Hardness Prediction for the University Course Timetabling Problem

Philipp Kostuch[1] and Krzysztof Socha[2]

[1] Oxford University, Department of Statistics
1 South Parks Rd., Oxford, OX1 3TG, UK
kostuch@stats.ox.ac.uk
http://www.stats.ox.ac.uk

—

[2] IRIDIA, Université Libre de Bruxelles, CP 194/6,
Av. Franklin D. Roosevelt 50, 1050 Brussels, Belgium
ksocha@ulb.ac.be
http://iridia.ulb.ac.be

Abstract. This paper presents an attempt to find a statistical model that predicts the hardness of the University Course Timetabling Problem by analyzing instance properties. The model may later be used for better understanding what makes a particular instance hard. It may also be used for tuning the algorithm actually solving that problem instance. The paper introduces the definition of hardness, explains the statistical approach used for modeling instance hardness, as well as presents results obtained and possible ways of exploiting them.

1 Introduction

Metaheuristics are used nowadays for solving numerous types of optimization problems. However, they always have to be carefully tuned to the particular problem they are to solve. This tuning is often done based on the assumption that problem instances of similar size and similar structure pose similar difficulty to solve them by a given (meta)heuristic algorithm. Hence, the algorithms are run on a set of training instances that resemble the ones that the algorithm is expected to solve later, hoping that it will ensure optimal performance.

Unfortunately, often, especially in case of constrained combinatorial optimization problems, very similar problem instances in terms of size and structure, prove not to be similar in difficulty. One of the problems that exemplifies such behavior is the University Course Timetabling Problem (UCTP).

In this paper we try to define a measure of hardness (i.e. the difficulty of solving) for instances of UCTP. Based on this measure, we notice that very similar instances (i.e. ones created with the same parameters for the instance generator) may vary significantly in their difficulty to be solved by a (meta)heuristic algorithm.

Later, we try to show that in the case of UCTP, the hardness is an intrinsic characteristic of an instance with respect to a given (meta)heuristic algorithm. We show that based on the analysis of an instance and some statistics, it is possible to predict the hardness of an instance for a given algorithm with reasonable accuracy without actually attempting to solve it. This predicted value may be later used to understand better what makes one instance harder than another, and also to tune the parameters of the algorithm, and hence allow to obtain better results.

The remaining part of this paper is organized as follows. Section 2 describes the UCTP. Section 3 explains the underlying idea and motivation for modeling the instance hardness. Following that, Section 4 introduces the statistical model. In Section 5 the performance of the model is presented and discussed. Eventually, Section 6 discusses the results obtained and indicates possible uses of the hardness prediction model.

2 University Course Timetabling Problem

The problem used to illustrate the thesis of this paper, is the University Course Timetabling Problem (UCTP) [1]. It is a type of constraint satisfaction problem. It consists of a set of n events $E = \{e_1, \ldots, e_n\}$ to be scheduled in a set of 45 timeslots $T = \{t_1, \ldots, t_{45}\}$ (5 days in a week, 9 time-slots a day), and a set of j rooms $R = \{r_1, \ldots, r_j\}$ in which events can take place. Additionally, there is a set of students S who attend the events, and a set of features F satisfied by rooms and required by events. Each student attends a subset of events. A feasible timetable is one in which all events have been assigned a time-slot and a room, so that the following hard constraints are satisfied:

- no student attends more than one event at the same time;
- the room is big enough for all the attending students and posses all the features required by the event;
- only one event is taking place in each room at a given time.

In addition, a feasible candidate timetable is penalized equally for each occurrence of the following soft constraint violations:

- a student has a class in the last slot of the day;
- a student has more than two classes in a row (one penalty for each class above the first two);
- a student has exactly one class during a day.

The infeasible timetables are worthless and are considered equally bad regardless of the actual level of infeasibility. The objective is to minimize the number of soft constraint violations in a feasible timetable. The solution to the UCTP is a mapping of events into particular time-slots and rooms.

