# A Comparative Study of Fuzzy Parameter Control in a General Purpose Local Search Metaheuristic

Warren G. Jackson, Ender Özcan, 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*—There is a growing number of studies on general purpose metaheuristics that are directly applicable to multiple domains. Parameter setting is a particular issue considering that many of such search methods come with a set of parameters to be configured. Fuzzy logic has been used extensively in control applications and is known for its ability to handle uncertainty. In this study, we investigate the potential of using fuzzy systems to control the parameter settings of a threshold accepting (TA) metaheuristic for improving the overall effectiveness of a cross-domain approach. We have evaluated the performance of various general purpose local search metaheuristics which mix multiple heuristics at random and apply the TA metaheuristic with fixed threshold, crisp (non-fuzzy) rule-based control of the threshold and various fuzzy systems controlling the threshold. The empirical results show that the approach using the TA with crisp rule-based control performs the best across six problem domains from a benchmark.**

## I. Introduction

Heuristic search methods are commonly preferred in optimisation to find good quality solutions of computationally hard problems [1] in a reasonable amount of time. A goal within the research field involves designing effective *cross-domain search methods*. Traditionally, metaheuristics were employed for solving problem instances from a given problem domain however this meant that when a new problem needed to be solved, the given search method required modification to make it work well on a different problem. Cross-domain search entails solving problems from characteristically different problem domains, thus an effective cross-domain search method must be able to perform well across all problem domains *without modification*. The development of such cross-domain search methods was the focus of a recent competition, the Cross-domain Heuristic Search Challenge (CHeSC) 2011, where *selection hyper-heuristics* were used for solving optimisation problems from six different domains [2]. Selection hyper-heuristics are high-level search methodologies that mixes a set of predefined low level heuristics and can be applied to different problems without modification. Özcan et al. [3] decomposed single-point based selection hyper-heuristics into two key components; a *heuristic selection method* and a *move acceptance criterion*. Emphasis within hyper-heuristic research has been focused on devising more effective intelligent heuristic selection methods through the use of machine learning techniques. It is however common that the choice of the move acceptance criterion is overlooked with "off the shelf" metaheuristics being used as their move acceptance component. One problem with metaheuristics and so hyper-heuristics is that the best settings of their parameters at any given time are uncertain and is essential for good performance [4]. This issue becomes a greater challenge within cross-domain search because good settings for their parameters can change between different problems.

Fuzzy logic [5] has been used extensively in control applications and has also been employed for controlling the parameters of a number of population-based metaheuristic search methods, each for solving a specific optimisation problem including mathematical function optimisation [6], [7], [8], travelling salesman problem [9], the assignment problem [10], and the clustering problem [11], improving their performances with respect to their non-fuzzy counterparts.

We recently employed fuzzy logic to control the list length parameter of a late acceptance based hyper-heuristic for solving maximum satisfiability problems in [12] however its success was marginal compared to its counterpart with a fixed list length. A drawback of the aforementioned study is that the fuzzy system for controlling the late acceptance move acceptance component was developed in the presence of a heuristic selection method which could potentially work against the move acceptance. For example, the fuzzy controller may increase the list length parameter to promote diversification but the heuristic selection method could choose local search heuristics.

It has been shown in aforementioned studies that fuzzy logic can be used within population-based search methods to improve their performance for solving a single problem. The aim of this study is to investigate the potential for fuzzy logic to be used to improve the performance of cross-domain search methods by controlling the parameters of a move acceptance method which are uncertain not only during different stages of the search, but also between different problem types. In this study, we develop a Threshold Accepting *local search metaheuristic* [13] for cross-domain search (XD-TA). XD-TA uses uniform random heuristic selection to eliminate the effects of learning mechanisms used in the heuristic selection criteria

**Algorithm 1:** Pseudo-code for the Threshold Accepting metaheuristic for minimisation.

```
1  $s \leftarrow generateInitialSolution()$;
2  $s_{best} \leftarrow s$;
3  $T \leftarrow$ initial threshold $> 0$;
4  while termination criteria not met do
5      $s' \leftarrow random \in \mathcal{N}(s)$;
6      if $f(s') < f(s) + T$ then
7          $s \leftarrow s'$;
8      end if
9      if $f(s') < f(s_{best})$ then
10         $s_{best} \leftarrow s'$;
11     end if
12     if a long time has passed without improvement or too many iterations have passed then
13         $T \leftarrow decrease(T)$;
14     end if
15 end while
16 return $s_{best}$
```

within hyper-heuristics. The threshold parameter of XD-TA is then either fixed, controlled using a crisp (non-fuzzy) method, or controlled using a fuzzy system. These three strategies are then used to conclude whether controlling parameters of move acceptance methods using fuzzy logic can improve cross-domain performance. Moreover, the crisp strategy is used as an experimental control to show whether parameter control is a confounding factor of any cross-domain improvement using fuzzy logic.

