Choquet Integral for Combining Heuristic Values for Exam Timetabling Problem

Tiago Cardal Pais · Edmund Burke

Abstract In this paper we present a constructive heuristic approach based on Choquet integral. We use this method to combine the information given by different basic heuristics. We use a fuzzy measure to model the importance of each heuristic as well as the interaction between them. We test our approach on 2 different testbeds and compare its performance against the individual heuristics. Moreover, we also compare the results against the best results reported in the literature.

Keywords Exam Timetabling, Fuzzy Measure, Choquet Integral, Construction Heuristics

1 Introduction

Problems related to timetabling are present in daily life. Solving timetabling problems is a crucial task and affects many institutions and services like hospitals, transportation enterprizes, educational establishments, among many others. These problems have been the object of increasing interest by the research community. Many interesting proposals have been presented, particulary in the field of Operations Research and Artificial Intelligence, to solve timetabling problems in sports (Easton et al. 2004; Trick 2001), transportations (bus,railways,planes) (Isaai and Singh 2001; Caprara et al. 2001; Qi et al. 2004), schools (Abramson et al. 1999; Colorni et al. 1998; Ribeiro Filho and Lorena 2001; Hansen and Vidal 1995; Schaerf 1999) and universities (Awad and Chinneck 1998; Burke et al. 2006; Burke and Newall 2003; Caramia et al. 2001; Casey and Thompson 2003; Carter et al. 1994, 1996; Corr et al. 2006; Dowsland and Thompson 2004; Erben 2001; Di Gaspero 2002; Di Gaspero and Schaerf 2001; Kendall and Mohd Hussin 2004; Merlot et al. 2003; Paquete and Fonseca 2001; Petrovic and Bykov 2002; Schimmelpfeng and Helber 2007; Thompson and Dowsland 1996, 1998; White and Xie 2001; Yang and Petrovic 2004).

A general definition of timetabling was given by Burke, Kingston, and de Werra (2004):

Tiago Cardal Pais · Edmund Burke
ASAP group, University Of Nottingham, Nottingham, NG8 1BB, UK
E-mail: {tpp,ekb}@cs.nott.ac.uk

"A timetabling problem is a problem with four parameters, T a finite set of times, R a finite set of resources, M, a finite set of meetings: and C, a finite set of constraints. The problem is to assign times and recourses to the meetings so as to satisfy the constraints as far as possible."
Hence, if we consider exams as meetings then we are facing exactly the problem that we want to tackle in this paper, that is, the exam timetabling problem.
In Burke and Newall (2004) an iterated construction algorithm is described. They make use of a construction ordering heuristic as the basic method for scheduling the exams. However, the authors introduce an iterated adaptive method which consists of changing the "degree" of each exam in each iteration. They proposed an incremental and exponential adaptation scheme. In the first case the "degree" is modified by one unit at each iteration. On the other hand, the exponential scheme increments the "degree" by 2^n, where n is the number of iterations by which a particular "degree" was modified. Moreover, they compare the performance of the algorithm when using different basic ordering heuristics. They use the largest degree first (LD), a flat ordering (which initializes every "degree" on 0), smallest degree first (SmD), saturation degree (SD) and random ordering. They tested the algorithm using the Toronto's data set (Carter et al. 1996). It can be observed that the adaptation mechanism helps to improve the initial timetable given by the original order of the exams.
In Asmuni et al. (2009) a fuzzy multiple heuristic ordering approach is presented. In this work a simple heuristic ordering was implemented, based on the Carter and Laporte (1996) algorithm. The following three different criteria were used to order exams: (1) largest degree (LD); (2) largest enrolment (LE); (3) and least saturation (SD) degree criterion. They use a fuzzy inference system to combine the different criteria previously mentioned. All possible combinations were tested ($LD + LE$, $SD + LE$ and $LD + SD$). The Mandani type fuzzy system that they used has a 9 rule structure, meaning that two linguistic variables were used to evaluate the exam "quality" and for each variable three linguistic terms were defined: "small", "medium" or "high". The output linguistic variable "examweight" is also defined by the same three linguistic terms. A pre-normalisation of data was also performed before computing the fuzzy inference system. In this process a linear transformation was used. Furthermore, a tuning process was also implemented. It consists, basically, of changing simultaneously, by small steps, the upper bound, centre and lower bound of the three membership functions. All results obtained for each instance used the best "tuned system". All approaches were tested using the Toronto's data set (Carter et al. 1996). They conclude that the approach using a tuned fuzzy system with the SD and LE as input variables gave, overall, the best results.
Qu et al. (2009a) presented an adaptive hybridisation of basic graph heuristics within a graph hyper-heuristic framework. They first started studying some statistical properties of a random constructive graph hyper-heuristic. They observed that sequences of SD heuristic hybridised with Largest Weighted Degree (LWD) gave better results than if hybridised with LD or LE heuristic. Following that, they proposed an adaptive approach which uses sequences of SD heuristic hybridized with LWD. This new method consists of two steps. Firstly, they iteratively hybridise the LWD heuristic into the first half of a sequence based on SD heuristic. At the end of each iteration, if the solution obtained was feasible, the hybridisation amount was increased by 0.03. On the other

hand, if that solution was feasible but had a higher cost than the best solution previously found, the hybridisation amount was decreased by 0.01. Secondly, they hybridised the entire sequence with LWD. However, in this step only the best sequences obtained from the previous step were used.
The paper is organized as follows. Section 2 presents the mathematical formulation of the exam timetabling problem that we adopted in this work. It is followed, in section 3, by a brief description of the key concepts for a better understanding of the proposed method. Section 4 contains all the details about the construction algorithm and how to combine all the basic heuristic values using Choquet integrals. Afterwards, a description of the experimental design is given in Section 5, as well as the experimental results and discussion. Finally, the conclusions are drawn in Section 6.

