Jackson, Warren G. and Özcan, Ender and John, Robert I. (2014) Fuzzy adaptive parameter control of a late acceptance hyper-heuristic. In: 14th UK Workshop on Computational Intelligence UKCI2014, 8-10 Sept 2014, University of Bradford.

# 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)"*. This framework provides six pre-implemented problem domains allowing researchers to concentrate on the development and analysis 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 for improved performance. 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] from

CHeSC 2011.

The rest of this paper is organised as follows. In Sect. II, a description of a late acceptance hyper-heuristic and its variant embedding a fuzzy system are provided. The empirical results discussing the performance of the fuzzy controlled late acceptance hyper-heuristic is presented in Sect. III. Concluding remarks are then given in Sect. 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 the late acceptance method 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 Sect. II-B will be referred to as F-LAHH.

### B. Fuzzy Control of Late Acceptance List Length

There are two options when controlling the list length parameter $L$ in late acceptance; increasing or decreasing it. Assuming that $L = N$, the list contains the objective function values of the visited solutions in the last $N$ 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 $N$ and the new list length $M$, the previous $N$ solution fitness values are preserved leaving the decision of how to fill the remainder of the list, from $N+1$ 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 late acceptance.

Two other possibilities considered include copying the fitness value $N$ times previously, or the worst fitness value recorded in the previous $N$ iterations over the remainder of the list. There is one potential problem with using the fitness value $N$ times previously. If this value was to be low compared to other fitnesses in the list, then extending the list would result in the late acceptance only accepting solutions below that threshold for $M - N$ 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 $N$ times previously and was therefore used in the F-LAHH.

Previous studies which use fuzzy systems to control various parameters within meta-heuristics used Mamdani inference [22], Centroid defuzzification, and in the majority of these studies used either Triangular or Gaussian membership functions. In one such study [23], it was reported that empirical analysis using both types of membership functions showed that Triangular membership functions gave better performance over Gaussian ones. Therefore, in this study the fuzzy system uses Mamdani type inference with the minimum t-norm, maximum t-conorm, and performs defuzzification using the Centroid method. It is a two input, one output system composed of three fuzzy sets where the two inputs were *current array length* (CAL), and *normalised fitness delta* (NFD) and the output was *new array length* (NAL) each with 3, 5 and 3 membership functions (referred to as MF's from herein) respectively. Initial experiments using 3 MF's for NFD had relatively poor performance hence 5 MF's were used to define NFD. The output of the fuzzy system has to be discretised to an integer value which is used for the new list length holding previous objective values in the late acceptance. Discretisation was performed by rounding to the nearest whole number. The input CAL has three triangular MF's *small*, *medium*, and *large* and covered the universe of discourse $U = [10000, 30000]$ and is illustrated in Fig. 1 along with the output NAL which

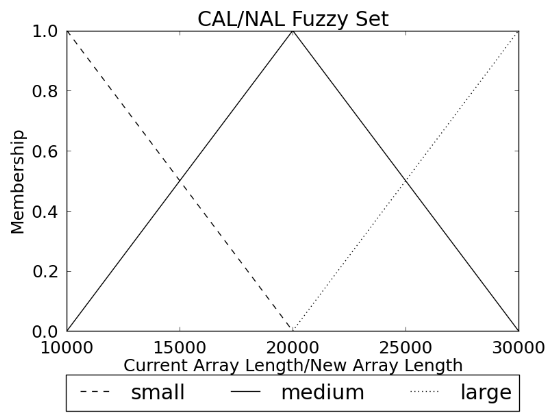


Fig. 1. Fuzzy Sets for Current Array Length (CAL) and New Array Length (NAL)

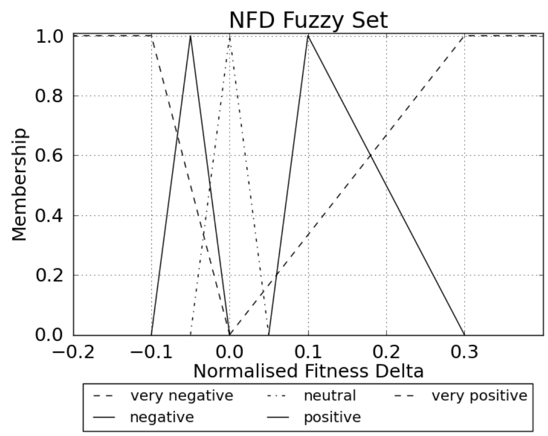


Fig. 2. Fuzzy Set for Normalised Fitness Delta (NFD)