The rest of this paper is structured as follows. The Threshold Accepting (TA) metaheuristic is defined in Section II along with an addition to TA for cross-domain search (XD-TA). In Section III we define the three variations of XD-TA used in the experimentation and outline the experimental methods. The results of the study are given in Section IV and the paper is concluded in Section V.

## II. Threshold Accepting

In this study, we use a modified Threshold Accepting (TA) algorithm for cross-domain search. In this section, we begin by introducing the original TA algorithm. We then propose a modification to TA, as inspired by a modification made to a very similar metaheuristic, to make it suitable for use within cross-domain search which is then used in the empirical study.

### A. The Threshold Accepting Metaheuristic

Threshold Accepting (TA) is a metaheuristic proposed by Dueck and Scheuer in [14]. The pseudo-code for TA is provided in Algorithm 1 which is reformulated for minimisation.

At each iteration of the search, a solution is randomly chosen from the neighbourhood ($\mathcal{N}(.)$) of the current solution ($s$) as the candidate solution ($s'$) where the neighbourhood is defined by perturbations of the current solution. TA then replaces the current solution with the candidate solution if and only if the objective value of the candidate solution ($f(s')$) is better than an acceptance threshold ($\tau$) calculated as the sum of the objective value of the current solution and threshold parameter ($f(s) + T$). Otherwise, TA rejects the candidate solution and the current solution stays the same for the next iteration where a different neighbouring solution can be chosen. The definition of TA provided by Dueck also mentions some mechanism for controlling the threshold parameter by decreasing $T$ whenever no improvement is made for some amount of time or a number of iterations have elapsed. An exact control mechanism however is not given.

### B. Threshold Accepting for Cross-domain Search

One of the problems faced when developing cross-domain search methods is caused by the differing ranges of objective values between different problem domains. The threshold parameter, $T$, within TA has to be set appropriately depending on the current problem. A suitable range of values for $T$ in one domain may be too large in another domain, causing too much diversification thus simulating a random walk of the search space, and in another domain may be too small causing a lack of diversification simulating hill-climbing local search.

A modification was made to the Record-to-Record Travel (RRT) algorithm, introduced by Dueck in [15], by Mısır et al. in [16] which they called Iteration Limited Threshold Accepting (ILTA). The only difference between TA and RRT is in the calculation of the acceptance threshold, $\tau$, with all other aspects of both methods being algorithmically identical. In TA, the acceptance threshold is calculated as the sum of the objective value of the *current* solution and a threshold parameter. Within RRT, the objective value of the *best* solution found so far is used rather than the current solution. In [16], the threshold parameter is encoded as a factor of the best solution found so far using the parameter, $\epsilon$. The threshold parameter is therefore dependent on the objective value of an existing solution for any given problem. The acceptance threshold in ILTA is calculated as $\tau = \epsilon \times f(s_{best}) + f(s_{best}) = (1+\epsilon) \times f(s_{best})$. A similar approach was also used for cross-domain search in [17] where they calculate $\tau$ as $(1 + \epsilon) \times f(s_{stageBest})$ in a stage-based hyper-heuristic. Since the only difference in RRT compared to TA is the use of best solution found so far rather than the current solution, a modification of TA making it more suitable for cross-domain search is trivial, calculating the acceptance threshold as $\tau = (1 + \epsilon) \times f(s)$.