2 Exam Timetabling Problem Definition

The exam timetabling problem can be formulated as a combinatorial optimisation problem. In order to compare our approach with other methods proposed in the literature, we adopt the following formulation followed by many authors. Let,

$$E = \text{total number of exams,} \tag{1}$$

$$P = \text{total number of periods,} \tag{2}$$

$$S = \text{total number of students,} \tag{3}$$

$$c_{ij} = \text{number of students enroled in exam } i \text{ and } j \text{ for } i,j = 1,\ldots,E, \tag{4}$$

$$a_{ij} = \begin{cases} 1 & \text{if } c_{ij} > 0 \\ 0 & \text{otherwise} \end{cases} \text{ for } i,j = 1,\ldots,E, \tag{5}$$

and consider the following variables:

$$x_i = \text{ period in which exam } i \text{ is scheduled } , i = 1,\ldots,E \tag{6}$$

The formulation is:

$$\min f = \frac{\sum_{i=1}^{E}\sum_{j=1}^{E} proximity\_cost(x_i,x_j) * c_{ij}}{2S} \tag{7}$$

$$\text{subject to: } \quad |x_i - x_j| \geq a_{ij} \text{ for } i,j = 1,\ldots,E \text{ and } i \neq j \tag{8}$$

$$1 \leq x_i \leq P \text{ and integer } \text{ for } i,j = 1,\ldots,E \tag{9}$$

where proximity cost is defined as:

$$proximity\_cost(x_i,x_j) = \begin{cases} 2^{5-|x_i-x_j|} & \text{if } 0 < |x_i - x_j| < 6 \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

for $i,j = 1,\ldots,E$ and $i \neq j$

The objective function (7) penalizes the proximity of exams with students in common, using as weights the number of students involved in both examinations and a factor that depends on the proximity of the periods, ranging from 16 to 0 (Carter et al. 1996). The constraint 8 ensures that any two exams indexed by i and j with students in common are not assigned to the same period.

3 Essential Concepts

Let us briefly present some useful concepts for our work before we describe the construction algorithm in more detail.

3.1 Fuzzy Sets

Definition 1 (Zimmermann (1996))**.** *If $\mathcal{X}$ is a collection of objects designated by x then a fuzzy set $\widetilde{A}$ in X is defined by a set of pairs:*

$$\widetilde{A} = \{(x, \mu(x)) | x \in X\}$$

where $\mu_{\widetilde{A}}(x)$ is the membership function of x em $\widetilde{A}$.

From now on we are going to refer to a membership function as $f_{\widetilde{A}}$ instead of the traditional way, as presented in the previous definition, $\mu_{\widetilde{A}}$ to avoid any confusion with the representation of a fuzzy measure μ (see Section 3.3).
For example we can consider the age of a person. Let X be the age domain and x the age of a certain person. Then the fuzzy set YOUNG may be defined by:

$$\widetilde{A} = \{(x, f(x)) | x \in X\}$$

where

$$f_{\widetilde{A}}(x) = \begin{cases} 0 \text{ , if } & x \geq 65 \\ \frac{65-x}{30} \text{ , if } & 35 \leq x \leq 65 \\ 1 \text{ , if } & x \leq 35 \end{cases}$$

Fuzzy sets are often represented by triangular, trapezoidal (triangular as a particular case) or gaussian membership functions. The membership function for YOUNG presented above is an example of a trapezoidal function. Generalising, a trapezoidal function is given by the following membership function:

$$f_{\widetilde{A}} : D \subset X \to [0, 1]$$

where

$$f_{\widetilde{A}}(x) = \begin{cases} 0 & \text{, if } x \leq a \vee x > d \\ \frac{x-a}{b-a} & \text{, if } a < x \leq b \\ 1 & \text{, if } b < x \leq c \\ \frac{d-x}{d-c} & \text{, if } c < x \leq d \end{cases} \tag{11}$$

where $x \in D \subset \mathcal{X}$ and $\widetilde{A}$ is a fuzzy set in $\mathcal{X}$.
The triangular function is a trapezoidal function where $b = c$.

3.2 Linguistic Variable

The linguistic values or terms of a linguistic variables are concepts defined by words or expressions of a natural language.

Definition 2 (Zadeh (1975))**.** *A linguistic variable is characterized by the quintuple* $(\mathcal{H}, T(\mathcal{H}), U, G, M)$ *, where* $\mathcal{H}$ *is the name of the variable,* $T(\mathcal{H})$ *is the set of terms or linguistic values of* $\mathcal{H}$*,* U *is the universe of the variable,* G *the semantic rule that generates the terms in* $T(\mathcal{H})$ *and* M *is the semantic rule associating to each term or linguist value its meaning trough the fuzzy set* $M(X)$ *(*$M(X)$ *is a fuzzy set on* U*).*

Let us consider the linguistic variable TEMPERATURE as in (Klir and Yuan 1995). We can have a pure numerical interpretation for this concept as depicted in case (b) in Figure 1, but we can represent it as a linguistic variable (case (a)), characterised by the linguist values { Very Low, Low, Average, High, Very High}.

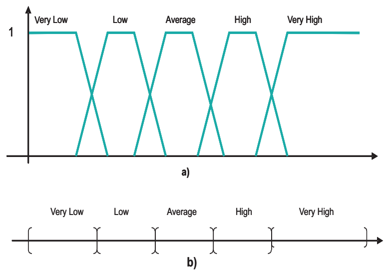


Fig. 1: a) Temperature as Linguistic Variable - b) Numerical representation of Temperature

3.3 Fuzzy Measures

Since here we are working with finite spaces, we are going to present a simplified definition of a fuzzy measure. More details about this topic can be found in Wang and Klir (1992).

