Fuzzy Adaptive Parameter Control of a Late Acceptance Hyper-heuristic

Warren G. Jackson, Ender Özcan, and Robert I. John
ASAP Research Group
School of Computer Science
University of Nottingham
Jubilee Campus
Wollaton Road
Nottingham, NG8 1BB, UK
Email: {psxwgj,ender.ozcan,robert.john}@nottingham.ac.uk

Abstract—A traditional iterative selection hyper-heuristic which manages a set of low level heuristics relies on two core components, a method for selecting a heuristic to apply at a given point, and a method to decide whether or not to accept the result of the heuristic application. In this paper, we present an initial study of a fuzzy system to control the list-size parameter of late-acceptance move acceptance method as a selection hyper-heuristic component. The performance of the fuzzy controlled selection hyper-heuristic is compared to its fixed parameter version and the best hyper-heuristic from a competition on the MAX-SAT problem domain. The results illustrate that a fuzzy control system can potentially be effective within a hyper-heuristic improving its performance.

I. INTRODUCTION

Hyper-heuristics are emerging high level methodologies that manage a set of low level heuristics during the search process for solving hard computational problems [1]. Özcan et al. [2] decomposed single-point search selection hyper-heuristics into two key components; a selection mechanism and a move acceptance criteria. Hyper-heuristics of this nature will be denoted as selection method-acceptance criteria in this paper herein. In such a framework, selection hyper-heuristics have an iterative cycle between heuristic selection and move acceptance. Operating on a single solution, a low-level heuristic is selected and applied at each point before a decision is made whether to accept or reject the candidate solution created by the application of the low-level heuristic. This process is repeated until some termination criteria is met.

The HyFlex [3] framework was initially developed in Java for the first Cross-domain Heuristic Search Challenge (CheSC 2011) [4] and is a software framework “designed to enable the development, testing and comparison of iterative general-purpose heuristic search algorithms (such as hyper-heuristics)” [9]. This framework provides six pre-implemented problem domains allowing researchers to concentrate on the development and application of high-level search methodologies for cross-domain search rather than on the implementation details of various problem domains and low-level heuristics.

Hyper-heuristics often employ meta-heuristics as their move acceptance criteria however one problem faced when using meta-heuristics are their uncertain parameter settings. For any given problem domain and problem instance, the best settings of such parameters is unknown. Within evolutionary algorithms, which are synonymous with meta-heuristics and hyper-heuristics, it has been shown that the optimal settings for their parameters change over time given the current stage of the EA [5] and therefore parameter control of the meta-heuristic’s parameters within the hyper-heuristic’s acceptance criteria is needed to achieve better performance.

Fuzzy logic [6] has been widely used in control applications and more recently to control parameters of meta-heuristics used for solving a range of NP-Hard problems including mathematical function optimisation [7], [8], [9], travelling salesman problem [10], the assignment problem [11], and the clustering problem [12]. All of these systems utilise information from the current state of the search, along with the current value of the parameter being controlled as inputs to the fuzzy system to decide on the parameter setting for the next iteration or stage of the search process. In other words, all of the fuzzy systems perform adaptive parameter control on the meta-heuristic parameters.

Late acceptance [13], [14] is a recently proposed meta-heuristic method which is similar to hill-climbing local-search in that the new (candidate) solution is compared with a previous solution. Late acceptance differs in that rather than comparing the candidate solution to the immediate previous solution, late acceptance compares the new solution with the solution visited L steps previously. Late acceptance has been used with hyper-heuristics and shown improvement on other meta-heuristic methods in [15], [16], [17], [18], [19] to solve a variety of combinatorial optimisation problems, however, all of these studies fixed the value of L for the execution of the hyper-heuristic.

In this study, a fuzzy system is developed using the Juzzy Framework [20] to control the list length parameter of late acceptance [13], [17] as the move acceptance component of a selection hyper-heuristic to demonstrate the application
of fuzzy control and its potential effectiveness in improving a hyper-heuristic’s high-level performance by improving its performance at a general level across all instances of a given problem domain. This hyper-heuristic is then tested against a fixed parameter version of the same hyper-heuristic at a value known to have good performance by previous empirical analysis and was applied to all instances of the MAX-SAT problem domain [21] available in the HyFlex Framework.