Another issue with TA arises due to $\tau$ being updated at each iteration. If $\tau$ is updated every iteration, it is likely that a sequence of solutions are accepted which are progressively worse than the previous causing a high degree of diversification within the search leading to poor performance. A multi-stage hyper-heuristic approach is used in [17] for cross-domain search. In the multi-stage hyper-heuristic, the search is split into a number of stages. In each "S1HH stage", a record of the best solution found in the current stage is used to calculate an acceptance threshold synonymous to that used in ILTA except $f(s_{stageBest})$ is used instead of $f(s_{best})$. To counteract the aforementioned issue, we introduced a second parameter to

**Algorithm 2:** Pseudo-code for the Cross-domain Threshold Accepting metaheuristic (XD-TA).

```
Input: IPS, $\epsilon$
1  $s \leftarrow generateInitialSolution()$;
2  $s_{best} \leftarrow s$;
3  while termination criteria not met do
4      $s' \leftarrow random \in \mathcal{N}(s)$;
5      if iterations == IPS then
6          $\epsilon \leftarrow update(\epsilon)$;
7          $\tau = (1 + \epsilon) \times f(s)$;
8          iterations = 0;
9      end if
10     if $f(s') \leq \tau$ then
11         $s \leftarrow s'$;
12     end if
13     if $f(s') \leq f(s_{best})$ then
14         $s_{best} \leftarrow s'$;
15     end if
16     iterations++;
17 end while
18 return $s_{best}$
```

XD-TA inspired by the multi-stage method called *iterations per stage* (IPS). In XD-TA, we therefore split the search into a number of stages whose length is defined by the number of iterations, IPS. The acceptance threshold is then only updated at the start of each stage using the same formula as before such that an upper limit is maintained for the duration of each stage. The pseudo-code for XD-TA is shown in Algorithm 2.

As with TA, at each iteration of the search, a solution is randomly chosen from the neighbourhood ($\mathcal{N}(.)$) of the current solution ($s$) as the candidate solution ($s'$) where the neighbourhood is defined by perturbations of the current solution. XD-TA then checks to see if a new stage has begun (Line 5 of Algorithm 2) in which case $\epsilon$ is updated by the control mechanism, if any, and $\tau$ is updated as $\tau = (1+\epsilon) \times f(s)$. XD-TA then replaces the current solution with the candidate solution if and only if the objective value of the candidate solution ($f(s')$) is better than the acceptance threshold ($\tau$) for the respective stage. Otherwise, XD-TA rejects the candidate solution and the current solution stays the same for the next iteration where a different neighbouring solution can be chosen.

## III. Experimentation

In this study, we aim to show whether fuzzy logic can be used to improve the performance of a Threshold Accepting cross-domain search method (XD-TA) by controlling its $\epsilon$ parameter. Moreover, whilst it could be observed that fuzzy does improve the cross-domain performance, we also want to be able to show whether control of the $\epsilon$ parameter is a confounding factor of any improvement of the fuzzy control.

The XD-TA metaheuristic search method was implemented in Java using the HyFlex Framework [18] which is a software framework designed to "*enable the development, testing, and comparison of iterative general-purpose heuristic search algorithms*". The HyFlex Framework consists of a total of 6 problem domains, each of which has a set of 10-12 problem instances, and provides a set of low-level heuristics, also known as move operators, for each problem domain. These heuristics belong to one of four types, *local search*, *mutation*, *ruin-recreate*, and *crossover*.

All XD-TA algorithms used in this study use the same underlying definition as explained in Section II-B with the only differences being in the parameter setting for IPS, and the implementation of the $update(\epsilon)$ method controlling the $\epsilon$ threshold parameter. The neighbourhood of a given current solution is defined by the underlying low-level heuristics of the HyFlex Framework. For the purpose of this study, crossover type heuristics were omitted due to the requirement of managing a population of solutions suitable to use for the crossover operation, and previous experience of using such heuristics degrading the performance of cross-domain search methods when random heuristic selection is used.Note that for efficiency reasons, rather than generating the whole neighbourhood of any given solution, the choice of a random neighbouring solution is implemented by choosing a random low-level heuristic and applying it to the current solution.

Three main variants of XD-TA had to be developed to meet the aims of the study. Firstly, XD-TA was implemented using a fixed setting of $\epsilon$ (XD-TA-1) to determine a baseline result of the XD-TA search method. The second version of XD-TA used a crisp (non-fuzzy) rule-based system for controlling the setting of $\epsilon$ (XD-TA-2) which is used to show whether control in general, or specifically fuzzy control, is required for improving the cross-domain performance of XD-TA. Finally a fuzzy system was designed for controlling the parameter setting of $\epsilon$ (XD-TA-3). The implementations of *update($\epsilon$)*, as shown in Line 6 of Algorithm 2, for the three aforementioned versions of XD-TA are defined in Sections III-A, III-B, and III-C respectively. $\epsilon$ was initially set to 0.00 in XD-TA-2 and XD-TA-3 however for XD-TA-1, the initial setting for $\epsilon$ was set to its best setting.