Definition 3 (Grabisch (1995))**.** *A fuzzy measure* μ *defined on the measurable space* $(X, \mathfrak{X})$ *is a set function* $\mu : \mathfrak{X} \to [0, 1]$ *satisfying the following axioms:*

i $\mu(\emptyset) = 0$*,* $\mu(X) = 1$*. This is the usual convention, although in general* $\mu(X)$ *can be any positive finite (or infinity) quantity.*

ii $A \subseteq B \Rightarrow \mu(A) \leq \mu(B)$ *(monotonicity).*

where $X \triangleq \{x_1, \ldots, x_n\}$*, and generally* $\mathfrak{X}$ *is a* σ*-algebra on a space* X*.*
$(X, \mathfrak{X}, \mu)$ *is said to be a fuzzy measure space.*

In this work, we assume that the σ-algebra $\mathfrak{X}$ is the power set of X. Hence, in this case we have $X = \{SD, CD, LD, LWD, LE\}$ and $\mathfrak{X} = \mathcal{P}(X)$.

3.4 Choquet Integral

Again, as in Section 3.3, we present a definition of Choquet integral for the particular case of discrete spaces.

Definition 4. *(Grabisch (1995)): Let $(X, \mathfrak{X}, \mu)$ be a fuzzy measure space. The Choquet integral of a function $f : X \to [0, 1]$ with respect to μ is defined by*

$$\mathfrak{C}_\mu(f(x_1), ..., f(x_n)) \triangleq \sum_{i=1}^{n} (f(x_{(i)}) - f(x_{(i-1)}))\mu(A_{(i)})$$

where $\cdot_{(i)}$ indicates a permutation such that $0 \leq f(x_{(1)}) \leq \ldots \leq f(x_{(n)}) \leq 1$, $A_{(i)} \triangleq \{x_{(i)}, \ldots, x_{(n)}\}$, and $f(x_{(0)}) = 0$.

To better illustrate how Choquet integral works we present the same example described in Murofushi and Sugeno (1989) paper. Consider that there is a rare book collection which consists of two volumes (let us label y_1 as volume 1, and y_2 as volume 2). There is a bookseller who is interested in buying this rare collection. Therefore, he offers: $\mu(\{y_1\})$ monetary units (m.u.) per volume 1; $\mu(\{y_2\})$ m.u. per volume 2; and $\mu(\{y_1, y_2\})$ m.u. for each entire collection. Obviously, he sets a higher values for the complete set, i.e. $\mu(\{y_1, y_2\}) \geq \mu(\{y_1\}) + \mu(\{y_2\})$. Suppose now, that there is a person that sells $x_1 = h(y_1)$ units of volume 1 and $x_2 = h(y_2)$ units of volume 2, where $x_1 \leq x_2$. We can say that he offers x_1 complete collections and $(x_2 - x_1)$ volumes 2. Therefore, he would get $x_1 \times \mu(\{y_1, y_2\}) + (x_2 - x_1) \times \mu(\{y_2\})$ m.u.

4 Construction Heuristic

We implemented a simple construction algorithm. The construction heuristic block is composed by two secondary order heuristics. These are: exam ordering heuristic; and period ordering heuristic. The former one concerns the order in which exams are scheduled. As the latter chooses in which period a particular exam should be scheduled. Furthermore, we also implemented a backtracking procedure which is based on the one described in Carter et al. (1996). A pseudocode of the algorithm is depicted in Algorithm 1.

Algorithm 1 Construction algorithm pseudocode

```
unscheduled_exams ← all exams
while unscheduled_exams is not empty do
  Select an exam from the list unscheduled_exams using a exam_ordering_heuristic
  Select a feasible period to schedule the previous selected exam using a
  period_ordering_heuristic
  if it is impossible to select a feasible period then
    Use the backtracking procedure described in Carter et al. (1996)
    if it is impossible to backtrack then
      break
    end if
  end if
  Update unscheduled_exams list
end while
```

The four period ordering heuristics implemented in this work are the following:

1. random: a random feasible period is chosen using an uniform distribution;
2. first period: the first feasible period is chosen;
3. biased: a roulette-wheel scheme is used to choose the period. The weights are computed using Eq. 7;
4. deterministic and random: the periods are sorted according to the weights obtained by using Eq. 7. Ties are broken randomly.

We implemented three different types of exam ordering. The logic behind them is similar to the one used for period ordering. The description of the heuristics is given below.

1. random: a random exam is chosen from the unscheduled ones using a uniform distribution;
2. biased: a roulette-wheel scheme is used to choose one of the unscheduled exams using an empirical distribution based on one of six heuristics described below;
3. deterministic and random: the exams are sorted according to one of six heuristics described below. Ties are broken randomly.

The heuristics implemented represent how “hard” it is to schedule a particular exam. Each one of the five basic heuristics are described below.

1. Saturation Degree (SD): increasingly order exams by the number of feasible periods in which an exam can be scheduled;
2. Colour Degree (CD): decreasingly order exams by the number of total conflicts that an exam has with the already scheduled exams;
3. Largest Degree (LD): decreasingly order exams by the number of total conflicts;
4. Largest Weighted Degree (LWD): decreasingly order exams by the number of total conflicts weighted by the number of students involved in each one;
5. Largest Enrolment (LE): decreasingly order exams by the number of enrolments;

4.1 Construction Heuristic using Choquet Integral

The more traditional aggregation methods (e.g. Weighted Sum, OWA) are easy to interpret but are too restrictive since they are not able to represent the interaction between the criteria. It is assumed that the criteria is independent, when in most practical cases that does not happen. The motivation to use this method is so that we can represent the information given by the individual heuristics (given above), as well as the interaction between those heuristics by using a fuzzy measure (see Section 3.3) in an understandable way. This way we can model criteria that are not independent.
Choquet integral can be regarded as an extension of Lebesgue integral. That is, if the measure at hand is additive, or in other words, it is a classical measure, then Choquet integral coincides with Lebesgue integral (Murofushi and Sugeno 1989). Therefore, it is easier to interpret its output than other fuzzy integrals. Moreover, Choquet integral has some algebraic properties that other fuzzy integrals do not have (Grabisch 1995). Hence, it makes it more suitable for multicriteria decision making problems (Murofushi and Sugeno 1989; Grabisch 1995, 1996).