The rest of this paper is organised as follows. In Section II, a description of a late acceptance hyper-heuristic and its variant embedding a fuzzy system are provided. The empirical results discussing late acceptance list extension strategies and the performance of the fuzzy controlled late acceptance hyper-heuristic compared to its fixed length counterpart is presented in Section III-B and Section III-C respectively. Concluding remarks are then given in Section IV.

II. A FUZZY CONTROLLED SELECTION HYPER-HEURISTIC

A. Previous Work

Jackson et al. [15] describe a selection hyper-heuristic combining a learning heuristic selection method with late acceptance. The heuristic selection method, referred to as RUA1-F1FPS is based on objective value (fitness) proportionate selection weighting heuristics obtained with values using a scoring system. The basic idea of the F1FPS component is to rank heuristics based on their acceptance within the move acceptance criteria. Once they have been ranked, their ranks are mapped to scores from the Formula 1 racing competition used between 2003 and 2009. That is, \{1, 2, 3, 4, 5, 6, 7, 8, 9+\} \mapsto \{10, 8, 6, 5, 4, 3, 2, 1, 0\}. These scores are then used to weight each heuristic in a roulette wheel selection scheme such that favourable heuristics have a higher probability of being selected. The RUA1 component is a variant of the basic F1FPS in that the scores are reversed by ranking the heuristics with the worst scores higher than heuristics with higher scores.

The assignment of scores follows an unfair allocation scheme where each heuristic is assigned a score based on its sorted position in an array rather than sharing scores over heuristics which have equal scores. The heuristic selection method assigns scores based on the acceptance of the candidate solution produced by the heuristic being applied, and heuristics ranked \(\geq 9^{th}\) gain scores of 1 to prevent starvation of heuristics. The move acceptance method LA requires setting of a single parameter. This parameter, \(L\), controls how many iterations previous the current solution quality is compared to when deciding whether to accept or reject a solution. \(L\) in this LA implementation is fixed throughout the execution of the hyper-heuristic. This selection hyper-heuristic will be referred to as LAHH from this point onward.

In [14], it is shown that a higher list length parameter value causes the search to take longer to converge. It is also shown that a better solution could be achieved and the search takes longer to converge in some cases. Given a time contract search procedure which has to terminate within a given time limit, such as hyper-heuristics, the parameter setting of the list length, \(L\), for late acceptance is crucial. This value needs to be set sufficiently high to facilitate a sufficiently long convergence time to obtain a better solution, but without exceeding the time limit. In this study, we describe a fuzzy system to control the setting of the list length of late acceptance under the same selection hyper-heuristic framework using the same heuristic selection method described above as in [15]. This variant of LAHH embedding the fuzzy system described in Section II-C will be referred to as F-LAHH.

B. Dynamic Late Acceptance List Length

There are two options when controlling the list length parameter \(L\) in late acceptance; increasing or decreasing it. The list contains the objective function values of the solutions previously accepted in the last \(L\) iterations of the hyper-heuristic. Decreasing the list length is handled trivially by discarding the remaining entries beyond the new list length. On the other hand, increasing the list length requires a strategy for setting the values of the additional entries.

When increasing the length of the list, there are multiple possibilities for extending the array. Given the current list length \(L\) and the new list length \(M\), the previous \(L\) solution fitness values are preserved leaving the decision of how to fill the remainder of the list, from \(L+1\) to \(M\). These possibilities include randomly generating a new solution and copying its fitness function value across the extended section of the list. However this would simulate a partial random restart rather than the intended effects of controlling the list length.

Two other possibilities considered include copying the fitness value \(L\) times previously, or the worst fitness value recorded in the previous \(L\) iterations over the remainder of the list. There is one potential problem with using the fitness value \(L\) times previously. If this value was to be low relative to other fitnesses in the list, then extending the list would result in the late acceptance only accepting solutions below that threshold for \(M - L\) iterations and thus having the exact opposite effect of what is intended by increasing its size. Initial empirical analysis of both variations indicated that copying the worst fitness value (objective value) performed slightly better than copying the value \(L\) times previously.

