A SA-ILS approach for the High School Timetabling Problem

Fonseca, G.H.G., Santos, H.G., Toffolo, T.A.M., Brito, S.S., Souza, M.J.F.

Received: date / Accepted: date

Abstract This work presents a heuristic approach proposed by one of the finalists of the Third International Timetabling Competition (ITC2011). The KHE school timetabling engine is used to generate an initial solution and then Simulated Annealing (SA) and Iterated Local Search (ILS) perform local search around this solution.

Keywords Simulated Annealing · Iterated Local Search · High School Timetabling Problem · Third International Timetabling Competition

1 Introduction

The diversity of School Timetabling models encountered around the world motivated the definition of an XML format to express different entities and constraints considered when building timetables[6]. The format evolved and a public repository[1] with a rich set of instances was built. To stimulated the research in this area, the Third International Timetabling Competition (ITC2011) occurred in 2012. This papers presents one of the finalists' solvers in this competition.

2 Solution Approach

Our approach uses the KHE school timetabling engine [2] to generate initial solutions and the metaheuristics Simulated Annealing and Iterated Local Search to perform local search around this solution. Since the constructive method is described in high level of details in[2], only the local search procedures will be detailed in the next sections.

Computing Department
Federal University of Ouro Preto
Ouro Preto, Minas Gerais, Brazil 35400-000

2.1 Local Search

Seven neighborhood structures were used in our local search approach:

- Event Swap (ES): two events e_1 and e_2 have their timeslots t_1 and t_2 swapped;
- Event Move (EM): an event e_1 is moved from timeslot t_1 to another timeslot t_2;
- Event Block Move (EBM): like ES, but when moving adjacent events with different duration keeps these events adjacent;
- Resource Swap (RS): two events e_1 and e_2 have their assigned resources r_1 and r_2 swapped, resources r_1 and r_2 should play the same role (e.g. both have to be teachers);
- Resource Move (RM): an event e_1 has his assigned resource r_1 replaced by a new resource r_2.
- Permute Resources (PR): given a resource r_1, up to n_{PR} events assigned to r_1 have their timeslots permutated; the events are chosen at random and the parameter n_{PR} is set to 7 so that all the permutations can be computed in a short amount of time.
- Kempe Move (KM): two times t_1 and t_2 are fixed and one seeks the best path at the bipartite conflict graph containing all events in t_1 and t_2; arcs are build from conflicting events which are in different timeslots and their cost is the cost of swapping the timeslots of these two events;

All local search methods can apply any move on the proposed neighborhoods (except PR, which is used only in the perturbation phase of ILS, returning the best different neighbor). If the instance requires assignment of resources (i.e. there exists at least one *AssignResourceConstraint* constraint), the kind of neighborhood is chosen based on the following probabilities: ES = 0.20, EM = 0.38, EBM = 0.10, RS = 0.20, RM = 0.10 and KM = 0.02, otherwise, the neighborhood RS and RM are not used and the odds are: ES = 0.40, EM = 0.38, EBS = 0.20 and KM = 0.02. These values were empirically adjusted.

2.1.1 Simulated Annealing Implementation

Proposed by [3], the metaheuristic Simulated Annealing is a probabilistic method based on an analogy to thermodynamics simulating the cooling of a set of heated atoms. This technique starts its search from any initial solution. The main procedure consists of a loop that randomly generates, at each iteration, one neighbor s' of the current solution s. Movements are probabilistically selected considering a temperature T and the cost variation of the movement Δ.

The developed implementation of Simulated Annealing is described in Algorithm 1. Parameters used were $\alpha = 0.97$, $T_0 = 1$, $SAmax = 10,000$ and $SAreheats = 5$. The method $selectNeighborhood()$ just chooses a neighborhood structure according to the neighborhood probabilities previously defined.