was defined in the same way as CAL using three triangular MF's *small*, *medium*, and *large* spans the universe of discourse $U = [10000, 30000]$. The input NFD has five MF's, two trapezoidal, and three triangular. These spanned the universe of discourse $U = [-1, 1]$ and were called *very negative*, *negative*, *neutral*, *positive*, and *very positive* and were defined as follows and illustrated in Fig. 2. Note that the MF *very negative* extends to $-1.0$ and *very positive* extends to $1.0$, however, for clarity, the figure only shows the range of $[-0.2, 0.4]$.

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 $\text{NFD} = (f_i - f_o)/f_{\text{worst}}$ such that 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

TABLE I
IF-THEN RULES USED IN THE FUZZY SYSTEM.

| | CAL | | |
|---|---|---|---|
| NFD | small | medium | large |
| very negative | small | small | medium |
| negative | small | medium | medium |
| neutral | large | large | large |
| positive | small | medium | large |
| very positive | large | large | large |

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 beings 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 more optimal 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, combined with the method of increasing the list length, this increase is favoured.

## III. Experimental Results

LAHH with list length $L_{\text{min}} = 13267$ and $L_{\text{max}} = 26733$, 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, $L_{\text{max}}$, on all twelve instances of the HyFlex MAX-SAT problem domain. Only five of those twelve instances were actually used in determining the winner of the CHeSC 2011 Competition. Each hyper-heuristic was ran 31 times on each problem instance. 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 F-LAHH was set to the best length of 10000 from the set of tested lengths, {10000, $L_{min}$, $L_{max}$, and 30000}. The results of each instance for LAHH and F-LAHH were compared using the Wilcoxon signed-rank test as a statistical test 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 CHeSC 2011 competition, AdapHH [24] on the relevant competition instances. The objective function value is the number of broken clauses in the solution and this, therefore, is a minimisation problem and 0 indicates that the solution satisfies all clauses.

The results summarised in Table II 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 three, four, and three instances respectively. Overall, the fuzzy controlled late-acceptance hyper-heuristic was able to perform better for five of the twelve instances. As well as being able to make some improvements over LAHH, the objective function values of the best runs in Table III show that it is able to improve over AdapHH, although median results show that while improving for one instance of the competition, it performed worse for two others, albeit for one of these, it managed to obtain a better best solution than AdapHH. In the CHeSC competition, hyper-heuristics were awarded scores based on their median performances for each problem instance of each problem domain

TABLE II
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 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 # | F-LAHH Best | F-LAHH Mean | vs. | LAHH Mean | LAHH Best |
|---|---|---|---|---|---|
| 0 | 2 | 7.48 | $\geq$ | 5.26 | 2 |
| 1 | 20 | 40.68 | $>$ | 29.35 | 19 |
| 2 | 15 | 31.39 | $>$ | 22.94 | 15 |
| 3 | 1 | 2.97 | $<$ | 3.71 | 1 |
| 4 | 1 | 3.07 | $\geq$ | 2.94 | 1 |
| 5 | 2 | 11.23 | $\geq$ | 7.16 | 3 |
| 6 | 5 | 6 | $\leq$ | 6.16 | 5 |
| 7 | 5 | 6.45 | $\geq$ | 6.23 | 5 |
| 8 | 5 | 7.81 | $\leq$ | 8.29 | 5 |
| 9 | 209 | 211 | $\leq$ | 211.06 | 209 |
| 10 | 1 | 4.61 | $>$ | 3.16 | 1 |
| 11 | 7 | 8.35 | $<$ | 8.65 | 7 |

TABLE III
Performance comparison of F-LAHH with AdapHH, the best hyper-heuristic for solving MAX-SAT problem instances in the CHeSC 2011 competition, using objective function values of the best solution found for each run over 31 runs using the problem instances used in the final round of the CHeSC 2011 competition

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

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 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. 3 during the best run for the instance#2 (for which F-LAHH performs well). 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 tended to increase over time, from about 22000 in the initial stages to about 26000 in the latter stages, and the amount by which the list length was changed decreased until the search stagnated, at which point changing the list length would have no effect and therefore the fuzzy system makes no change to the list length. On the other hand, it is observed that the worst run on this instance did not allow enough diversification and therefore converged too quickly resulting in solutions whose quality was worse than if more diversification was allowed.