In addition to such strategies, an exhaustive record late acceptance method was devised which maintains a full list of \(L_{\text{max}}\) previously accepted solution fitnesses and compares the candidate solution fitness with that accepted \(L\) iterations previously such that for all list lengths from 1 to \(L_{\text{max}}\), a complete record of \(L\) previous solutions is available. A theoretical and empirical comparison of the two list length extension strategies is available in Section III-B to which the overall outcome was that the method of keeping an exhaustive record of all \(L_{\text{max}}\) previous solution fitnesses was better than copying the worst solution fitness in the previous \(L\) iterations and is therefore the strategy used for extending the list length within F-LAHH.

C. Fuzzy Control of Late Acceptance List Length

Previous studies which use fuzzy systems to control various parameters within meta-heuristics used Mamdani infer-
The execution of the hyper-heuristic was split into 50 equal stages defined as the given execution time divided by 50. In each stage, the initial, $f_i$, and final, $f_o$ objective values were recorded. These were used along with the current worst solution accepted, which by the definition of late acceptance is equal to the very initial solution, $f_{\text{worst}}$. Normalised Fitness Delta is then calculated using $\Delta F = (f_i - f_o) / f_{\text{worst}}$. The lower and upper bounds of this measure are known to be $-1$ and $1$ respectively and is reflected in the universe of discourse in the NFD fuzzy set. Current Array Length is the length of the list used for late acceptance in the current stage. New Array Length is the length of the list which should be used for late acceptance in the next stage.

The fuzzy system is comprised of 15 rules (Table I). A rule is defined by three variables $C, F, N$ which relate to the fuzzy sets CAL, NFD, and NAL respectively. The rules are defined as IF (CAL = $C$) AND (NFD = $F$) THEN (NAL = $N$). When defining the rules of the system, the effects of different list lengths for late acceptance were considered along with what should happen if the search begins to stagnate. A higher value of $L$ causes the search to take longer to converge while a smaller value of $L$ will cause the search to stagnate very quickly. It has previously been shown that a longer convergence time will eventually lead to a better quality solution. Setting this parameter to a high value then would appear to be the best solution however there are other problems concerning the execution time of the hyper-heuristic and the total number of iterations. If the parameter is set too high, then the search would degrade into a random walk with a threshold value equal to the initial solution’s objective value. At any given point of the search, the optimal value of this parameter is then uncertain as to what we should assign it and needs to be controlled.

The NFD indicates if for the current stage, the search was able to intensify or diversify the search based on the stage’s first and last solution objective function values and by what ratio with respect to the current worst solution. It was decided that in any given stage, a diversification of $\geq 10\%$ with respect
to the current worst and current best solutions is considered a high amount of diversification and an intensification of 30% is considered a high amount of diversification. For these reasons, when the intensification is high, the length of the list is increased to the largest possible size. If the diversification is high, then the list length is decreased to the next smallest size. The reason we used the next smallest size rather than small for all CAL’s is because we want to prolong the convergence but prevent further diversification.

The remaining three NFD MF’s negative, neutral, and positive have different thought processes associated with design of their rules. negative and positive describe the case where there was slight intensification or slight diversification. It is unknown whether in the next stage, these slight intensification or diversification’s will continue or the search stagnates. However, we want to promote slight intensification and slight diversification as this leads to a longer convergence and thus a better quality solution. Therefore if NFD is defined as negative or positive, then NAL would equal CAL, with the exception of a large CAL and negative NFD where it was decided that the new array length should be medium to prevent too much diversification, this was also reinforced by empirical analysis of setting NAL to be medium or large in which the system with the medium NAL outperformed that with the large NAL. The neutral MF defines a stagnated search, i.e. there is no diversification or intensification during the current stage and thus the new array length is chosen to be high, independent on the current array length, to allow the search to have the chance to diversify enough to continue the search.