The instances we use in our research come from a generator written by Paechter[3].

[3] `http://www.dcs.napier.ac.uk/~benp`

3 Idea and Motivation

The initial motivation of the research presented in this paper comes from submissions to the International Timetabling Competition[4]. While creating metaheuristic algorithms for solving the UCTP instances used in the competition, it became obvious that although the instances are of similar size, the difficulty of solving them varies greatly.

Based on this observation, we developed the hypothesis that the difficulty of solving any given instance of UCTP is a function of some parameters of the instance that are intrinsic to it, and more complex than just its size [2]. If this hypothesis is true, it should be possible to define a *hardness* of an instance in terms of some of its characteristics. It should also be possible to predict values obtained by an algorithm just by looking at the instance.

We should make more precise here, what we mean exactly by the hardness of an instance. In principle, all the instances of UCTP under investigation are known to have at least one perfect solution. Hence, if we defined the hardness as the measure of how close to 0 any instance can be solved, the hardness of all those instances would be exactly the same. In our case however, the goal was to solve the given instances as well as possible, but within a given time limit. We will hence define the hardness of the instance as the quality of the solution obtained within this time limit by a given algorithm. Also, as the results obtained by a (meta)heuristic algorithm for a given problem instance tend to have some variance, we will consider in our investigation only the mean of the obtained values as the measure of hardness.

Following the hypothesis that the hardness depends on some intrinsic characteristic of an instance, we decide to measure several different features of the instances–the model's explanatory variables or *covariates*. If we find the right covariates, and if the hypothesis is true, the combination of covariates may be used to create a regression model for instance hardness–that is, find the functional relationship between the combination of covariates and the hardness of the instance.

Considering the problem investigated, we chose to measure some summary statistics of the instance to be used as covariates. Altogether there are 27 distinct covariates that we initially consider as important for the regression model. Sec. 4 describes in more detail how the final subset of those has been chosen for the final model.

For the purpose of choosing and fitting the model we used the results obtained by $\mathcal{MAX}$-$\mathcal{MIN}$ Ant System ($\mathcal{MM}$AS) [3] designed particularly for this problem [4]. $\mathcal{MM}$AS is a version of Ant Colony Optimization (ACO) metaheuristic initially proposed by Dorigo [5]. ACO has been in recent years widely used for solving combinatorial optimization problems [6] also including constrained satisfaction problems [7].

The final model chosen will obviously apply only to the results obtained by our $\mathcal{MM}$AS. Certainly, a comparison of models found for different algorithms

[4] `http://www.idsia.ch/Files/ttcomp2002/`

may provide some information whether the same features of the instance make it difficult for different algorithms. However interesting this topic might be, we do not focus on it in this paper.

4 Statistical Model

4.1 Normal linear models

The task formulated in the previous section is the archetypal statistical problem of regression. Among the large variety of models developed in the regression context, the best known are *linear models* [8]. The formulaic description for the i-th observation under a linear model is

$$y_i = \sum_{j=1}^{p} \beta_j x_{i,j} + \epsilon_i \tag{1}$$

where y_i is the observed response, $x_{i,j}$'s the covariate values for the i-th observation, β_j's the model parameters and ϵ_i the error of the i-th observation.

The term *linear* refers to the fact that the parameters appear only linearly in the *predictor* $\eta(x, \beta) = \sum_{j=1}^{p} \beta_j x_{i,j}$. This does however not mean that such a model can only cater for linear influences of covariates. It is for example possible to incorporate a covariate x in a linear and a quadratic term x^2 if this seems to be justified based on domain specific knowledge. Also possible are so-called interaction terms, i.e. multiplicative effects between covariates. All a linear model requires is for these terms to have a linear parameter. It is also common that in a linear model the predictor acts on the same scale as the observation, by which we mean that the function linking the predictor to the observation is the identity (as has already been assumed in Eq. 1).

