some of the common meta-heuristic algorithms related to different categories which are presented by Birattari et al. [4], where $\surd$ means that the algorithm is totally categorized under this category and x means it does not, whilst ¬ means it is a partly related to this category.

Table 1: Summary of the meta-heuristic algorithms characteristics.

Category	SA	TS	GA	ACO	ILS	grasp	GLS
Trajectory	$\surd$	$\surd$	x	x	x	x	$\surd$
Population	x	x	$\surd$	$\surd$	x	x	
Memory	x	$\surd$	¬	$\surd$	¬	x	$\surd$
Various neighborhood	x	x	¬	x	$\surd$	x	x
Dynamic	x	¬	x	x	x	x	$\surd$
Nature inspired	$\surd$	x	$\surd$	$\surd$	x	x	x

Note: ACO: ant colony optimization; TS: Tabu search; ILS: iterated local search; GRASP: greedy randomized adaptive search procedure; SA: simulated annealing; GLS: guided local search, GA: genetic algorithm.

SA is a trajectory-based algorithm or a single point algorithm [4]. The single point algorithms consider a single element of the solution space at each iteration by jumping from position to another in the solution space.

SA uses a special strategy to escape from local optimum. At first, SA will explore the search space (diversification) by easily accepting worse solution based on probability acceptance criterion. Later on, the probability is decreased during the search space process that leads to focus more on promising regions (intensification) [1]. In fact, the acceptance probability depends on the temperature which is reduced during the search process to give balance between the diversification and intensification. However, SA could be trapped into local optimum and may consume a longer time to find good solutions [24]. Hence, many researchers attempted to improve the SA performance by using different approaches, such as adaptive SA (ASA) suggested by Ingber [10], or by

hybridize SA with other meta-heuristics approaches, such as genetic algorithm [21]. In fact, hybrid techniques can achieve different advantages; such as the best performance; by combining the advantages of each individual technique into one hybrid algorithm. Therefore, this research is motivated to investigate the process of SA performance improvement by escaping from local optimum and reducing the computational time.

This research focuses on improving a SA performance by investigating the effect of SA components: temperature, cooling schedule and neighborhood structure on the SA performance. The good initial temperature will balance the diversification and intensification properly by avoiding the waste of computational time when the initial temperature is high or when trap in local optimum which is a result of a very low temperature [22].

Elmohamed et al. [8] claimed that adjusting the decrement amount of the given temperature during the search process is an important issue, in order to avoid the wasting of computational time. Furthermore, The high temperature with slow cooling schedule leads SA to accept many bad solutions (a zigzag walk), while the low temperature with fast cooling schedule leads SA to trapped quickly in the local optimum by rejecting most of the unimproved solutions and accepting the improved solutions only (descent method). Meanwhile, an effective neighborhood structures may easily improve the solution quality [14]. Moreover, choosing the most suitable neighborhood is an important issue to avoid the discounted neighborhood and save the computational time [15].

AlHadid et al. [2] proposed a hybrid SA with EMC technique to divert the search effectively to another promising region by escaping the search space from local optimum to another promising region space. AlHadid et al. [2] stated that the proposed technique results has outperformed the standard SA and gave comparable results to other approaches when tested on ITC2007-Track3 university course timetabling datasets.

According to the previous dissection, we have identified two main research questions (in regard to SA algorithm) to be answered:

a) How can we enhance the SA performance by escaping from local optimum?

b) How can we reduce the SA computational time?

In order to answer the main research questions, we have identified four research questions which need to be answered, the questions are:

Q1) How to initialize the initial temperature dynamically based on the problem instance individually?

Q2) What is the effective cooling schedule that can adaptively adjust the amounts of temperature decrement during the SA search process?

Q3) How to design a good neighborhood structure that can help SA to effectively search for good quality solution and minimize the random swap and move?

Q4) How to choose a good neighborhood structure among different neighborhood structures during the search process that is able to avoid the disconnected neighborhood?

2. Related work

2.1 Dynamic initial temperature

This section presents Dynamic Initial Temperature (D-IT) which proposed by Tarawneh et al. [20] to improve performance of SA by initializing the initial temperature dynamically. Normally, when the temperatures are very high the search space of SA could be very wide by accepting many worse solutions. Meanwhile, the range of accepting worse solution(s) becomes very small when the initial temperature is low.