III. EXPERIMENTAL RESULTS

A. Experimental Setup

Hyper-heuristics herein were compared using all twelve problem instances from the MAX-SAT problem domain available in the HyFlex Framework. Each hyper-heuristic was ran 31 times on each problem instance where a run terminates after 10 nominal minutes with respect to the CHeSC 2011 competition machine which translated to 438s on our machine which uses an Intel Core i7-3820 CPU running at a default (turbo boost) clock speed of 3.70GHz with a total of 16GB of RAM. The initial list length for all variants of F-LAHH was set to 10000. The results of each instance for both approaches were compared using the Wilcoxon signed-rank test with 95% confidence interval as a statistical test to determine if and which approach was on average the best. The objective function values for MAX-SAT problem solutions, as used in each table of results, equals the number of broken clauses and thus is a minimisation problem where a value of 0 indicates that the solution satisfies all clauses.

B. Late Acceptance List Length Extension Methods

Two extension methods were considered for extending the list length of late acceptance and were used within the fuzzy control late acceptance based hyper-heuristic detailed in Section II-C. First a strategy which uses the worst solution fitness value in the current list of L previously accepted solutions, referred to herein as F-LAHH_{max}, and a second method which maintains an exhaustive list of L_{max} previously accepted solution fitnesses, referred to herein as F-LAHH_{exh} and shortened as F-LAHH_{exh}.

From a theoretical standpoint, both extension methods have their own advantages and disadvantages. In the initial stage(s), F-LAHH_{max} can be seen as a bad approach since when the list of previous solution fitnesses is shortened and then lengthened, the data concerned is lost and replaced with the maximum element in the remaining list. This can inhibit the amount of diversification of the search in the initial stages leading to premature convergence and bad performance, particularly for those problem instances which require more diversification in the initial stage(s) for good performance such as larger instances and those which are characteristically harder to solve. On the other hand, F-LAHH_{exh} preserves these fitness values and will allow more diversification compared to F-LAHH_{max} and overcomes the issue imposed by that strategy. In the latter stages however, F-LAHH_{exh} can be seen as disadvantageous when compared to F-LAHH_{max} because late acceptance will stagnate in its final stages causing what can be visualised as a list of closely oscillating fitness values. This means that F-LAHH_{exh} rejects more worsening moves which are required to prevent the search from stagnating and to progress to find possibly better quality solutions. This is particularly true for smaller instances, or those which are easier to solve, where the search can quickly stagnate and an extension method which allows more diversification during these stages is desired such as F-LAHH_{max}.

Both extension methods were independently made use of in the fuzzy controlled late acceptance based hyper-heuristic F-LAHH and were compared using the experimental setup detailed in Section III-A to find out if and which extension approach was the best. The results, shown in Table II, show that F-LAHH_{exh} is significantly better than F-LAHH_{max} across four instances but also significantly worse for two other instances. In general, F-LAHH_{exh} is better than F-LAHH_{max} over half of the instances (six) and worse for the other half, albeit a higher proportion of improving instances were statistically significantly better compared to F-LAHH_{max}. Median performance analysis shows that for all but three problem instances, F-LAHH_{exh} either improves or equals the median performance of F-LAHH_{max} which is an improvement over the F-LAHH_{max} strategy. For each 31 runs, the objective function value of the best overall solution was recorded. F-LAHH_{exh} and F-LAHH_{max} had identical best solution fitnesses for ten of the twelve instances and F-LAHH_{exh} was able to find a better quality solution for one of the instances with an objective function value of 19 compared to 20 of F-LAHH_{max} and vice versa with fitnesses of 2 versus 3 for one other instance. Based on these results, it was therefore decided that the F-LAHH_{exh} approach was to be used in the rest of the study for the comparison of a fuzzy controlled late acceptance based hyper-heuristic to a fixed length counterpart.

Theoretical observations aforementioned about the advantages and disadvantages of both F-LAHH_{exh} and F-LAHH_{max}
strategies that F-LAHH$_{exh}$ should perform better for larger instances than smaller instances, and that F-LAHH$_{max}$ should perform better for smaller instances than larger instances are shown empirically in Fig. 3 which shows a clear distinction between the sizes of an instance, and whether F-LAHH$_{exh}$ should perform better for larger instances than smaller instances, and that F-LAHH$_{max}$ should perform better for smaller instances than larger instances are shown empirically in Fig. 3 which shows a clear distinction between the sizes of an instance, and whether F-LAHH$_{exh}$ will improve over the performance of F-LAHH$_{max}$ based on their mean average performance. There is therefore a trade-off when using either strategy between being able to solve small or large instance sizes.