The experimental design for tuning of the three versions of XD-TA, and the final comparison of the cross-domain performance of each is as follows. For the parameter tuning experiments, a total of 8 problem instances were chosen consisting of one small and one large problem instance from each of the 4 "public domains". These were One-dimensional Bin Packing (BP) instances 1 and 11, Permutation Flow Shop (FS) instances 1 and 11, Personnel Scheduling (PS) instances 5 and 9, and Maximum Satisfiability (SAT) instances 5 and 11, as detailed in the CHeSC instance summary[1]. For XD-TA-1 and XD-TA-2, the parameters were tuned by evaluating all combinations of considered parameter settings, as detailed in their respective definitions, comparing all configurations over 31 trials per configuration each with a time based

[1]http://www.asap.cs.nott.ac.uk/external/chesc2011/reports/CHeSCInstance-Summary.pdf

termination criteria of 600 seconds as defined by the CHeSC 2011 competition using a Kruskal-Wallis One-way ANOVA test, referred to herein as a KW-ANOVA test, with $n_0$ that the results obtained by all configurations come from the same distribution. Note that the Kruskal-Wallis test was used since the results obtained did not always come from a normal distribution and hence required a non-parametric variant of the standard ANOVA. The configuration which most frequently appeared in the best performing group of configurations over the 8 training instances was then chosen to be best. Initial tuning of XD-TA-3 followed the same format as the tuning of XD-TA-1 and XD-TA-2 using initial definitions of the fuzzy sets, as detailed in the description for XD-TA-3, and a fuzzy rule-base which emulates XD-TA-2. Since there was a much larger number of design choices, resulting in an exponential increase of possible configurations, from this point on in the design of XD-TA-3, each aspect of the design was considered in order where the best configuration of each was used in the subsequent iteration of improvement.

To evaluate the cross-domain performance of the three parameter control techniques; fixed $\epsilon$, crisp controlled $\epsilon$, and fuzzy controlled $\epsilon$, all three variants of XD-TA-1, XD-TA-2 and XD-TA-3 were evaluated using their best configurations against a total of 30 problem instances, 5 problem instances each from 6 problem domains, as used in the CHeSC 2011 competition, detailed in the CHeSC instance summary (see previous footnote[1]), as these instances were chosen by the organisers to best reflect the effectiveness of cross-domain search methods. Each method was ran a total of 31 times per instance for the same 600 CHeSC competition seconds as in the tuning experiments and the results. To reduce the chances of making a type-1 statistical error, all three methods were compared using a Kruskal-Wallis ANOVA test, referred to herein as KW-ANOVA, to show not only whether fuzzy control of XD-TA can improve its cross-domain performance, but also whether control in general is a confounding factor of the improvement. That is, if fuzzy control can improve over a fixed XD-TA, does a crisp control system also improve its performance and if so, is the fuzzy approach statistically significantly better than the crisp mechanism.

### A. *XD-TA with Fixed $\epsilon$ (XD-TA-1)*

In this version of XD-TA, $\epsilon$ was fixed for the duration of the search and thus $update(\epsilon)$ simply returned the current value of $\epsilon$. The parameters of $\epsilon$ and `IPS` used for the fixed XD-TA were derived from a combinatorial parameter tuning experiment where each considered parameter setting of $\epsilon$ was paired with each setting considered for `IPS`. Within this study, there were four possible choices of $\epsilon \in \{0.01, 0.05, 0.10, 0.15\}$, as used in the crisp and fuzzy XD-TA's below, and two possible choices of `IPS` $\in \{100, 1000\}$ forming 8 possible configurations of XD-TA. These 8 configurations were tested against a total of 8 training problem instances which consisted of one large and one small sized instance each selected from 4 different problem domains. A Kruskall-Wallis one-way ANOVA test between all 8 configurations over 8 training instances consisting of one small and one large problem instance, each from 4 problem domains showed that the best configuration was with $\epsilon = 0.01$ and `IPS` $= 100$ whose results are shown in Table I. Note that with the objective of minimising, the better configurations are those that have *smaller* mean ranks.