Since the information given by the heuristics had different units, we used fuzzy sets (see Section 3.1) to normalise its values into the unit interval. The goal of using this information is to decide which exam is "harder" to schedule. Bearing that in mind, we modeled each basic heuristic as a linguistic variable (see Section 3.2). Hence, we used a triangular membership function (see Eq. 11) to represent the linguistic terms, such as "low SD" value, "high CD" value, "high LD" value, "high LWD" value and "high LE" value. The membership functions are relative to each iteration, e.g., if the highest SD value is 5 (in one particular iteration) all exams with that value are going to have a membership of 0. The membership functions for each linguistic term are defined below.

$$f_{\widetilde{lowSD}}(x_{SD}) = \frac{maxSD - x_{SD}}{maxSD} \tag{12}$$

$$f_{\widetilde{highCD}}(x_{CD}) = \frac{x_{CD}}{maxCD} \tag{13}$$

$$f_{\widetilde{highLD}}(x_{LD}) = \frac{x_{LD}}{maxLD} \tag{14}$$

$$f_{\widetilde{highLWD}}(x_{LWD}) = \frac{x_{LWD}}{maxLWD} \tag{15}$$

$$f_{\widetilde{highLE}}(x_{LE}) = \frac{x_{LE}}{maxLE} \tag{16}$$

Where $maxSD$, $maxCD$, $maxLD$, $maxLWD$ and $maxLE$ is the maximum SD, CD, LD, LWD, LE value in the current iteration, respectively.
Moreover, if some of the $maxCD$, $maxLD$, $maxLWD$, $maxLE$ values are equal to zero, in some iteration, the respective function returns 0 by default. On the other hand, if $maxSD$ value is equal to zero the function returns 1 by default. The weights presented in Table 1 and 2 were chosen using a "rule of thumb". We set the individual weights according to how each individual heuristic performed . The interaction's weights were defined by analysing the information given by the heuristics, i.e., if the heuristics present some kind of complementary information, the weight given to that interaction should be at least higher than the sum of the individual weights. For example, the SD and CD heuristic values capture different information of the timetable being constructed. One could expect that the interaction between these two criteria to be synergetic, i.e. it provides a better understanding of the problem than both heuristics separately. Hence, the weight given to the interaction between these two is higher than the sum of both together. On the other hand, if heuristics present similar information a lower weight should be given. For instance, the LD and LWD heuristic values share, to some degree, the same information; hence the weight given to the interaction between these two is less than the sum of both together. For both individual and interaction weights, we used a trial-and-error approach based on how well the heuristic performed on the hec-s-92 data set.

Table 1: Individual and two-way interaction weights for μ fuzzy measure

Weight	Individual Criterion	Weight	Criteria
0	empty	0.51	SD,LD
0.5	SD	0.515	SD,LWD
0.01	LD	0.52	SD,LE
0.015	LWD	0.8	SD,CD
0.02	LE	0.02	LD,LWD
0.2	CD	0.04	LD,LE
		0.21	LD,CD
		0.045	LWD,LE
		0.3	LWD,CD
		0.32	LE,CD

Table 2: Interaction's weights for μ fuzzy measure

Weight	Criteria	Weight	Criteria
0.6	SD,LD,LE	0.7	SD,LD,LWD,LE
0.62	SD,LWD,LE	0.9	SD,LD,LWD,CD
0.85	SD,LD,CD	0.98	SD,LD,LE,CD
0.88	SD,LWD,CD	0.95	SD,LWD,LE,CD
0.9	SD,LE,CD	0.6	LD,LWD,LE,CD
0.35	LD,LWD,CD	1	SD,LD,LWD,LE,CD
0.06	LD,LWD,LE		
0.4	LD,LE,CD		
0.43	LWD,LE,CD		
0.55	SD,LD,LWD		

With all the values fuzzyfied and a fuzzy measure set we can use the Choquet Integral (see Section 3.4) to combined all the information. As we did with the other basic heuristics, the exams are ordered decreasingly according to the values obtained by using this method. That is, if one exam is attributed value 1 it means that it is very hard to schedule, according to the information given by the five basic heuristics. Hence it should be scheduled before all other exams.
To better illustrate how the process works, an example is here presented. Consider two exams,e_1 and e_2. After the basic heuristic values were computed and normalized, we obtained the following values: $x_{e_1} = (0.4, 0.5, 0.5, 0.6, 0.8)$ and $x_{e_2} = (0.8, 0.4, 0.7, 0.7, 0.4)$, corresponding to the SD, CD, LD, LWD, LE heuristic, respectively. The Choquet integral value for exam e_1 is computed as:

$$\begin{aligned}
\mathfrak{C}_\mu(x_{e_1}) =& \mu_{\{SD,CD,LD,LWD,LE\}} x_{e_1SD} + \\
&+ \mu_{\{CD,LD,LWD,LE\}}(x_{e_1CD} - x_{e_1SD}) + \\
&+ \mu_{\{LD,LWD,LE\}}(x_{e_1LD} - x_{e_1CD}) + \\
&+ \mu_{\{LWD,LE\}}(x_{e_1LWD} - x_{e_1LD}) + \\
&+ \mu_{\{LE\}}(x_{e_1LE} - x_{e_1LWD}) = \\
=& 1 \times 0.4 + 0.6 \times (0.5 - 0.4) + 0.35 \times (0.5 - 0.5) + \\
&+ 0.045 \times (0.6 - 0.5) + 0.02 \times (0.8 - 0.6) = \\
=& 0.4685
\end{aligned}$$

An analogous process is also applied to exam e_2, giving the value $\mathfrak{C}_\mu(x_{e_2}) = 0.6150$. Hence, the next exam to be scheduled would be the exam e_2.