C. Fuzzy Controlled F-LAHH$_{exh}$ versus Fixed Length LAHH

LAHH with list length $L_{min} = 13267$ and $L_{max} = 26733$, i.e. the minimum and maximum values output by the fuzzy system, were compared and the best setting selected for comparison with F-LAHH to ensure that if F-LAHH demonstrated any improvement, then it is due to the parameter control. F-LAHH was therefore compared to LAHH with fixed list length, $L_{max}$, on all twelve instances of the MAX-SAT problem domain using the same experimental setup as before used in Section III-A. In this section herein, F-LAHH$_{exh}$ is simply referred to as F-LAHH. The results of each instance for LAHH and F-LAHH were compared to determine if F-LAHH has any significant improvement over the fixed, uncontrolled LAHH on average. F-LAHH was also compared to the best performing hyper-heuristic for the MAX-SAT domain from the ChESE 2011 competition, AdapHH [24], on five of the twelve problem instances as were used in the competition.

The results summarised in Table III show that this initial fuzzy system was able to significantly improve over the best fixed length hyper-heuristic for two instances. Being an initial, un-tuned fuzzy system to illustrate the potential of parameter control using fuzzy systems in hyper-heuristic’s, the fuzzy system also performed insignificantly better, insignificantly worse, and significantly worse for five, four, and one instance respectively. Overall, the fuzzy controlled late acceptance hyper-heuristic was able to perform better for seven of the twelve instances. Median performance analysis shows that F-LAHH equalled the performance of LAHH for nine out of twelve instances and for the remaining three instances, LAHH found solutions with one less broken clause for each. Best performance analysis (the fitness of the best solution found over 31 runs) shows that F-LAHH matched the performance of LAHH for all instances.

As well as being able to make some improvements over LAHH, the objective function values of the best runs in Table IV show that it is able to match the performance of the best hyper-heuristic from the ChESE competition for solving MAX-SAT problem instances, although median results show that AdapHH is better than F-LAHH for two other instances of the competition. In the ChESE competition, hyper-heuristics were awarded scores based on their median performances for each problem instance of each problem domain relative to those of all other entrants and so due to F-LAHH’s median performance, AdapHH would still be declared the better hyper-heuristic using the competition’s scoring system.

The progress plot of the late acceptance list length, objective function values of the best and current solution at each stage entry is shown in Fig. 4 for one of the best runs for the instance#2 (for which F-LAHH performs well compared to LAHH). From this plot, we can see that the fuzzy system controls the list length in each stage to allow an adequate amount of diversification and intensification improving the quality of the solution in hand. A general trend was observed where the list length on average tended to increase over time, from between about 18000 and 24000 in the initial stages (with a few spikes approaching 26000) to between 23000 and 26000 in the latter stages, and the amount by which the list length changes decreased over time. The graph clearly shows how the list length is adapted based on the amount of improvement of the current solution compared to the solution current in the previous stage to prevent too much intensification or diversification. If the solution current in a stage improves compared to the previous stage, the list length increases to

![Table II](image)

TABLE II

| Instance # | F-LAHH$_{exh}$ Median Avg. | vs. | F-LAHH$_{max}$ Median Avg. |
|------------|-----------------------------|----|-----------------------------|
| 0          | 5                           |     | 5                           |
| 1          | 26                          | <   | 33                          |
| 2          | 20                          | <   | 23                          |
| 3          | 3                           | ≥   | 2                           |
| 4          | 3                           | ≥   | 2                           |
| 5          | 6                           | ≤   | 7                           |
| 6          | 6                           | ≥   | 6                           |
| 7          | 6                           | ≤   | 6                           |
| 8          | 8                           | ≥   | 8                           |
| 9          | 211                         | >   | 211                         |
| 10         | 3                           | ≤   | 4                           |
| 11         | 9                           | >   | 8                           |

![Fig. 3](image)

Fig. 3. Number of variables and clauses in each satisfiability problem showing instances improved (upwards triangles), and worsened (downwards triangles), for list length acceptance strategy F-LAHH$_{exh}$ compared to F-LAHH$_{max}$.