TABLE I
MEAN RANKS FOR A KRUSKAL-WALLIS ONE-WAY ANOVA TEST BETWEEN EACH CONFIGURATION OF XD-TA-1 WITH THE BEST PERFORMING AND NON-STATISTICALLY SIGNIFICANTLY DIFFERING CONFIGURATIONS FOR EACH INSTANCE SHADED GREY.

| Problem | IPS = 100 | | | | IPS = 1000 | | | |
|---|---|---|---|---|---|---|---|---|
| $\epsilon$ | 0.01 | 0.05 | 0.10 | 0.15 | 0.01 | 0.05 | 0.10 | 0.15 |
| BP #1 | 18.29 | 81.32 | 175.32 | 191.61 | 44.71 | 111.77 | 184.35 | 188.61 |
| BP #11 | 16.77 | 147.32 | 147.65 | 163.21 | 46.23 | 137.19 | 175.65 | 161.98 |
| FS #1 | 32.69 | 128.95 | 161.13 | 182.10 | 38.23 | 120.63 | 159.73 | 173.55 |
| FS #11 | 42.40 | 125.58 | 158.85 | 171.84 | 60.26 | 130.53 | 142.73 | 163.81 |
| PS #5 | 139.47 | 143.95 | 122.39 | 125.29 | 123.31 | 113.61 | 114.81 | 113.18 |
| PS #9 | 96.05 | 119.98 | 86.50 | 92.19 | 137.71 | 151.26 | 148.68 | 163.63 |
| SAT #5 | 229.00 | 194.97 | 117.26 | 73.63 | 128.00 | 117.06 | 84.10 | 51.98 |
| SAT #11 | 196.71 | 195.29 | 100.90 | 37.44 | 161.42 | 166.42 | 95.92 | 41.90 |

**Algorithm 3:** Pseudo-code for the $update(\epsilon)$ method used in the crisp rule-based XD-TA. Note that if *iterations* is not a multiple of `IPS` then the value of $\epsilon$ remains unchanged.

```
Input: $\epsilon$, IPS, CNIM, MAX_CNIM
$\epsilon s = 0.00, 0.01, 0.05, 0.10, 0.15$;
if iterations mod IPS == 0 then
    if CNIM $\geq$ MAX_CNIM then
        $\epsilon \leftarrow$ next highest value of $\epsilon \in \epsilon s$;
    else
        $\epsilon \leftarrow 0.00$;
end if
return $\epsilon$
```

### B. *XD-TA with Crisp Rule-based Control of $\epsilon$ (XD-TA-2)*

An adaptive mechanism using only crisp rules was created for adaptively controlling the $\epsilon$ parameter of XD-TA. The crisp rule-based system selected a value of $\epsilon$ from the set of $\epsilon$ values tested for the fixed XD-TA method including 0.00 for when the control method decides that a threshold value of 0.00 should be used. That is, $\epsilon \in \{0.00, 0.01, 0.05, 0.10, 0.15\}$. An additional parameter, the current number of consecutive non-improving moves (CNIM), was introduced to the XD-TA method which the control system uses to determine whether the value of $\epsilon$ should be increased or reset to its initial setting. The $update(\epsilon)$ method in the crisp adaptive XD-TA was then defined as shown in Algorithm 3.