5 Experimental Design

To test the performance of the algorithm we used two data sets. The first one was a collection of real problems and is available in an online repository created by Michael Carter[1]. The second data set was put online for the International Timetabling Competition[2] (ITC 2007) (McCollum et al. 2007). In this work, we used the average performance to test whether the algorithm performs significatively better when distinct heuristics are used for ordering exams. For this purpose, we performed 330 runs for each heuristic and for each data file. In total, we ran the algorithm 41580 times ($330 runs \times 6 heuristics \times 21 datafiles$). For each data file and each heuristic, we computed an average of 33 runs. This way, we obtained 10 samples of the average performance of the algorithm for each data file and each heuristic. We assumed that the 10 samples were normally distributed since we were dealing with averages of 33 random independent and identically distributed variables. Hence, to compare the performance of different heuristics we used the statistical t-test with a significance level of 0.95. The following null hypothesis was used:

$$H0 : \mu_{dfi1_{hj1}} \geq \mu_{dfi2_{hj2}} \tag{17}$$

where μ represents the mean value and $dfi1, dfi2, hj1, hj2$ are the data files $i1, i2 \in${Carter's data files, ITC's data files} and the heuristics $j1, j2 \in${SD, CD, LD, LWD, LE, CI}, respectively.
In the next section we only present the results regarding the determinist order of exams and periods. The results of the other ordering strategies were considerably worse when compared to the aforementioned ones.

5.1 Computational Results

Tables 3 and 4 depict the minimum (min), maximum (max) and average (avg) values over the 330 runs for all data files (lines) and for each heuristic (columns). The best results are presented in boldface.
We also computed how many time each heuristic was statistically better than other k heuristics (with $k = 0, 1, 2, 3, 4, 5$) across all 21 data files. The results are depicted in Figure 2. For instance, we can observe that SD heuristic (see Figure 2a) performed better than 1 heuristic in 2 data files and was also better than all other heuristics in the other 2 data files.

5.2 Discussion

From analysing Figure 2 it can be seen that amongst the basic heuristics, the one that performs better is the SD heuristic followed by the CD heuristic.

[1] ftp://ftp.mie.utoronto.ca/pub/carter/testprob
[2] http://www.cs.qub.ac.uk/itc2007/

Table 3: Computational results of the basic and Choquet heuristics for the Carter's data set

Data Set		SD	CD	LD	LWD	LE	CI
car-f-92	min	4.56	4.74	5.15	5.17	5.03	**4.44**
	max	**6.17**	7.51	7.61	7.44	7.54	6.99
	avg	**5.07**	5.77	6.07	6.06	6.12	5.28
car-s-91	min	5.25	5.39	5.54	5.69	5.83	**5.18**
	max	7.15	7.76	8.77	8.33	8.34	**6.95**
	avg	5.73	6.25	6.70	6.65	6.82	**5.66**
ear-f-83	min	40.74	40.22	41.02	43.65	43.67	**39.55**
	max	58.78	61.46	58.73	60.06	61.34	**57.53**
	avg	46.56	49.00	49.13	51.03	51.04	**45.43**
hec-s-92	min	12.59	12.68	14.16	14.14	12.73	**12.20**
	max	**22.80**	26.61	27.14	23.15	23.40	25.27
	avg	**16.07**	17.22	18.41	17.53	18.11	16.48
kfu-s-93	min	15.92	16.12	16.41	16.05	16.70	**15.46**
	max	26.00	29.44	26.12	26.81	26.98	**22.98**
	avg	18.55	19.23	20.57	20.00	20.44	**17.54**
lse-f-91	min	11.96	12.03	12.95	12.28	12.48	**11.83**
	max	19.25	19.43	19.25	20.01	19.08	**15.55**
	avg	14.25	14.81	14.24	14.96	15.30	**12.89**
pur-s-93	min	4.95	4.96	5.04	5.05	5.15	**4.93**
	max	5.85	6.01	6.05	7.42	7.82	**5.81**
	avg	5.33	5.38	5.37	5.58	5.64	**5.19**
rye-s-93	min	10.48	10.65	12.35	10.33	10.92	**10.04**
	max	17.84	19.15	19.02	18.49	19.17	**16.07**
	avg	12.78	13.35	15.07	13.76	14.14	**11.85**
sta-f-83	min	**159.35**	159.71	162.11	163.09	161.71	160.50
	max	185.63	**182.97**	199.70	192.51	202.30	184.89
	avg	169.94	169.95	180.09	173.50	174.71	**169.46**
tre-s-92	min	8.90	9.04	10.18	9.25	9.47	**8.71**
	max	12.40	13.12	13.34	13.73	12.67	**11.07**
	avg	10.10	10.76	11.66	11.20	11.27	**9.27**
uta-s-92	min	3.64	3.67	4.04	3.76	3.86	**3.49**
	max	**4.69**	5.49	6.09	6.80	6.70	5.17
	avg	3.95	4.22	4.83	4.74	4.84	**3.80**
ute-s-92	min	28.93	**28.65**	31.34	29.55	29.45	29.44
	max	43.78	45.25	45.21	46.60	49.48	**39.15**
	avg	34.91	34.09	36.82	37.12	36.55	**33.44**
yor-f-83	min	43.29	43.07	45.27	46.38	45.74	**42.19**
	max	56.59	59.20	58.13	56.27	58.62	**54.77**
	avg	49.10	50.64	51.65	51.07	51.46	**47.94**

This can be somewhat explained since these two heuristics have a dynamic behaviour, while the other three are static, as mentioned in Section 4.

As it can be observed in Table 3 and 4, the CI heuristic almost always obtained the best results across all data sets. It obtained the best minimum, maximum and average results in 15, 17 and 18 out of 21 data files, respectively. Figure 2f shows that the CI heuristic was significatively better than all of other heuristics in 81% of the instances. Moreover, this heuristic was better than at least 4 heuristic in 90% of the cases.