The proposed D-IT mechanism by Tarawneh et al. [20] is used to estimate the suitable range of accepting worse solution at early stage, by performing a preliminary experiment to clarify the acceptance ratio related to different worse solutions for each temperature and deviation average γ , where γ is calculated using the following equation:

$$\gamma = \frac{\sum_{i=1}^{n} |\Delta f|}{n} \tag{1}$$

Where $\Delta f = f(s^*) - f(s)$ *and* n *is the total current iterations*

Table 2: The acceptance ratio for different averages deviation under different temperatures ranges

		Temperatures									
		10000000000	1000000	10000	5000	2000	1000	500	100	10	1
γ	150	100%	100%	99.0%	95.8%	94.3%	85.5%	71.5%	27.4%	0.0%	0.0%
	50	100%	100%	99.0%	98.9%	97.6%	95.5%	88.3%	28.9%	0.0%	0.0%
	5	100%	100%	99.0%	99.0%	99.9%	99.9%	97.5%	94.9%	70.5%	0.0%

From Table (2), the high temperatures (i.e. T= 10000000000) lead the SA to accept every worse solution. Such as, when the is equal 150, 50 or 5. Namely, high temperature makes the SA to waste more computational time by accepting every worse bad solution. In this study, we select the initial temperature value based on the average deviation of the penalty value γ (see Eq.1). However, according to the preliminary experiment, we suggest several initial temperatures ranges to select the suitable one among them according to the average deviation of the penalties value that will be calculated for several iterations during the SA process. The initial temperatures range in Table 3.

Table 3: The initial temperature selection

γ	T_0
50 and more	Random (1000, 2000)
5 to 49	Random (500, 1000)
1 to 4	Random (100, 500)

Note: γ is the deviation average of the penalties

Table 4: The differences between the dynamic initial temperatures with other from the literature

Technology	First Initial Temperature	Average(Y)	Comments	Disadvantages
Dynamic	Dynamic	$\lvert\gamma\rvert$	For several consecutive iterations, inside the SA and for every Δ (increase or decrease)	Need a good chosen cooling schedule.
SA with Estimated Temperature [16]	Infinite	$\overrightarrow{\lvert\gamma\rvert}$	For every iteration (for each current solution neighbors), when is increased	The temperature is not control parameter anymore, but the acceptance probability.
Solving the Course Scheduling problem Using SA [3]	10000	$\overrightarrow{\lvert\gamma\rvert}$	Before the SA starts, for several iteration, when is increased	The initial temperature still high.

Note $\overleftrightarrow{\lvert\gamma\rvert}$ means that the deviation average for (positive and negative values).

From Table 4, we found that the other researchers' techniques have some limitations such as initializing high initial temperature values even when they used a small neighbors size, also the temperature becomes uncontrolled SA parameter anymore when it is estimated for each iteration, which means that the problem of setting the initial temperature value is still exists. Therefore, the dynamic Initial Temperature overcomes these problems by choosing the initial temperature value properly. Where the initial temperature T_0 is selected from the Table 2 suggested according to preliminary experiment from Table 1.

2.2 Adaptive Cooling Schedule (ACS)

The performance of SA depends heavily on temperature cooling schedule [5], where the good cooling schedule is substantial feature to get the optimal solution and to reduce the consuming time [17].

In this research, two cooling schedule proposed in the literature review are investigated in order to identify the good and suitable one as follow:

1. The first one is called Static or Geometric cooling schedules, which proposed by Kirkpatrick et al. [11] as the following equation:

   T_k+1=α T_k; where ($0 < \alpha < 1$) denotes the cooling rate or factor; T_k is the current initial temperature.

2. The second one is called Adaptive cooling schedules presented by Lewis et al. [12] where the temperature is decremented to be adjusted during the process of the algorithm according to the objective function values which is used to decide amount of temperature decreasing as following equations:

$$\lambda_0 = 1 - \beta \tag{2}$$

$$\lambda_{i+1} = \lambda_i + \frac{\beta + \beta}{M} \tag{3}$$

$$T_{i+1} = T_i - \lambda_{i+1}(\frac{T_0}{M}) \tag{4}$$

Where β represents a parameter to determine the value of λ which influence the concavity amount that presented in cooling schedule and M represents the iteration number used to decrease the initial temperature T_0 to a zero value or close to it. However, this work investigates both geometric and adaptive cooling schedules, which were presented by Kirkpatrick et al. [11] and Lewis et al. [12] In order to identify the suitable cooling schedule for our study.