The parameters of the system were tuned using the same systematic approach used for tuning the fixed XD-TA method. Within the crisp adaptive XD-TA however we have introduced two additional parameters, `MAX_CNIM` and `INCR`. Three settings were considered for `MAX_CNIM` as $0.05 \times$ `IPS`, $0.10 \times$ `IPS`, and $0.20 \times$ `IPS`. The two incremental strategies, `INCR`, are used to determine when `CNIM` is increased. The generation `INCR` strategy increments `CNIM` whenever a move is rejected whereas the acceptance `INCR` strategy increments `CNIM` whenever a move is accepted but the cost of the accepted solution is not better than that of the current solution. Unlike with XD-TA-1, there was no clear best performing configuration of XD-TA-2. A Kruskall-Wallis one-way ANOVA test between all 12 configurations, whose results are given in Table II, showed that a number of configurations performed well. In general, using the acceptance `INCR` strategy over the generation strategy improved the cross-domain performance of each configuration of `IPS` and `CNIM`. Within the results for the acceptance strategy there did not appear to be a correlation between the performance of the resulting search method and the `IPS` or `CNIM` parameter settings. Two configurations of the crisp controlled XD-TA algorithm were in the best performing group of configurations for 7 out of the 8 training instances with the rest only appearing in such a group between 4 and 6 times. The two best configurations were with the acceptance incremental strategy and when {`IPS` = 100; `CNIM` = 20% of `IPS`} and {`IPS` = 1000; `CNIM` = 5% of `IPS`}. The best configuration for XD-TA-2 was chosen to be when `INCR` = `ACC`, `IPS` = 1000, and `CNIM` = $0.05 \times$ `IPS` = 50 since it performed the best, in terms of its mean ranks, for 2 instances compared to only 1 instance for the other configuration.

### C. XD-TA with Fuzzy Control of $\epsilon$ (XD-TA-3)

Finally, a third variant of XD-TA was developed which uses fuzzy logic to control the threshold parameter, $\epsilon$. The fuzzy systems developed in this study were implemented using the Java version of fuzzylite [19]. Initially, a fuzzy system was designed to emulate the crisp control system using the $\epsilon$ values as constant outputs within a Sugeno (TSK) fuzzy inference system using the MIN and MAX t-norm and t-conorm in the antecedents, and algebraic product for the activation operator in the consequents. The initial fuzzy system, referred to herein as XD-TA-3-I, is a two input-one output system which uses the current value of $\epsilon$ and the number of consecutive number of non-improving moves (`CNIM`) as its inputs, and its output is the new value of $\epsilon$ to be used. As stated before, the fuzzy system uses TSK inference and therefore the output of the fuzzy system is defined trivially using 5 constants, one for each $\epsilon$ setting as used in XD-TA-2. The two input fuzzy sets were designed using triangular membership functions (TriMF's). The first, `current-`$\epsilon$, was designed using 5 TriMF's, one for each $\epsilon$ setting with $U = [0, 0.15]$, where each membership function (MF) had a membership of 1.0 at the point of its respective $\epsilon$ value, and each MF crossed with adjacent MF's at membership 0.5 as shown in Figure 1. The `CNIM` fuzzy set had $U = [0,$ `CNIM_MAX`$]$ and was designed in the same way as `current-`$\epsilon$ but consisted of 2 MF's to decrease the size of the rule-base and hence decrease the complexity of the fuzzy rules and cost penalty on the search method of evaluating the fuzzy system. In the event that the current value of `CNIM` exceeds `CNIM_MAX`, then `CNIM` takes the value of `CNIM_MAX`. The fuzzy rules of XD-TA-3-I are shown in Table III.

XD-TA-3-I was then tuned by evaluating all combinations of `IPS`, `INCR` and `CNIM` as in XD-TA-2 where the upper bound of $U$ of the `CNIM` fuzzy set is defined as `CNIM_MAX` $\leftarrow$ `CNIM`. These results show that the best configuration of XD-TA-3-I was `INCR` = `ACC`, `IPS` = 1000, and `CNIM_MAX` = 20 however XD-TA-3-I performed poorly compared to XD-TA-2. We therefore rendered the control surface of XD-TA-3-I which showed that the value of $\epsilon$ was reduced very slowly as `CNIM` decreased potentially causing the poor performance which we put down to the design of the `CNIM` fuzzy set. The `CNIM` fuzzy set was then redesigned using Z- and S-curve MF's such that $\epsilon$ decreased at a faster rate, and the parameters of the `LOW` and `HIGH` MF's were modified to be (`CNIM_MAX` / 2, `CNIM_MAX`), producing XD-TA-3-II. XD-TA-3-II was compared to XD-TA-3-I using the best parameter settings from the previous tuning experiment and showed that XD-TA-3-II performed better over the 8 training instances and the new definition of the `CNIM` fuzzy set was used in subsequent designs of the fuzzy system for XD-TA-3 as shown in Figure 2.