Algorithm 1: Developed implementation of Simulated Annealing

```
Input: f(.), N(.), α, SAmax, T_0, SAreheats, s, timeout
Output: Best solution s* found.
s* ← s; IterT ← 0; T ← T_0; reheats ← 0;
while reheats < SAreheats and elapsedTime < timeout do
    while IterT < SAmax do
        IterT ← IterT + 1;
        k ← selectNeighborhood();
        Generate a random neighbor s' ∈ N_k(s);
        Δ = f(s') − f(s));
        if Δ < 0 then
            s ← s';
            if f(s') < f(s*) then s* ← s';
        else
            Take x ∈ [0, 1];
            if x < e^{−Δ/T} then s ← s';
    T ← α × T;
    IterT ← 0;
    if T < 0.1 then
        reheats ← reheats + 1;
        T ← T_0
return s*;
```

2.1.2 Iterated Local Search

The method Iterated Local Search (ILS)[4] is based on the idea that a local search procedure can achieve better results by optimizing different solutions generated through disturbances on the local optimum solution.

Our ILS algorithm starts from an initial solution s_0 obtained by the Simulated Annealing procedure and makes disturbances of size p_{size} under s_0 followed by a descent method. A disturbance is the unconditional acceptance of a neighbor generated by neighborhoods PR or KM, both with 0.5 of probability.

The descent phase uses a Randomic Non Ascendent Method, which accepts only neighbors if they are better than or match the current solution. Tested moves are excluded from the neighborhood and return only when an improvement to the current best solution is reached. The local search phase ends when there is no remaining neighbor to be explored.

The local search produces a solution s' which will be accepted if it is better than the best solution s^* found. In such case, the disturbance size p_{size} gets back to the initial size p_0. If the iteration $Iter$ reaches a limit $Iter_{max}$, the disturbance size is incremented. Yet if the disturbance size reaches a bound p_{max}, it goes back to the initial size p_0. Algorithm 2 presents the developed implementation of ILS. The considered parameters are $ILS_{max} = 10,000$, $p_0 = 1$, $p_{max} = 10$ and $MaxIter_p = 10$.

Algorithm 2: Developed implementation of ILS

```
Input: f(.), N(.), ILS_max, p_0, p_max, MaxIter_p, s, timeout
Output: Best solution s* found.
s ← descentPhase(s); s* ← s;
p_size ← p_0; Iter ← 0;
for i ← 0 until ILS_max do
    if elapsedTime ≤ timeout then
        for j ← 0 until p_size do
            s ← s_p ∈ N(s);
        s' ← descentPhase(s);
        if f(s' < f(s*)) then
            s ← s'; s* ← s';
            Iter ← 0; p_size ← p_0;
        else
            s ← s*;
            Iter ← Iter + 1;
        if Iter = MaxIter then
            p_size ← p_size + p_0;
            if p_size ≥ p_max then  p_size ← p_0;
return s*;
```

3 Concluding Remarks

This paper presented a SA-ILS approach proposed by one the competition finalists of the Third ITC2011. Even though final results will only appear in the organizer's paper [5] we are confident that our method produced good results: we are finalists of the competition and our method improved several best known solutions for the competition instance set. Possible improvements are the development of an augmented set of (larger) neighborhoods and a proper experimental study to fine tune parameter selection. This almost surely will improve these results.

References

1. Benchmarking project for (high) school timetabling (2012). URL http://www.utwente.nl/ctit/hstt/
2. Kingston, J.H.: A software library for school timetabling (2012). Available at http://sydney.edu.au/engineering/it/~jeff/khe/, May 2012
3. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science **220**, 671–680 (1983)
4. Lourenco, H.R., Martin, O.C., Stutzle, T.: Iterated local search. In: F. Glover, G. Kochenberger (eds.) Handbook of Metaheuristics, chap. 11. Kluwer Academic Publishers, Boston (2003)
5. Post, G., Di Gaspero, L., Kingston, J.H., McCollum, B., Schaerf, A.: The third international timetabling competition. In: Proceedings of the 9th International Conference on the Practice and Theory of Automated Timetabling
6. Post, G., Kingston, J., Ahmadi, S., Daskalaki, S., Gogos, C., Kyngas, J., Nurmi, C., Musliu, N., Pillay, N., Santos, H., Schaerf, A.: XHSTT: an XML archive for high school timetabling problems in different countries. Annals of Operations Research pp. 1–7