2.3 Neighborhood structure

Lü and Hao [13] claimed that neighbourhood structure is one of the most important features of the local search algorithms, where the effective and good neighbourhoods structure influence the SA performance positively [14]. Furthermore, Fleischer and Jacobso [9] stated that to improve the solution quality and the SA performance we must chose a good neighbourhood structure and size. Also, [7] mentioned that:

> A smooth topology with shallow local minima which imposed by a neighbourhood structure is preferred, rather than a bumpy topology with many deeply local minima

Several researchers claimed that SA performance is better when the neighbourhood structure size is relatively small [6]. Meanwhile, other researchers demonstrated that SA performance improves when a large neighbourhood structure is used [15].

In this work, we use three neighbourhood structures in order to improve the solution quality, two neighbourhood structures are common (simple move as NS1 and simple swap as NS2), and a new proposed neighbourhood structure NS_3 which is presented by Tarawneh et al. [20].

NS_3: Generally, the selection mechanism in neighbourhood structure happens randomly (e.g.: - select two lectures randomly which belong to two different

rooms and slots). Thus, the search may need more time to reach a good solution. Tarawneh et al. [20] presented a new neighbourhood structure to reduce the random selection. NS_3 Neighbourhood structure mechanism calculates the total soft constraints penalties for each timeslot and sums them up for the whole week (Table 4 as example). Then, the lectures with the highest penalty swap with other lectures randomly.

Table (4) shows that the presented neighbourhood structure mechanism selects the timeslot with highest penalties (i.e. timeslot= three), and randomly select another timeslot. Then it interchanges both of them with any conflict. Therefore, the soft constraints violation weight (penalties) is calculated for each timeslot every time when a new neighbourhood structure is applied. However, this neighbourhood is used to minimize the random selection in order to avoid the disconnected neighbourhood structure that leads the search to escape from local optimum solution. it works as a flipping procedure for the SA search space.

Table 5: The soft constraints penalties for each time period and slot

	Room1					Room2					Room3					
	Days(1-5)					Days(1-5)					Days(1-5)					Penalties
Timeslots (1-6)	0	5	6	30	0	100	0	1	4	2	10	20	0	1	160	339
	6	15	10	0	0	2	4	2	30	100	1	0	60	1	3	243
	10	0	80	10	0	1	20	1	1	1	3	0	0	0	0	127
	10	1	4	5	2	0	10	0	10	20	6	4	150	1	10	233
	1	2	8	10	0	20	30	15	0	1	2	2	6	0	9	106
	2	3	6	1	8	120	30	40	0	4	0	0	9	0	1	224

Figure 5 shows the SA pseudo code, which contain D-IT, A-CS and the new neighbourhood structure.

Hint: for the parameters settings, we setup them for each part of our experiments separately.

3. Experiment and result

3.1 Experimental results of the Adaptive Cooling Schedule (ACS), D-IT with adaptive neighborhood structure

In the next experiment we compare the geometric cooling schedule (GCS) presented by Kirkpatrick et al. [11] with static initial temperature and standard neighbourhoods structure N1 and N2 against the adaptive cooling seclude (ACS) presented by Lewis et al. [12] with D-IT presented by Tarawneh et al. [20] and A-CS presented by Tarawneh and Ayob [19]. The parameters setting that we used in this experiment are presented in Table 6.

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

NS2: Randomly swaps two different lectures from different time slots and rooms.

NS3: Adaptive neighborhood structure presented by Tarawneh et al. [20].

Enhanced SA pseudo code

Given an initial solution, Sol;
Set Best solution Sol*=Sol;
Set the iteration counter, i = 0;
Set the initial deviation average, = zero;
Set the initial temperature counter iterations, k;
Set the minimum temperature, T_min;
while *termination criteria does not met* **do**
  i=i+1 // update the iterations counter.
  Generate k neighbors from N neighborhood structures;
  Assign the best neighbor to Sol";
  Calculate $\Delta = f(Sol'') - f(Sol)$
  //Calculate the quality deference between the new solution and the current solution.
  if $i < k$ **then**
    $\gamma = \frac{\gamma + abc(\Delta)}{i}$
    // Calculate the deviation average.
    Select T_0 according to Table 3.
  end
  if $\Delta \leq 0$ **then**
    // New solution.
    $Sol \leftarrow Sol''$ *// Update the current solution.*
    if $f(Sol'') < f(Sol*)$ **then**
      // Sol" has better quality than Sol.*
      $Sol'' \leftarrow Sol*$ *// Update the best solution so far.*
    end
  else
  end
  Generate a random number η *between [0, 1];*
  if $(e^{-\Delta f/T_i}) > \eta$ **then**
    // The solution Sol is accepted.
    $Sol \leftarrow Sol''$
  end
  Update the temperature; //A-CS **end**