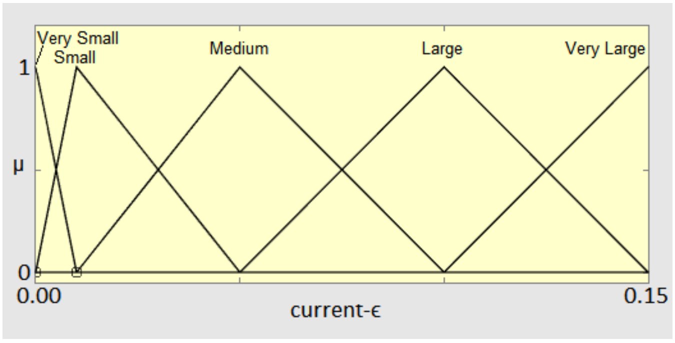

Fig. 1. Input fuzzy set `current-`$\epsilon$ as used in XD-TA-3.

TABLE III
FUZZY TSK RULES USED IN XD-TA-3-I IN THE FORMAT
IF `CURRENT-`$\epsilon$ IS _ AND CNIM IS _ THEN `NEW-`$\epsilon$ = _.

| | | CNIM | |
|---|---|---|---|
| | | LOW | HIGH |
| `current-`$\epsilon$ | VERY SMALL | 0.00 | 0.01 |
| | SMALL | 0.00 | 0.05 |
| | MEDIUM | 0.00 | 0.10 |
| | HIGH | 0.00 | 0.15 |
| | VERY HIGH | 0.00 | 0.15 |

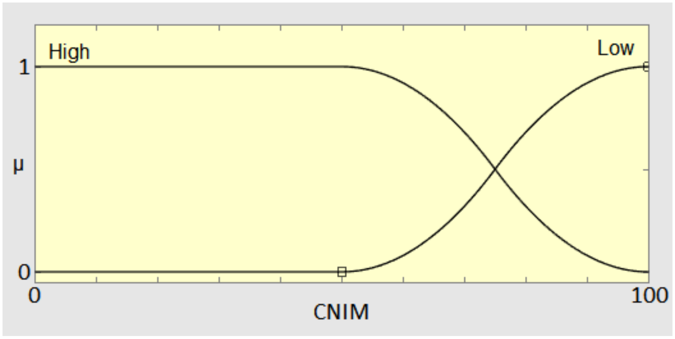

Fig. 2. Input fuzzy set `CNIM` as used in XD-TA-3.

TABLE II
Comparison of all configurations of `IPS`, `CNIM`, and `INCR` for XD-TA-2 using a Kruskal-Wallis One-way ANOVA with CI = 95%.

| | Generation Strategy | | | | | | Acceptance Strategy | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| IPS | 100 | | | 1000 | | | 100 | | | 1000 | | | $\chi^2(11)$ | p-value |
| CNIM | 5 | 10 | 20 | 5 | 10 | 20 | 5 | 10 | 20 | 5 | 10 | 20 | | |
| BP #1 | 332.10 | 220.32 | 100.84 | 350.13 | 266.68 | 166.52 | 285.61 | 123.19 | 63.87 | 21.10 | 197.90 | 109.74 | 336.55 | $1.70\times10^{-65}$ |
| BP #11 | 327.00 | 163.53 | 163.53 | 355.61 | 285.90 | 174.61 | 261.94 | 48.42 | 16.00 | 82.39 | 220.74 | 138.32 | 342.03 | $1.18\times10^{-66}$ |
| FS #1 | 261.53 | 223.73 | 178.47 | 267.39 | 232.63 | 154.34 | 191.39 | 135.02 | 154.35 | 125.29 | 193.71 | 120.16 | 77.07 | $5.42\times10^{-12}$ |
| FS #11 | 248.76 | 199.53 | 189.81 | 193.40 | 169.73 | 166.90 | 198.56 | 187.23 | 178.63 | 168.87 | 167.15 | 169.44 | 15.97 | $1.42\times10^{-1}$ |
| PS #5 | 210.10 | 198.82 | 181.11 | 153.39 | 150.94 | 172.98 | 222.00 | 234.47 | 219.10 | 134.02 | 168.23 | 193.16 | 29.88 | $1.70\times10^{-3}$ |
| PS #9 | 146.16 | 124.29 | 170.98 | 211.89 | 182.35 | 193.05 | 153.06 | 152.15 | 160.02 | 244.16 | 254.65 | 245.24 | 55.93 | $5.23\times10^{-8}$ |
| SAT #5 | 153.15 | 218.45 | 243.92 | 106.71 | 146.13 | 163.65 | 190.34 | 228.16 | 274.82 | 184.79 | 153.74 | 174.15 | 67.20 | $1.15\times10^{-10}$ |
| SAT #11 | 138.68 | 191.87 | 201.97 | 182.19 | 194.27 | 184.44 | 168.03 | 175.92 | 197.29 | 220.58 | 184.87 | 197.89 | 13.19 | $2.81\times10^{-1}$ |