To complete a regression model, a distribution for the error terms has to be specified. In *normal linear models*–also just called *normal models*–the ϵ_i are assumed to be independent, identically distributed (i.i.d.) according to $N(0, \sigma^2)$.

4.2 Casting our problem into a normal model framework

To apply the normal model approach to a given problem, one has to ensure that the underlying modeling assumptions are met. We will do so in this section for the prediction of hardness based on instance characteristics.

The first observation for our data is that we are confronted with a more complex problem than in Eq. 1. Namely, we can separate the error into two terms. One being the standard modeling error associated with a given realization of the covariates. The other being a measurement error of the response as different metaheuristic runs will–on the same instance–yield different results. This can be formulated as

$$y_{i,k} = \sum_{j=1}^{p} \beta_j x_{i,j} + \epsilon_i + \widetilde{\epsilon}_{i,k} \tag{2}$$

where $y_{i,k}$ is the k-th measurement on the i-th instance, $x_{i,j}$'s the covariate values for the i-th observation, β_j's the model parameters, ϵ_i the error of the i-th instance and $\widetilde{\epsilon}_{i,k}$ the measurement error for the k-th measurement on the i-th instance.

This kind of model is known as a linear mixed effects model in the literature [9]. Instead of following this approach, we circumvent the problem of two different error terms by averaging out the measurement error $\widetilde{\epsilon}_{i,k}$. For this, we have 15 runs on every instance and use the mean value as response. This allows us to stay within the normal model framework. It means a loss of efficiency in the modeling process as we throw away information that could otherwise be utilized, but eliminates the measurement error that in the algorithm under study is fairly large (see next paragraph). So, the model for the mean values has smaller variation in the error term which opens up the chance of better predictions.

To assess whether averaging over 15 instances is enough to neglect the measurement error in the further treatment of the data we have 50 runs on a particular instance and calculate the means for 25 random samples of size 15. The mean for all 50 values was 150, while the upper and lower quantiles of the means based on 15 runs were 145 and 155 respectively with 3 outliers being further away. To put the mean values in relation to the individual measurements and to give an idea of the measurement error itself, we note the lower and upper quantiles for the 50 runs to be 135 and 160.

So, while there is still some variation, this spread is small enough to continue with our approach, noting that the remaining measurement error will be a lower bound on the achievable prediction quality. This particular instance was chosen as the algorithm could only achieve relatively high values on it and a consistent observation over all instances was that the variance of the measurement error increases with the response value.

This heteroscedasticy of the measurement errors is the inspiration for the second concern we have fitting the data with a normal model: the distribution assumption for the remaining error term. As mentioned in Sec. 4.1 the ϵ_i's are assumed to be i.i.d. $\sim N(0, \sigma^2)$. Two things can go wrong with this. The errors may not be normally distributed or they may not have constant variance. In particular, the error variance may be a function of the response value as was observed for the measurement error, where it grew with the response. Unfortunately, with the ϵ_i's we are not in a position to test this *a priori*. It will only be possible to assess the appropriateness of the distribution assumption *a posteriori* via diagnostic tools on the residuals of the fitted model. We will do this in Sec. 5.

We mention two techniques that deal with such deviations from the normal model. The first is to achieve constant variance by transforming the response to a different scale, e.g. using the log function on it before fitting the model [10]. As we will implement this, we shall explain the rational behind such a transformation. If we assume the error not to be an additive term, but rather a fraction of the response, taking the log will transform a multiplicative error into an additive one that fits into the normal model framework. If the problem

lies deeper, i.e. a non-normal error distribution, Generalized Linear models (for example with a Poisson distribution) have to be used [11].

4.3 Fitting the Model

This section deals with the non-mathematical side of model fitting. By mathematical model fitting we mean the process of determining the parameter values β when all other properties of the model are fully specified. The mathematics behind this process are simple as for normal models minimizing the log-likelihood (which is the fitting approach for more general models) coincides with minimizing the sum of squares (SSQ) of the errors.

