Constructing initial neighborhoods to identify critical constraints

D. Moody, G. Kendall and A. Bar-Noy
City University of New York Graduate Center and School of Computer Science, University of Nottingham, UK

Abstract—**Recent course scheduling competitions have seen solution approaches which construct an initial solution quickly, and then employ a local search to improve the solution. With the use of different seeds, this process is repeated, searching for the best solution. Solutions with constraint violations provide little guidance on which constraints to relax in order to produce a better quality solution. Our approach seeks to construct several high quality initial solutions and analyze their characteristics which enables us to predict the relative success of the local search phase. With this capability, sets of initial solutions can be generated with selected constraint relaxations, leading to a prediction of which constraint relaxation can most improve the final solution, leading to a good quality solution.**

I. Introduction

The Metaheuristics Network sponsored an International Timetabling Competition in 2003 [9], involving a course scheduling problem. This competition was followed by a second competition, with different tiers focusing on variations of course scheduling. The objective is to assign courses to a day / time / room, avoiding "hard" conflicts and minimizing "soft" constraints An evaluation function is used to determine the value of the solution based upon the soft constraint violations.

The majority of approaches to these problems involve a two-phase approach. The first phase establishes an initial solution, which is usually free of hard constraints (though not an absolute requirement). The second, and more intensive phase, is a local search, where course assignments are swapped or moved to other time periods, reducing the soft constraint violations and improving the quality of the solution. Burke and Newall describe such an approach in [2]. Kostuch [6] reports the best results with a two phase approach.

The creation of the initial solution is typically highly stochastic and hence may or may not serve the local search phase well. Both phases are repeatedly performed with different seeds, to obtain the best overall solution. The initial solution represents little more than a starting point for the second phase.

Manuscript received February 29,2008

Douglas L. Moody (email: dmoody@citytech.cuny.edu) and Amotz Bar-Noy (email: amotz@sci.brooklyn.cuny.edu) are with the City University of New York Graduate Center, Department of Computer Science. Graham Kendall is with the School of Computer Science, University of Nottingham, UK (email:gxk@cs.nott.ac.uk).

Our approach has two main objectives. The first is to develop a method to predict whether an initial solution will serve as a good base for the local search phase. The second objective is to use this prediction capability to generate sets of initial solutions, where each set relaxes a different constraint of the problem. These sets of initial solutions, differing only by their constraint relaxation, can identify the constraints most affecting the solution quality. The scheduler, could then re-consider a course's constraints, or accept the existing soft constraint violations. McCollum [7] speaks of a similar goal stating the need for "*identification and comparison of key dataset characteristics and potential linkages with the likely best search approach to be taken*".

II. Problem Definition and Initial Solution Generation

In this paper we focus on the course timetabling problem of the Second International Timetabling Competition [8], held in 2007. This problem, within Tier 3 of the competition, consists of instances with courses (each having multiple sections), curricula associated with sets of courses, course availability within weekly time-periods, and room capacities. Each instance has the following constraints:

- Two sections cannot be scheduled in the same room during the same time period;
- Teachers cannot teach two sections during the same time period;
- Courses with the same curriculum can be scheduled during the same time period;
- All sections of a course cannot be scheduled during an unavailable time period;
- Sections should not violate room capacities;
- Sections of a course with the same curriculum should be in a time period next to another section in the curriculum;
- The course sections must run over a minimum number of days
- All course sections should share the same room.

The first four constraints are hard constraints, while the last four are soft constraints. Violations of the soft constraints increase the evaluation function.

Our approach considers each course as a tile, with each section being a block in the tile. A similar approach was used by Kingston in [4,5]. In Kingston's problem a "form" is the equivalent of a course, while a "meeting" is equivalent to a section. The sections are positioned in a greedy and constructive manner within the scheduling grid, according to that course's constraints. Bar-Noy and Moody showed tiling to be an effective approach to establishing a partial solution in [1]. Each section is placed in a "slot", which is a combination of a room/time period. Each tile placement minimizes the soft constraint violations of the course, given the previously scheduled tiles. During the tiling step, the room capacity constraint is treated as a hard constraint. As more courses are scheduled, remaining courses become increasingly more difficult to place, without violating hard constraints. The courses are ordered for assignment based upon their scheduling "difficulty", a function of the number of sections, curricula and unavailable time period constraints. Different orderings are generated through the introduction of a random factor, which can lower a course assignment in the order, delaying the tiling of the course. This stochastic factor enables various initial solutions to be constructed, though the tiling process is constructive.