Only two of the constructive methods (Burke and Newall 2004; Qu et al. 2009a) seem to perform better than the rest. However, as was described in Section 1, all methods (with the exception of Carter and Laporte (1996)) have incorporated a

Table 4: Computational results of the basic and Choquet heuristics for the International Timetabling Competition (ITC) data set

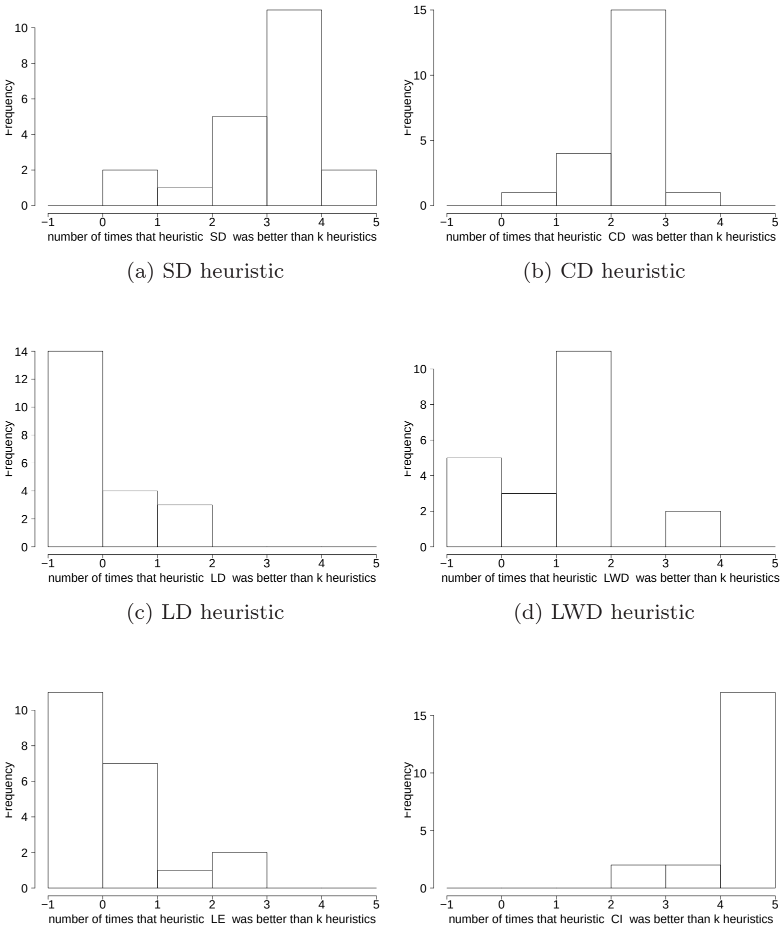
Data Set		SD	CD	LD	LWD	LE	CI
ITC1	min	1.20	1.20	1.30	1.22	1.22	**1.12**
	max	1.51	1.49	1.55	1.42	1.44	**1.30**
	avg	1.36	1.36	1.42	1.31	1.33	**1.21**
ITC2	min	0.28	0.28	0.30	0.28	0.28	**0.26**
	max	0.40	0.38	0.40	0.39	0.39	**0.35**
	avg	0.34	0.33	0.35	0.33	0.34	**0.30**
ITC3	min	1.91	1.90	1.99	1.88	1.93	**1.81**
	max	2.29	2.31	2.29	2.25	2.23	**2.06**
	avg	2.10	2.09	2.10	2.07	2.07	**1.93**
ITC4	min	13.99	13.66	14.49	15.77	15.91	**13.54**
	max	**22.82**	24.63	25.28	28.10	29.59	28.49
	avg	16.91	17.48	18.46	20.95	20.90	**16.67**
ITC5	min	0.49	0.50	0.59	0.53	0.55	**0.44**
	max	0.72	0.73	0.82	0.69	0.74	**0.61**
	avg	0.60	0.60	0.71	0.60	0.63	**0.52**
ITC6	min	4.63	4.78	4.87	**4.43**	4.67	4.50
	max	6.09	6.29	6.88	7.65	7.41	**5.49**
	avg	5.27	5.34	5.63	5.47	5.77	**4.90**
ITC7	min	**0.10**	0.11	0.13	0.14	0.14	0.11
	max	0.17	0.17	0.19	0.19	0.19	**0.16**
	avg	0.14	0.14	0.16	0.17	0.16	**0.13**
ITC8	min	**0.18**	0.18	0.23	0.22	0.24	0.18
	max	0.32	0.33	0.33	0.33	0.35	**0.30**
	avg	**0.25**	0.25	0.28	0.28	0.30	0.25

more sophisticated method to improve the construction process. Therefore, the heuristic described in this work held a much greater potential since the fuzzy measure used in this work was not a subject of any kind of sophisticated tuning procedure. Nevertheless, if we compare the results obtained by the CI heuristic with some other constructive methods described in the literature (which are depicted in table 5) we can observe that it presents very competitive results in most of the instances of the Carter's data set. Moreover, the CI heuristic is faster (see Table 6) than most of the construction heuristics depicted in Table 5. This makes it suitable to be used as a generator for population based algorithms since it builds good quality timetables.

6 Conclusions

In this work we presented a construction algorithm which uses a fuzzy measure and Choquet integral to combine the information given by 5 basic heuristics (see Section 4). The exams are then decreasingly ordered according to the value obtained by the Choquet integral and scheduled into a time period. This is chosen in virtue of minimising the total cost of the timetable (which is given by Equation 7).

The new method proposed in this work performs better than all basic heuristics in most of the test instances. However, in some of them the SD heuristic obtained better results. Nevertheless, we expect to enhance the performance of



(a) SD heuristic

(b) CD heuristic

(c) LD heuristic

(d) LWD heuristic

(e) LE heuristic

(f) CI heuristic

Fig. 2: Histogram of the number of times that each heuristic performed better than k others

CI heuristic by using other techniques (e.g. differential evolution (Price et al. 2005)) to tune the weights of the fuzzy measure. Moreover, instead of tuning the weights for each instance we can use a training data set. Through this, we expect the heuristic to perform well across a different range of instances with different characteristics.