What we concern ourselves with, is the process of selecting a subset of the available covariates to form the final model [12]. The need for covariate selection arises from the purpose for which we model: prediction. We have a training set of 120 new instances generated from the range of input variables to the instance generator that were used for the competition instances. If we fit the model with all covariates, we get a lower bound on how good we can fit this set, but the prediction quality will be poor as we have over-fitted our data, see Table 1. Therefore, the following procedure has been implemented in S-Plus for both the model with unchanged response and the one with the log-response.

Simulated Annealing [13] is used to determine a good set of covariates. This is done as the search space–especially when interactions are included–gets too large for enumeration. Standard covariate selection methods such as forward and backward selection are greedy methods that can get stuck in bad local optima. As search space we have all possible subsets of the full covariate set. We choose a random starting location, and the neighborhood is defined by either randomly dropping or adding a covariate. Because of the expensive fitness evaluation, we can only run a moderate number of iterations.

For the fitness evaluation of a solution, 6-fold cross-validation is used on the training set, i.e. it is randomly divided in 6 sets of 20 instances, and then 5 sets are used to fit the current model and the remaining set is used to assess the prediction power by recording the error SSQ for this 6^{th} set. This is repeated for leaving out all sets in turn. The fitness for the current solution is the average over the 6 values. One modification is put in place when the predicted value for an instance is negative: negative predictions are set to 0 before we calculate the error term. This is reasonable as a negative value in the prediction can be interpreted as an instance that is solved to optimality in less time than allowed.

The Simulated Annealing engine is run 25 times and for every run we record the covariates included in the final model. Then we drop those covariates that were included only 3 or less times in these final models. On the remaining set of covariates we restart the whole procedure leading after another 25 runs to a second set of covariates that are to be dropped for good. In both models this turns out to be sufficient to end up with a stable set of covariates, as Simulated Annealing on the remaining covariates did not drop anymore.

5 Results

With the above approach we arrive at two final models: one for the unchanged response and the other for the log response. Two reasons let us focus on the log model. Firstly, analytical plots for the residuals indicate that the log model meets the normality assumption better. Secondly, we observe that the fitting criterion is different for the two models, with the log model pursuing the–in our opinion–more reasonable goal.

We explain this difference briefly. Both models were fitted such that they minimize a fitness function based on the error SSQ, i.e. on the gap in absolute values between prediction and observation. But since the modeled values are on different scales, this leads to different fitting goals on the original response scale. The model with unchanged response controls the spread around the observation in terms of absolute value. In contrast, the error for the log response enters the model on the original scale as $prediction \times e^{\epsilon_i}$, where for small errors $e^{\epsilon_i} \approx 1 + \epsilon_i$ holds. So, the log response model minimizes approximately the squared sum of percentage errors.

Another observation of interest in this context is that using squared error values leads to a worse average error in favor of limiting extreme errors. In the case of a normal distribution, which we assume for our errors, the so calculated *standard deviation* σ has a ready interpretation as signifying a confidence level of approximately 70%.

Table 1 summarizes the results of the models fitted on the log response. The first two are the models with all covariates included (Full) and the model reduced by Simulated Annealing (Final). The last model (Interaction) will be introduced later. In the final model we dropped 1 more term after the automated selection, as it was statistically insignificant. As a test set, we used the 20 competition instances.

We see that the full model fits the training set better but its prediction power is worse. For both models the prediction error is larger than the error on the training set, and clearly larger than the predicted error $\widehat{\sigma}$ based on the cross-validation. We note that this prediction is biased and will always under-estimate the real error. The average percentage errors have been included as they are more intuitively understandable, but we point out that optimization has been carried out with respect to σ-values.