At the tiling step conclusion, remaining unscheduled sections, due to the room capacity constraint, are scheduled. If unscheduled sections still exist, this course assignment order is not included in the set of initial solutions.

III. INITIAL SOLUTION ANALSYIS AND LOCAL SEARCH PHASE

Each initial solution is analyzed for its support of the local search phase. Several local search approaches are available, and we utilize a simple swapping method. The swapping method involves switching the slots of two sections, or moving a section to an unused slot. Our implementation calculates the impact on the evaluation function when swapping each section with every other section. The swap causing the largest reduction is actually performed, breaking ties randomly. The schedule is then reevaluated for positive swap candidates. This process repeats until no swaps can be made to improve the evaluation function. This hill-climbing method does not make a swap, unless there is an immediate benefit to the evaluation function.

Our objective is to analyze the quality of the initial solutions and predict whether they will lead to a high quality final solution. This approach embeds a common constraint processing technique of "look-ahead strategies" described by Dechter [3]. In this strategy, the remaining domains of unassigned variables are used for ordering purposes. We extend this concept to evaluate the overall value of the solution in terms of reaching a high quality final solution.

Our analysis of an initial solution looks at several characteristics of the solution:

- **Solution Value**: the evaluation function value of the initial solution before the local search phase
- **Total Swap Count**: The total number of possible swaps in the solution which have a positive effect on the evaluation function.
- **Swapability Value**: This section prorates the reduction in the evaluation function over all possible moves. For example, if a section can be moved to two time periods, each swap's value is multiplied by 0.5 and added to this value.
- **Movement Value**: This section counts the total number of slots a section can be assigned to, given the section currently in that slot is removed from the time period. A section's movement to another slot is often restricted by other sections within the slot's time-period having the same curricula or having the same instructor. If section A in time-period 1 is swapped with section B in time-period 2, we need to be sure section A does not violate a hard constraint with any other course (other than A) in time-period 2. We need not worry about hard constraint relationships between A and B, since they will be in different time-periods after the swap. This value indicates how tightly constrained the schedule is with respect to hard constraints.

We seek to create a prediction factor based upon a function of the initial solution characteristics discussed above. The following tables in figures 1 and 2 show the initial solution analysis for three instances from the competition. Each instance has had 20 initial solutions generated, differing only by the assignment order of the courses in the tiling step. For each initial neighborhood, the four characteristics above are presented, along with the final solution value after the local search phase.

Seed	Initial Solution Value	Move-ment Value	Swap-ability Factor	Total Swap Count	Final Solution Value
1	193	14596	1451	1542	16
2	225	14240	2173	1434	64
3	286	14591	1415	1868	74
4	276	14474	1665	1584	78
5	230	14224	1523	1590	22
6	315	14439	2119	1494	88
7	224	14089	2414	2014	25
8	230	14132	1582	1892	56
9	263	14371	2036	1558	28
10	331	14283	2557	1886	28
11	286	14258	2668	2228	52
12	253	14251	2148	1976	59
13	225	14259	1437	1842	24
14	270	14295	1951	1912	28
15	271	14305	2551	1650	45
16	221	14438	1861	1736	47
17	260	14316	3586	2144	32
18	227	14049	1390	1862	54
19	186	14576	1476	1288	23
20	304	14287	2320	1558	25

Fig. 1. Results From Instance 1

Seed	Initial Solution Value	Move-ment Value	Swap-ability Factor	Total Swap Count	Final Solution Value
1	514	112138	2289	23996	179
2	567	112705	3342	25782	173
3	596	113062	3443	26312	182
4	573	111655	2676	24900	189

5	528	112097	2709	23354	188
6	563	112405	1900	25818	185
7	562	112398	2422	25256	181
8	616	112111	3824	26426	197
9	572	112091	3589	25098	185
10	562	113192	2979	25778	195
11	601	112817	2623	27270	200
12	558	112666	2951	26218	184
13	564	112066	3366	25936	204
14	615	112523	3037	26264	199
15	566	112059	3512	25566	216
16	556	113078	2723	26060	205
17	520	111665	2365	24954	181
18	574	112342	3038	25164	186
19	564	112296	2449	25182	207
20	562	113190	2979	25578	165

Fig. 2. Results From Instance 8

IV. Results

Our results, shown in Figures 1 and 2, show a high correlation between the Initial Solution Value and the Final value. In the three instances shown, as well as for the other instances not reported here, we note that the best final solution value came from one of the top five initial solutions.