Two improving instances occupy (700,3500) and two worsening instances occupy (300,1200).
TABLE III
Performance comparison of F-LAHH and LAHH with $L = L_{\text{MAX}}$ using objective function values of the best solution found for each run over 31 runs for each HyFlex MAX-SAT instance. $A \leq \text{vs.} B$: $A < B$ ($A > B$) indicates that $A$ ($B$) is better than $B$ ($A$) and this performance difference is statistically significant within a 95% confidence interval based on the Wilcoxon signed-rank test. $A \leq B$ ($A \geq B$) indicates that $A$ ($B$) performs slightly better than $B$ ($A$) but is not a significant improvement.

| Instance # | LAHH | F-LAHH |
|------------|------|--------|
|            | Median | Mean  | vs. | Median | Mean  |
| 0          | 5     | 4.77   | <   | 5     | 5.26  |
| 1          | 26    | 26.71  | <   | 26    | 29.35 |
| 2          | 20    | 20.38  | <   | 20    | 22.94 |
| 3          | 3     | 3.39   | ≥   | 3     | 3.71  |
| 4          | 3     | 3.07   | ≥   | 2     | 2.94  |
| 5          | 5     | 6.32   | ≤   | 5     | 7.16  |
| 6          | 6     | 6.07   | ≤   | 6     | 6.16  |
| 7          | 6     | 6.32   | ≥   | 6     | 6.23  |
| 8          | 8     | 8.19   | <   | 8     | 8.29  |
| 9          | 211   | 211.61 | >   | 211   | 211.06 |
| 10         | 3     | 3.81   | ≥   | 2     | 3.16  |
| 11         | 9     | 8.77   | ≥   | 8     | 8.65  |

TABLE IV
Performance comparison of F-LAHH to AdapHH, the best hyper-heuristic from the CHeSC 2011 competition for solving MAX-SAT problem instances, using objective function values of the best solution found for each run over 31 runs using all problem instances from the final round of the CHeSC 2011 competition.

| Instance # | F-LAHH | AdapHH |
|------------|--------|--------|
|            | Best   | Median | Best   | Median |
| 3          | 1      | 3      | 1      | 3      |
| 4          | 1      | 3      | 1      | 2      |
| 5          | 3      | 5      | 3      | 5      |
| 10         | 1      | 3      | 1      | 3      |
| 11         | 7      | 9      | 7      | 8      |

prevent premature convergence and allow diversification in the current stage. Conversely, if the solution in the current stage worsens compared to the previous stage, then the list length is decreased to prevent too much diversification which would prevent the search to converge on any good quality solutions. As the search process finds better solutions and proceeds to converge, the list length setting tends to increase so to prevent the search from stagnating too early.

Two other traces show the importance of correctly controlling the list length. On one hand, the worst run of the best instance in Fig. 5 shows that too much diversification is allowed throughout the duration of the run and results in the solution taking too long to improve enough to find a good solution in the given time limit. On the other hand, the worst run of one of the worst instances in Fig. 6 did not allow enough diversification in the initial stages and therefore converged too quickly resulting in solutions whose quality was worse than if more diversification was allowed. It was also observed that instances where F-LAHH performs badly stagnates in the first half of the search, caused by lack of diversification, and good runs on good instances never truly stagnates, in fact, given more time it is likely that the search will find an even better quality solution for such instances.

As can be seen in Fig. 5, there is initially a large amount of room for allowing much worse solutions than the current solution. The fuzzy control system tries to intensify the search by greatly reducing the current list length however the problem occurs when the solution improves at such a rate (required given the search state). The fuzzy system increases the list length too much trying, what it thinks, to prevent stagnation, but in reality the feature of the current search state means that a lower list length is required than it thought. This “confusion” causes a repeating pattern of greatly increasing and decreasing the list length over the whole duration of the search thus frequently allowing worsening solutions and results in a lack of intensification, leading ultimately to poor performance.