We will now discuss the final model and its remaining covariates in more detail. Table 2 gives the 8 covariates for this model, the covariate parameters and their values on the the response scale, i.e. e^{β_i}. The covariate values–except for slack–have been normalized, i.e. were transformed to have 0 mean and variance 1. This allows a direct comparison of the effect size independent of the scaling of the covariates. Slack was not subjected to this procedure as it is a discrete covariate with only two levels.

Before we can interpret the covariates we have to introduce a graph closely related to the UCTP. Events are the nodes in this graph and they are joined by an edge if the events cannot be placed in the same time-slot. There can be two reason for an edge: a student who attends both events or the two events

Table 1. Summary of the errors in the 3 models fitted on log(response). Values are the %-errors between prediction and observation on the original scale.

		Training set			Test set	
Model	parameters	σ	Av. Error	$\widehat{\sigma}$	σ	Av. Error
Full	27	18.0	13.2	25.7	28.4	21.8
Final	9	19.4	13.8	21.0	27.2	19.8
Interaction	18	17.2	12.4	21.0	22.0	16.6

Table 2. Final set of 8 covariates used by the regression model ordered by their level of importance.

Name	Coefficient	Effect on Response scale
av. weighted event degree	1.39	4.02
av. event size	-1.15	0.32
slack	-1.22	0.29
av. weighted room options	-0.63	0.53
sd. events per room	0.62	1.85
av. event degree1	0.51	1.66
sd. event degree2	0.27	1.31
no of 1-option events	0.20	1.22

need both the same room. The original graph (referred to as 1) comprises only student conflicts while the second graph (referred to as 2) includes room conflicts as well.

Four effects in the model are related to the node degree in this graph. The weighted event degree (the weights are the number of students that cause a given edge) and the ordinary event degree in graph 1 increase the prediction as they grow. The effect of the weighted event degree seems to be larger but in fact it is almost perfectly correlated with the average event size. This means that in a normal instance these two effects will have to be subtracted giving a much smaller effect for the average weighted event degree. Dropping either one of the correlated terms and hoping that the other would absorb the effect of both was considered but led to a worse fit. The last effect referring to this graph is the standard deviation in event degree in graph 2. The problem becomes harder as the spread increases.

Three effects are related to the room constraints. Most prominently, the average weighted number of room options per event (the square root was used for weighting to put emphasis on small numbers) is negatively correlated with the hardness. Positively correlated is the number of events that have only one room options. We recorded them as they form a special subset of events, causing additional edges between graph 1 and graph 2. The third covariate measures the standard deviation in the number of possible candidate events from the room's point of view. The more spread out this distribution is the harder the problem.

The last effect in the model-and the most decisive-is the factor variable slack, which indicates whether 40 time-slots are a lower bound for the placement of the events, i.e. whether the number of rooms times 40 equals the number of events. If slack is present, we can expect the problem to be substantially easier. For this statement-as for all others made in this section-we assume a *ceteris paribus* situation, i.e. that we can change 1 covariate at a time keeping the others fixed. Obviously, this assumption is hard to meet in a real problem and therefore this effect analysis has to be seen with this limitation in mind.

After isolating these 8 covariates, we included them and all 2nd order interactions in another normal model. This model was then pruned according to a standard statistical criterion (AIC stepwise selection, taken from the MASS library [10], which is a greedy search method that trades off the log likelihood of a model against the number of covariates used). We note that this criterion optimizes prediction power only indirectly. The results for this model can be seen in Table 1. It achieves a better fit on the training set than the full model and a better prediction than the final model. The gap between the predicted error and the observed error has markedly decreased. This clearly indicates that further research should be directed towards more complicated models.

6 Conclusions

The paper presents a successful attempt to predict the quality of the solutions found by the $\mathcal{MAX}$-$\mathcal{MIN}$ Ant System for the University Course Timetabling Problem. We have attempted to predict the hardness of a complex constrained optimization problem for a metaheuristic algorithm. Without making any explicit assumptions about the inner operation of the algorithm (i.e. we treated it as a black box algorithm), we have managed to develop a reasonably simple regression model allowing to predict the competition results with an average error of less than 17%.