Acknowledgements This research has been supported by Fundação para a Ciência e Tecnologia (FCT) Portugal grant number SFRH/BD/43486/2008.

Table 5: Best computational results of some constructive methods and the Choquet heuristic for the Carter's data set

Data Set	(Carter and Laporte 1996)	(Burke and Newall 2004)	(Asmuni et al. 2009)	(Qu et al. 2009a)	Best reported (Qu et al. 2009b)	CI
car-f-92	6.2	4.32	4.54	4.32	**3.93**	4.44
car-s-91	7.1	4.97	5.29	5.11	**4.5**	5.18
ear-f-83	36.4	36.16	37.02	35.56	**29.3**	39.55
hec-s-92	10.8	11.61	11.78	11.62	**9.2**	12.20
kfu-s-93	14.0	15.05	15.80	15.18	**13.0**	15.46
lse-f-91	10.5	10.96	12.09	11.32	**9.6**	11.83
pur-s-93	**3.9**	-	-	-	-	4.93
rye-s-93	7.3	-	10.38	-	**6.8**	10.04
sta-f-83	161.5	161.91	160.42	158.88	**134.9**	160.50
tre-s-92	9.6	8.38	8.67	8.52	**7.9**	8.71
uta-s-92	3.5	3.36	3.57	3.21	**3.14**	3.49
ute-s-92	25.8	27.41	28.07	28.00	**24.4**	29.44
yor-f-83	41.7	40.77	39.80	40.71	**36.2**	42.19

Table 6: Computational times (in seconds) for the *CI* heuristic. The values are an average of 300 runs.

Data Set	car-f-92	car-s-91	ear-f-83	hec-s-92	kfu-s-93	lse-f-91	pur-s-93
Times	5.03	9.13	0.34	0.15	2.89	1.66	382.81
Data Set	rye-s-93	sta-f-83	tre-s-92	uta-s-92	ute-s-92	yor-f-83	-
Times	3.42	0.11	0.60	6.97	0.24	0.43	-

References

D. Abramson, M. Krishnamoorthy, and H. Dang. Simulated annealing cooling schedules for the school timetabling problem. *Asia Pacific Journal of Operational Research*, 16:1–22, 1999.

H. Asmuni, E.K. Burke, J.M. Garibaldi, B. McCollum, and A.J. Parkes. An investigation of fuzzy multiple heuristic orderings in the construction of university examination timetables. *Computers and Operations Research*, 36(4):981–1001, 2009.

R. Awad and J. Chinneck. Proctor Assignment at Carleton University. *Interfaces*, 28 (2):58–71, 1998.

E. K Burke and J. P Newall. Enhancing Timetable Solutions with Local Search Methods. *Lectue Notes in Computer Science*, pages 195–206, 2003.

E. K. Burke and J. P. Newall. Solving examination timetabling problems through adaption of heuristic orderings. *Annals of operations Research*, 129(1):107–134, 2004.

EK Burke, J. Kingston, and D. de Werra. Applications to timetabling. *Handbook of Graph Theory*, pages 445–474, 2004.

E.K. Burke, A.Eckersley, B.McCollum, S.Petrovic, and R.Qu. Hybrid variable neighbourhood approaches to university exam timetabling. Technical Report NOTTCS-TR-2006-2, School of Computer Science and IT, University of Nottingham, 2006.

A. Caprara, M. Fischetti, PL Guida, M. Monaci, G. Sacco, and P. Toth. Solution of real-world train timetabling problems. In *System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on*, page 10, 2001.

M. Caramia, P. DellÓlmo, and G.F. Italiano. New algorithms for examination timetabling. In *Algorithm Engineering*, volume 1982 of *Lecture Notes in Computer Science*, pages 230–242. Springer, 2001.

M. W. Carter, G. Laporte, and J.W. Chinneck. A general examination scheduling system. *Interfaces*, 24(3):109–120, 1994.

Michael W. Carter and Gilbert Laporte. Recent developments in practical examination timetabling. In *Selected papers from the First International Conference on Practice and Theory of Automated Timetabling*, pages 3–21, London, UK, 1996. Springer-Verlag. ISBN 3-540-61794-9.

M.W. Carter, G. Laporte, and S.Y. Lee. Examination Timetabling: Algorithmic Strategies and Applications. *Journal of the Operational Research Society*, 47(3): 373–383, 1996.

S. Casey and J. Thompson. Grasping the examination scheduling problem. In *Practice and Theory of AutomatedTimetabling IV*, volume 2740/2003 of *Lecture Notes in Computer Science*, pages 232–244. Springer Berlin / Heidelberg, 2003.

A. Colorni, M. Dorigo, and V. Maniezzo. Metaheuristics for High School Timetabling. *Computational Optimization and Applications*, 9(3):275–298, 1998.

PH Corr, B. McCollum, MAJ McGreevy, and P. McMullan. A New Neural Network Based Construction Heuristic for the Examination Timetabling Problem. *Lecture Notes in Computer Science*, 4193:392, 2006.

L. Di Gaspero. Recolour, shake and kick: A recipe for the examination timetabling problem. In *Proceedings of the fourth international conference on the practice and theory of automated timetabling*, pages 404–407, Gent, Belgium, 2002.

Luca Di Gaspero and Andrea Schaerf. Tabu search techniques for examination timetabling. In *PATAT '00: Selected papers from the Third International Conference on Practice and Theory of Automated Timetabling III*, pages 104–117, London, UK, 2001. Springer-Verlag. ISBN 3-540-42421-0.

KA Dowsland and JM Thompson. Ant colony optimization for the examination scheduling problem. *Journal of the Operational Research Society*, 56(4):426–438, 2004.