Traces for runs of instances where F-LAHH did not perform well suggested various areas of improvement. Some runs of good instances with poor performance allowed too much diversification throughout the whole execution of the hyper-heuristic resulting in more of a random walk nature and the search is never made to intensify enough to converge on good solutions. This was attributed with frequent and erratic changes in the list length throughout the whole run and does not share the same nature of tendency to increase over time as with the instances where F-LAHH performed well. This phenomena is illustrated in Fig. 5 for the instance#2. A feature of the worst runs of instances where F-LAHH did not perform well was that solutions of acceptable quality were found within the initial stage and very little diversification was facilitated in successive stages, as can be seen in Fig. 6, causing the best solution found by the search not to be very good compared to runs where this was not the case. This suggests that either the list length is too small in the initial stage or in every stage in general, or that a finer grain of control is needed and therefore the number of stages should be increased.

IV. Conclusions and Future Work

The initial fuzzy system effectively controlled the single parameter of late acceptance in the F-LAHH hyper-heuristic which was able to improve the results of seven of the twelve instances, significantly so for two of these. This indicates that by using fuzzy logic to control the parameter of late acceptance, we are able to improve the resulting hyper-heuristic.

This initial design contains many other parameters which currently use a fixed setting such as the number of stages, the length of each stage, and the initial list length. In future work, such parameters should also be controlled as their settings effect the effectiveness of the fuzzy system. The number of stages that the execution of the hyper-heuristic is split into influences the number of times the fuzzy system is invoked. If this setting is too low, the system would not have chance to change the size of the list length and the hyper-heuristic may have already stagnated causing relatively poor solutions to be found whereas if this setting is too high, there are two factors which effect the overall performance, one being the execution time of the fuzzy system taking away too much time from the application of the low-level heuristics, and the other being...
that the number of heuristic applications with respect to the list length is too small for the change to have any effect.

Analysis of run traces shows bad runs of bad instances quickly find acceptable solutions and do not allow sufficient diversification to find better solutions whereas poor performing runs for good instances allow too much diversification throughout the whole search and therefore take longer than the allocated time to converge on a good quality solution hence its comparatively poor performance. Inclusion of other search state measures are therefore considered for future work to overcome this problem.

The definitions of the fuzzy sets work for the MAX-SAT problem domain and show promising room for improvement, however for a higher-level hyper-heuristic which works well across multiple domains, F-LAHH may or may not perform well. The definitions of the fuzzy sets are uncertain, especially for a higher-level hyper-heuristic. Use of type-2 fuzzy sets to overcome these problems are therefore considered for future work.

REFERENCES

[1] E. K. Burke, M. Gendreau, M. Hyde, G. Kendall, G. Ochoa, E. Özcan, and R. Qu, “Hyper-heuristics: A survey of the state of the art,” Journal
of the Operational Research Society, vol. 64, no. 12, pp. 1695–1724, 2013.

[2] E. Özcan, B. Bilgin, and E. E. Korkmaz, “Hill climbers and mutational heuristics in hyperheuristics,” in Proceedings of the International Conference on Parallel Problem Solving From Nature (PPSN 2006), ser. LNCS, vol. 4193. Reykjavik, Iceland: Springer, 2006, pp. 202–211.

[3] E. K. Burke, T. Curtois, M. Hyde, G. Kendall, G. Ochoa, S. Petrovic, and J. A. Vazquez-Rodriguez, “Hyflex: A flexible framework for the design and analysis of hyper-heuristics,” in Proceedings of the Multidisciplinary International conference on Scheduling: Theory and Applications (MISTA 2009). Dublin, Ireland, 2009, pp. 790–797.

[4] G. Ochoa, M. Hyde, T. Curtois, J. Vazquez-Rodriguez, J. Walker, M. Gendreau, G. Kendall, B. McCollum, A. Parkes, S. Petrovic, and E. Burke, “Hyflex: A benchmark framework for cross-domain heuristic search,” in Proceedings of Evolutionary Computation in Combinatorial Optimization (EvoCOP 2012), ser. LNCS, vol. 7245. Malaga, Spain: Springer, 2012, pp. 136–147.

[5] A. Eiben, R. Hinterding, and Z. Michalewicz, “Parameter control in evolutionary algorithms,” Evolutionary Computation, IEEE Transactions on, vol. 3, no. 2, pp. 124–141, Jul 1999.

[6] L. Zadeh, “Fuzzy sets,” Information and Control, vol. 8, no. 3, pp. 338 – 353, 1965.