Table 6: SA parameters setting (standard and enhanced comparison)

Parameters	Standard SA(S-SA)	Enhanced SA(E-SA)
Initial Temperature	1000000	D-IT
Neighborhoods structure	NS1 and NS2	NS3
Cooling rate	0.99	Adaptive
Termination condition	460 seconds	460 seconds

Table (6) illustrates the results of S-SA and E-SA. The first column indicates the instances, columns 2-5, 6-9 reports the best and worst solutions, Mean, and standard deviation over 31 runs for each instance and Column 10 reports the P-value.

Table 7. Computational results of the SA performance using standard components (S-SA) and Enhanced components(E-SA).

Table (7) shows that E-SA improves the SA performance compared to S-SA. SA permanence is statistically significant performance depending on enhanced simulated annealing components (in all instances except the small instances comp1 and comp11). Thus, we conclude that adaptive cooling schedule and

dynamic initial temperature with adaptive neighborhood structure will leads the SA to improve the solution quality compared to SA with standard and static components. *Note: Best results in italic bold.*

Table 7 : Enhanced components on ITC2007track[3] datasets

	E-SA				S-SA				
Instances	Best	Max.	Mean	STD	Best	Max.	Mean	STD	P-Value
Comp01	***5***	6	5.09	0.288	5	9	5.26	0.773	0.071
Comp02	***86***	105	97.35	5.684	88	108	99.32	4.962	0.001
Comp03	***93***	121	104.81	8.023	105	134	121.06	8.095	0
Comp04	***65***	83	73.27	5.806	67	90	78.23	6.893	0.01
Comp05	***330***	350	340.48	5.372	338	356	344.78	5.789	0.001
Comp06	***84***	120	102	9.778	89	115	102.77	6.869	0.001
Comp07	***54***	71	63.65	5.251	57	82	68.84	6.187	0
Comp08	***54***	76	66.94	6.303	59	85	74.87	6.994	0
Comp09	***143***	169	157.94	7.550	149	176	161.10	8.010	0.002
Comp10	***36***	53	44.45	5.824	43	63	54.65	5.851	0
Comp11	***0***	1	0.15	0.359	0	2	0.45	0.624	0.102
Comp12	***341***	369	356.42	8.955	355	385	372.16	8.214	0
Comp13	***95***	119	105.09	7.296	96	127	111.00	10.237	0.025
Comp14	***77***	100	88.61	7.680	81	119	100.48	10.875	0
Comp15	***83***	107	94.74	6.831	87	123	103.10	9.724	0.001
Comp16	***60***	84	72.15	7.652	68	90	78.55	5.579	0.001
Comp17	***84***	103	94.58	6.071	88	119	102.06	9.536	0.002
Comp18	***94***	120	104.84	7.453	100	123	111.71	6.953	0.002
Comp19	***91***	121	105.87	8.385	85	125	111.29	8.814	0.009
Comp20	***46***	68	56.26	6.309	50	74	62.55	6.985	0.003
Comp21	***103***	122	112.42	6.656	111	135	118.58	5.993	0

4. Conclusion

In this paper, we employed the standard SA components to solve the university course timetabling (UCT) problem using the benchmark dataset ITC2007-Track3. We investigated the SA components as follows: initial temperature, cooling schedule and neighborhood's structure, by initializing the initial temperature dynamically and identifying a good cooling schedule to decrease the temperature properly. Moreover, we worked to improve the SA solution quality using adaptive neighborhood structure that swaps the selected time slot randomly with the timeslot that has the highest cost (penalty).

However, SA still could get stuck or trapped in local optimum, in the future work we suggest to hybridizing SA with other approaches or Artificial intelligence (AI) algorithms, in addition to test the solutions using a real world data set for further investigations.