Traces for runs of instances where F-LAHH did not perform well suggested various areas of improvement. The best runs

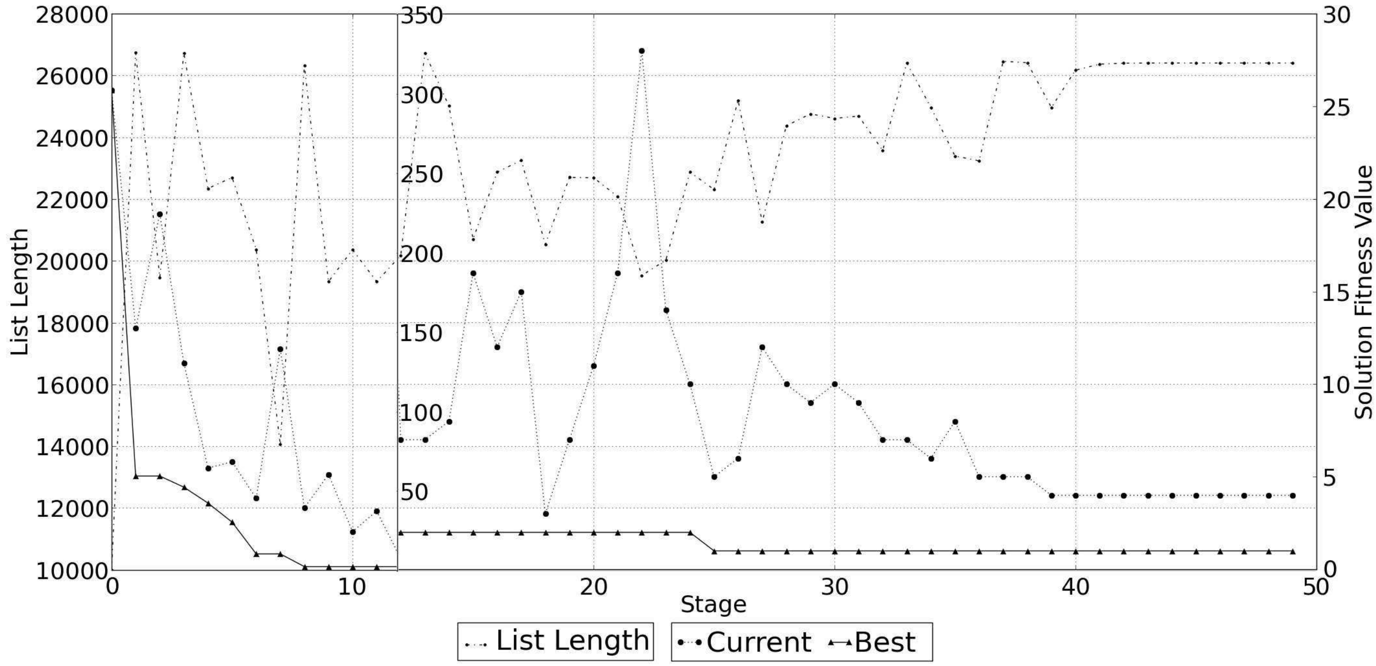


Fig. 3. Trace of list length, objective function value of the best and current solutions at the entry of each stage of the best run for the instance#2.

of which did not allow too much diversification, resulting in acceptable solutions, which is in contrast to the best instance, on par with LAHH however some runs 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 between about $14000$ and $26000$ throughout the whole run and does not share the same nature of tending to increase over time as with the instances where F-LAHH performed well. This phenomena is illustrated in Fig. 4 for the instance#3. A feature of the worse runs of the instance which F-LAHH did not perform well was that the amount of improvement during the initial stage was small compared to good runs which caused the value of NFD to be associated with the MF *positive* rather than a larger improvement which is associated, with membership $1.0$, to the MF *very positive* in the NFD fuzzy set. This meant that the fuzzy system set the list length for the second stage smaller than that given by a higher NFD value and resulted in bad solutions.

## IV. Conclusions and Future Work

The initial fuzzy system to adaptively control the single parameter of late acceptance in the F-LAHH hyper-heuristic was able to improve the results of five of the twelve instance, 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 is an initial design with 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 prematurely converged causing sub-optimal 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. From initial analysis of the traces, we also found that there are cases where the fuzzy system sets the list length too high or too low which causes too much or too little diversification and leads to bad quality solutions. Particularly bad runs showed that too much diversification is allowed throughout the whole run which could also be due to the method of deciding which values to use when increasing the list length and so, in future work, this value could be decided by a fuzzy system.