Other factors can be used to separate the set of high quality initial solutions. Consider seed 1 and 19 in instance 1, shown in Figure 1. Seed 19 had a better initial value, but less number of total swaps available. Comparing seed 19 to seed 20, the final solution results were nearly identical although there is a 70% difference in the initial solution value. In this instance, the swapability factor was 50% higher in seed 20, indicating potential in performing swaps to improve the evaluation function. The best final solution of instance 8 came from the use of seed 20. However that seed's initial value of 562 was 48 higher than seed number 1. This greater degree of improvement in the local search phase could possibly can be attributed to the higher movement and swap factors within seed 20 versus seed 1.

Our results demonstrate, for certain seeds, that we have an ability to predict, within a range of a probability distribution, the relative quality of the final solution. We intend to continue to analyze these results, so we can pick the initial solution for the local search phase, from a given set of initial solutions. Our work will produce a prediction factor, based upon these and perhaps other initial solution characteristics. Using the prediction factor, we will first show how our prediction factor can forecast the solution quality of our swapping technique. We will then investigate if our forecasting can predict the success of other local search methods.

V. Future Work

The prediction factor will enable our work to concentrate on generating sets of initial solutions, with each set differing only in a constraint relaxation. For example, the unavailability of a course can be relaxed, the minimum number of sections can be reduced, or the curricula modified for a course. These modifications could be done by a manual scheduler, to try to improve the schedule. Our approach will enable a scheduler to make these constraint relaxations, and quickly view the effect on the schedule's quality.

The Traveling Tournament Problem, introduced by Trick [10], provides another problem to evaluate our approach's ability to predict the success of the search phase. In this problem, a subset of all tiles are placed in the schedule, without violating hard constraints. Remaining tiles are broken into blocks, which are scheduled in a greedy fashion. The success of this second (greedy) phase may be predicted by analyzing factors similar to those discussed for the course scheduling problem.

We will use our predictability factor to continue to forecast the success of the local search phase. We will also validate our approach by submitting initial solutions to other local search phases. This step will verify that our predictability factor can also provide an indication of the solution quality for other local search approaches.

VI. Conclusions

Scheduling professionals in the field need high quality schedules and guidance on schedule constraint relaxation to achieve a useable schedule. Our approach uses a tiling method to quickly construct quality initial solutions, and assigning to each initial solution a factor predicting the quality of the final solution.

References

[1] Bar-Noy, A. and Moody, D. "A Tiling Approach for Fast Implementation of the Traveling Tournament Problem," Practice and Theory of Automated Timetabling (PATAT06, Brno, August 2006), Conference Proceedings, pp. 351-358.

[2] Burke, E.K. and Newall, J.P. "A Multi-Stage Evolutionary Algorithm for the Timetable Problem," the IEEE Transactions on Evolutionary Computation, Vol. 3.1. pp 63-74.

[3] Decter, Rina, Constraint Processing, First Edition. San Francisco: Morgan Kaufmann, 2003. pp. 117-144.

[4] Kingston, J. "Hierarchical Timetable Construction", Practice and Theory of Automated Timetabling (PATAT06: Brno, August 2006), Conference Proceedings, pp. 196-208.

[5] Kingston, J. "A Tiling Algorithm for High School Timetabling", Practice and Theory of Automated Timetabling (PATAT04: Pittsburgh, August 2004), Conference Proceedings, pp. 233-250.

[6] Kostuch, P. "The University Course Timetabling Problem with a 3-Phase Approach", Practice and and Theory of Automated Timetabling (PATAT04: Pittsburgh, August 2004), Conference Proceedings, pp. 251-266.

[7] McCollum, B. "University Timetabling: Bridging the Gap between Research and Practice", Practice and Theory of Automated Timetabling (PATAT06: Brno, August 2006), Conference Proceedings, pp. 15-33.

[8] "PATAT2008 International Timetabling Competition". . Online document at http://www.cs.qub.ac.uk/itc2007. Retrieved: November 29, 2007.

[9] Paechter, B. "International Timetabling Competition". Metaheuristics Network. 31 March 2003.

[10] Trick,M.A., Challenge Traveling tournament instances. Online document at http://mat.gsia.cmu.edu/TOURN/. Last update as of writing: January 18, 2006.