References

[1] M. Affenzeller, A. Beham, M. Koffer, G. Kronberger, S.A. Wagner, S. Winkler, *Metaheuristic optimization*, In Hagenberg Research, Springer, 2009,

103-155.

[2] I. AlHadid, K. Kaabneh, H. Tarawneh, *Hybrid simulated annealing with meta-heuristic methods to solve uct problem*, Modern Applied Science, 12 2018.

[3] E. Aycan, T. Ayav, *Solving the course scheduling problem using simulated annealing*, In Advance Computing Conference, 2009, IACC 2009, IEEE International, 462-466.

[4] M. Birattari, L. Paquete, T. Stützle, K. Varrentrapp, *Classification of metaheuristics and design of experiments for the analysis of components*, Teknik Rapor, AIDA-01-05, 2001.

[5] C. Blum, A. Roli, *Metaheuristics in combinatorial optimization: Overview and conceptual comparison*, ACM computing surveys (CSUR), 35 (2003), 268-308.

[6] K.M. Cheh, J.B. Goldberg, R.G. Askin, *A note on the effect of neighborhood structure in simulated annealing*, Computers & Operations Research, 18 (1991), 537-547.

[7] R.W. Eglese, *Simulated annealing: a tool for operational research*, European Journal of Operational Research, 46 (1990), 271-281.

[8] M.S. Elmohamed, P. Coddington, G. Fox, *A comparison of annealing techniques for academic course scheduling*, In International Conference on the Practice and Theory of Automated Timetabling, 1997, 92-112.

[9] M. Fleischer, S.H. Jacobson, *Information theory and the finite-time behavior of the simulated annealing algorithm: Experimental results*, INFORMS Journal on Computing, 11 (1999), 35-43.

[10] L. Ingber, *High-resolution path-integral development of financial options*, Physica A: Statistical Mechanics and its Applications, 283 (2009), 529-558.

[11] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, *Optimization by simulated annealing*, Science, 220 (1983), 671-680.

[12] R. Lewis, B. Paechter, O. Rossi-Doria, *Metaheuristics for university course timetabling*, In Evolutionary scheduling, Springer, 2007, 237-272.

[13] Z. Lü, J.-K. Hao, *Adaptive tabu search for course timetabling*, European Journal of Operational Research, 200 (2010), 235-244.

[14] P. Moscato, *An introduction to population approaches for optimization and hierarchical objective functions: A discussion on the role of tabu search*, Annals of Operations Research, 41 (1993), 85-121.

[15] F. Ogbu, D.K. Smith, *The application of the simulated annealing algorithm to the solution of the n/m/cmax owshop problem*, Computers & Operations Research, 17 (1990), 243-253.

[16] E. Poupaert, Y. Deville, *Simulated annealing with estimated temperature*, AI Communications, 13 (2000), 19-26.

[17] F. Romeo, A. Sangiovanni-Vincentelli, *A theoretical framework for simulated annealing*, Algorithmica, 6 (1991), 302.

[18] A. Schaerf, *A survey of automated timetabling*, Artificial Intelligence Review, 13 (1999), 87-127.

[19] H. Tarawneh, M. Ayob, *Adaptive neighborhoods structure selection mechanism in simulated annealing for solving university course timetabling problems*, Journal of Applied Sciences, 13 (2013), 1087-1093.

[20] H. Tarawneh, M. Ayob, Z. Ahmad, *Simulated annealing with dynamic initial temperatures for university course timetable problem*, Journal of Engineering and Applied Sciences, 8 (2013), 58-68.

[21] H. Ueda, D. Ouchi, K. Takahashi, T. Miyahara, *Comparisons of genetic algorithms for timetabling problems*, Systems and Computers in Japan, 35 (2004), 1-12.

[22] J.M. Varanelli, J.P. Cohoon, *A fast method for generalized starting temperature determination in homogeneous two-stage simulated annealing systems*, Computers and Operations Research, 26 (1999), 481-504.

[23] A. Wren, *Scheduling, timetabling and rosteringa special relationship*, Practice and Theory of Automated Timetabling, 1996, 46-75.

[24] Z. Xinchao, *Simulated annealing algorithm with adaptive neighborhood*, Applied Soft Computing, 11 (2011), 1827-1836.

Accepted: 6.01.2019