[7] D.-P. Tian and N.-Q. Li, “Fuzzy particle swarm optimization algorithm,” in Artificial Intelligence, 2009. JCAI ’09. International Joint Conference on, 2009, pp. 263–267.

[8] Y. Shi and R. Eberhart, “Fuzzy adaptive particle swarm optimization,” in Evolutionary Computation, 2001. Proceedings of the 2001 Congress on, vol. 1, 2001, pp. 101–106.

[9] F. Herrera and M. Lozano, “Adaptive control of the mutation probability by fuzzy logic controllers,” in Parallel Problem Solving from Nature PPSN VI, ser. LNCS. Springer Berlin Heidelberg, 2000, vol. 1917, pp. 335–344.

[10] A. Alsawy and H. Hefny, “Fuzzy-based ant colony optimization algorithm,” in Computer Technology and Development (ICCTD), 2010 2nd International Conference on, 2010, pp. 530–534.

[11] C. Li, J. Yu, and X. Liao, “Fuzzy tabu search for solving the assignment problem,” in Communications, Circuits and Systems and West Sino Expositions, IEEE 2002 International Conference on, vol. 2, 2002, pp. 1151–1155.

[12] H.-B. Xu, H.-J. Wang, and C.-G. Li, “Fuzzy tabu search method for the clustering problem,” in Machine Learning and Cybernetics, 2002. Proceedings. 2002 International Conference on, vol. 2, 2002, pp. 876–880 vol.2.

[13] E. K. Burke and Y. Bykov, “A late acceptance strategy in hill-climbing for exam timetabling problems,” in Proceedings of the International Conference on the Practice and Theory of Automated Timetabling (PATAT 2008), Montreal, Canada, 2008, p. Extended Abstract.

[14] ——, “The late acceptance hill-climbing heuristic,” University of Stirling, Tech. Rep., 2012.

[15] W. Jackson, E. Özcan, and J. Drake, “Late acceptance-based selection hyper-heuristics for cross-domain heuristic search,” in Computational Intelligence (UKCI), 2013 13th UK Workshop on, Sept 2013, pp. 228–235.

[16] J. H. Drake, E. Özcan, and E. K. Burke, “Controlling crossover in a selection hyper-heuristic framework,” School of Computer Science, University of Nottingham, Tech. Rep. No. NOTTCS-TR-SUB-1104181638-4244, 2011.

[17] E. Özcan, Y. Bykov, M. Birben, and E. K. Burke, “Examination timetabling using late acceptance hyper-heuristics,” in Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2009). Trondheim, Norway: IEEE Press, 2009, pp. 997–1004.

[18] P. Demeester, B. Bilgin, P. D. Causmaecker, and G. V. Bergh, “A hyperheuristic approach to examination timetabling problems: Benchmarks and a new problem from practice,” Journal of Scheduling, vol. 15, no. 1, pp. 83–103, 2012.

[19] M. Misir, W. Vancroonenburg, K. Verbeek, and G. V. Bergh, “A selection hyper-heuristic for scheduling deliveries of ready-mixed concrete,” in Proceedings of the Metaheuristics International Conference (MIC 2011), Udine, Italy, 2011, pp. 289–298.

[20] C. Wagner, “Juzzy - a java based toolkit for type-2 fuzzy logic,” in Advances in Type-2 Fuzzy Logic Systems (T2FUZZ), 2013 IEEE Symposium on, April 2013, pp. 45–52.

[21] M. Hyde and G. Ochoa, “Hyflex competition instance summary,” summary of problem domains and instances used in the CHeSC 2011 Competition.

[22] E. Mamdani and S. Assilian, “An experiment in linguistic synthesis with a fuzzy logic controller,” International Journal of Man-Machine Studies, vol. 7, no. 1, pp. 1 – 13, 1975.

[23] F. Valdez, P. Melin, and O. Castillo, “Fuzzy control of parameters to dynamically adapt the pso and ga algorithms,” in Fuzzy Systems (FUZZ), 2010 IEEE International Conference on, 2010, pp. 1–8.

[24] M. Misir, P. D. Causmaecker, G. V. Bergh, and K. Verbeek. (2011) An adaptive hyper-heuristic for ceshc 2011.