K. Easton, G. Nemhauser, and M. Trick. Sports scheduling. *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*, 2004.

W. Erben. A Grouping Genetic Algorithm for Graph Colouring and Exam Timetabling. *Lecture Notes in Computer Science*, pages 132–158, 2001.

M. Grabisch. Fuzzy integral in multicriteria decision making. *Fuzzy Sets and Systems*, 69(3):279–298, 1995.

M. Grabisch. The application of fuzzy integrals in multicriteria decision making. *European journal of operational research*, 89(3):445–456, 1996.

M.P. Hansen and R.V.V. Vidal. Planning of high school examinations in Denmark. *European Journal of Operational Research*, 87(3):519–534, 1995.

MT Isaai and MG Singh. Hybrid applications of constraint satisfaction and metaheuristicsto railway timetabling: a comparative study. *Systems, Man and Cybernetics, Part C, IEEE Transactions on*, 31(1):87–95, 2001.

Graham Kendall and Naimah Mohd Hussin. An investigation of a tabu search based hyper-heuristic for examination timetabling. In Graham Kendall, Edmund K. Burke, and Sanja Petrovic, editors, *Selected papers from MISTA 2003*. Kluwer Publication, 2004. URL `http://www.asap.cs.nott.ac.uk/publications/pdf/nbh_mista03_extend.pdf`.

G. J. Klir and B. Yuan. *Fuzzy Sets and Fuzzy Logic*. Prentice Hall, 1995.

Barry McCollum, Paul McMullam, Edumund K. Burke, Andrew J. Parkes, and Rong Qu. The second international timetabling competition: Examination timetabling track. Technical Report QUB/IEEE/Tech/ITC2007/Exam/v4.0/17, Queen's University Belfast, September 20 2007.

L.T.G. Merlot, N. Boland, B.D. Hughes, and P.J. Stuckey. A hybrid algorithm for the examination timetabling problem. In *Practice and Theory of Automated-Timetabling IV*, volume 2740/2003 of *Lecture Notes in Computer Science*, pages 207–231. Springer Berlin / Heidelberg, 2003.

T. Murofushi and M. Sugeno. An interpretation of fuzzy measures and the Choquet integral as an integral with respect to a fuzzy measure. *Fuzzy sets and Systems*, 29(2):201–227, 1989.

Luis F. Paquete and Carlos M. Fonseca. A sutdy of examination timetabling with multiobjective evolutionary algorithms. In *Proccedings of 4th Metaheuristics International Conference*, 2001.

Sanja Petrovic and Yuri Bykov. A multiobjective optimisation technique for exam timetabling based on trajectories. In *Proceedings of the 4th International Conference on Practice and Theory of Automated Timetabling (PATAT 2002)*, volume 2740 of *Lecture Notes in Computer Science*, pages 179–192, Gent, Belgium, Aug 21-23 2002. Springer. URL `http://www.asap.cs.nott.ac.uk/publications/pdf/yuri_Patat02sv.pdf`.

Kenneth Price, Rainer M. Storn, and Jouni A. Lampinen. *Differential Evolution: A Practical Approach to Global Optimization (Natural Computing Series)*. Springer-Verlag New York, Inc., 2005. ISBN 3540209506.

X. Qi, J. Yang, and G. Yu. Scheduling problems in the airline industry. *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*, 2004.

R. Qu, E.K. Burke, and B. McCollum. Adaptive automated construction of hybrid heuristics for exam timetabling and graph colouring problems. *European Journal of Operational Research*, 198(2):392–404, 2009a.

R. Qu, EK Burke, B. Mccollum, LT Merlot, and SY Lee. A survey of search methodologies and automated system development for examination timetabling. *Journal of Scheduling*, 12(1):89, 2009b.

G. Ribeiro Filho and L.A.N. Lorena. A Constructive Evolutionary Approach to School Timetabling. *Applications of Evolutionary Computing - Lecture Notes in Computer Science*, 2037:130–139, 2001.

A. Schaerf. Local search techniques for large high school timetabling problems. *Systems, Man and Cybernetics, Part A, IEEE Transactions on*, 29(4):368–377, 1999.

K. Schimmelpfeng and S. Helber. Application of a real-world university-course timetabling model solved by integer programming. *OR Spectrum*, 29(4):783–803, 2007.

J.M. Thompson and K.A. Dowsland. Variants of simulated annealing for the examination timetabling problem. *Annals of Operations research*, 63(1):105–128, 1996.

Jonathan M. Thompson and Kathryn A. Dowsland. A robust simulated annealing based examination timetabling system. *Comput. Oper. Res.*, 25(7-8):637–648, 1998. ISSN 0305-0548. doi: http://dx.doi.org/10.1016/S0305-0548(97)00101-9.

M.A. Trick. A Schedule-Then-Break Approach to Sports Timetabling. *Lecture Notes in Computer Science*, pages 242–253, 2001.

Zhenyuan Wang and George J. Klir. *Fuzzy measure theory*. Plenum Publishing Corporation, 1992.

George M. White and Bill S. Xie. Examination timetables and tabu search with longer-term memory. In *PATAT '00: Selected papers from the Third International Conference on Practice and Theory of Automated Timetabling III*, pages 85–103, London, UK, 2001. Springer-Verlag. ISBN 3-540-42421-0.

Yong Yang and Sanja Petrovic. A novel similarity measure for heuristic selection in examination timetabling. In *Proccedings of the 5th International Conference on the Practice and Theory of Automated Timetabling (PATAT 2004)*, Pittsburg, August 18-20 2004. URL `http://www.asap.cs.nott.ac.uk/publications/pdf/yxy_Patat04.pdf`.

L.A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning, Part I. *Information Sciences*, 8(3):199–249, 1975.

H. J. Zimmermann. *Fuzzy Set Theory and its applications*. Kluwer Academic Publishers, 1996.