The statistical model we developed may be further exploited in at least two distinctive ways. One option is to use the knowledge acquired in order to modify the algorithm and improve its performance. This may include exploiting the understanding of important covariates, or exploiting the estimated hardness. The second option is to use the model for sensitivity analysis of the instance. When considering the UCTP as a real world problem, it may be possible to change some characteristics. Based on the model, the changes with the best marginal benefit can be determined (e.g. should one more room be made available for the courses or is it better to fit the existing rooms with additional features).

6.1 Future Work

In the future we plan to investigate alternative model choices that we left for now unexplored. This includes linear models with a more systematic interaction investigation, generalized linear models and non-linear models.

Independently, we would like to investigate in more detail, how the predicted hardness may be used for improving the performance of the algorithm. Also, we would like to develop similar models for other algorithms solving the UCTP. This way we could see if it is possible to maintain this level of prediction accuracy for the other algorithms, and also if the covariates that are important for one algorithm are similarly important for the others. This research will show whether the concept of hardness can be to some extend generalized, i.e. can be made less algorithm-dependent.

Acknowledgments. Philipp Kostuch acknowledges the support of a Marie Curie Training Site fellowship funded by the Improving Human Potential (IHP) programme of the Commission of the European Community (CEC), grant HPRN-CT-2000-00032. The research work was also supported by the "Metaheuristics Network", a Marie Curie Research Training Network funded by the Improving Human Potential programme of the CEC, grant HPRN-CT-1999-00106, and by the "ANTS" project, an "Action de Recherche Concertée" funded by Scientific Research Directorate of the French Community of Belgium.

References

1. Rossi-Doria, O., Sampels, M., Chiarandini, M., Knowles, J., Manfrin, M., Mastrolilli, M., Paquete, L., Paechter, B.: A comparison of the performance of different metaheuristics on the timetabling problem. In Burke, E.K., De Causmaecker, P., eds.: Proceedings of PATAT 2002. Volume 2740. (2003)
2. Kostuch, P.: University Course Timetabling, Transfer Thesis, Oxford University, England (2003)
3. Stützle, T., Hoos, H.H.: $\mathcal{MAX}$-$\mathcal{MIN}$ Ant System. Future Generation Computer Systems **16** (2000) 889–914
4. Socha, K.: $\mathcal{MAX}$-$\mathcal{MIN}$ Ant System for International Timetabling Competition. Technical Report TR/IRIDIA/2003-30, Université Libre de Bruxelles, Belgium (2003)
5. Dorigo, M., Di Caro, G.: The Ant Colony Optimization meta-heuristic. In Corne, D., Dorigo, M., Glover, F., eds.: New Ideas in Optimization, McGraw-Hill (1999)
6. Maniezzo, V., Carbonaro, A.: Ant Colony Optimization: an Overview. In Ribeiro, C., ed.: Essays and Surveys in Metaheuristics, Kluwer Academic Publishers (2001)
7. Roli, A., Blum, C., Dorigo, M.: ACO for maximal constraint satisfaction problems. In: Proceedings of the 4th Metaheuristics International Conference (MIC 2001). Volume 1. (2001) 187–191
8. Neter, J., Wasserman, W., Kutner, M.: Applied Linear Statistical Models. 3 edn. Irwin (1990)
9. Davidian, M., Giltinan, D.: Non-linear Models for Repeated Measurement Data. Chapman & Hall (1995)
10. Venables, W., Ripley, B.: Modern Applied Statists with S-PLUS. 3 edn. Springer (1999)
11. McCullagh, P., Nelder, J.: Generalized Linear Models. 2 edn. Chapman & Hall (1989)
12. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer (2001)
13. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by Simulated Annealing. Science **220** (1983) 671–680