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. These definitions of these fuzzy sets are uncertain, especially for a higher-level hyper-heuristic. Therefore, use of type-2 fuzzy sets to overcome these problems are considered for future work.

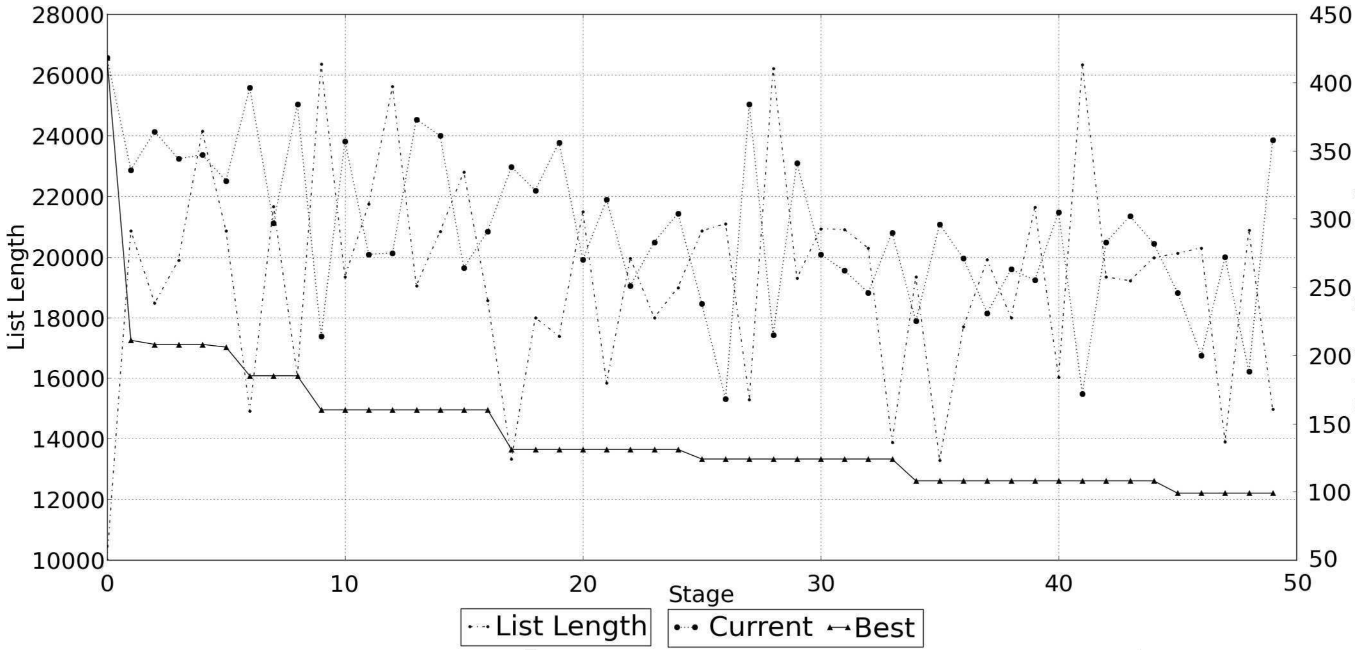


Fig. 4. Trace of list length, objective function values of the best and current solutions at the entry of each stage of the worst run for the instance#3.

## 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, T. P. Runarsson, H.-G. Beyer, E. K. Burke, J. J. M. Guervós, L. D. Whitley, and X. Yao, Eds., 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. Lecture Notes in Computer Science, M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. Merelo, and H.-P. Schwefel, Eds. 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 vol.2.

[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] E. K. Burke and Y. Bykov, "The late acceptance hill-climbing heuristic," University of Stirling, Tech. Rep., 2012.

[15] W. Jackson, E. Ozcan, 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. Berghe, "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. Verbeeck, and G. V. Berghe, "A selection hyper-heuristic for scheduling deliveries of ready-mixed concrete," in *Proceedings of the Metaheuristics International Conference (MIC 2011)*, L. D. Gaspero, A. Schaerf, and T. Stützle, Eds., 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. Berghe, and K. Verbeeck. (2011) An adaptive hyper-heuristic for chesc 2011.