TABLE IV
Fuzzy TSK Rules used in XD-TA-3(-V) in the format
If `current`-$\epsilon$ IS _ and `CNIM` IS _ Then `new`-$\epsilon$ = _.

| | | CNIM | |
|---|---|---|---|
| | | LOW | HIGH |
| `current-`$\epsilon$ | VERY SMALL | 0.00 | 0.01 |
| | SMALL | 0.00 | 0.05 |
| | MEDIUM | 0.01 | 0.10 |
| | HIGH | 0.05 | 0.15 |
| | VERY HIGH | 0.10 | 0.15 |

The performance of the fuzzy system was also improved by experimenting with different rule-bases. These were designed by instead of setting the output to be 0.00 if `CNIM` is `LOW`, then we decrease the output by choosing the next smallest $\epsilon$ value, XD-TA-3-III, and next next smallest, XD-TA-3-IV. An ANOVA test showed that despite considering the current value of $\epsilon$ when `CNIM` was `LOW`, XD-TA-3-II, XD-TA-3-III and XD-TA-3-IV did not perform statistically significantly different for 6 out of 8 training instances with XD-TA-3-II performing significantly better than both others for Bin Packing problems. This could have been due to the gradual decrease of $\epsilon$ rather than the instant decrease used in XD-TA-2. We then had the idea that the performance of XD-TA-3-III could exceed that of XD-TA-3-II if we control $\epsilon$ continually, that is `IPS` = 1, rather than using the stage based approach used in XD-TA-2. Since the value of $\epsilon$ is now updated at each iteration, the setting for `CNIM_MAX` used in previous versions may be too low as it is checked 1000 times more frequent. We therefore increased `CNIM_MAX` to be 100, the addition of which and the use of continual control constitutes a fifth version XD-TA-3-V.

The configuration, XD-TA-3-V, of the fuzzy system was used as the final system, referred to herein as just XD-TA-3, which, to recap, has the following parameter settings. IPS = 1, INCR = GEN, MAX_CNIM = 100, uses the input fuzzy sets shown in Figures 1 and 2, with the output described at the beginning of this section. The TSK rules of XD-TA-3-V are stated in Table IV, and the overall control surface is illustrated in Figure 3.

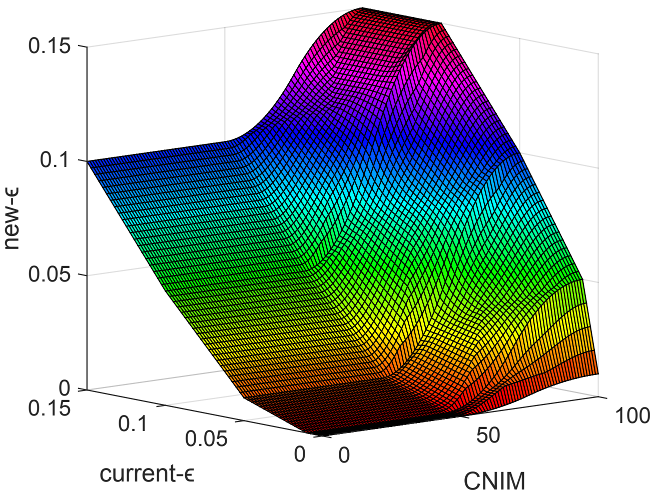

Fig. 3. Control surface of fuzzy system used to control $\epsilon$ in XD-TA-3.

## IV. Results

In this study, we wanted to see whether fuzzy logic could be used to improve the performance of a cross-domain search method by controlling the parameters of its move acceptance. Moreover, we wanted to test whether control of its parameters is a confounding factor of improvement in the cross-domain effectiveness of the resulting search method.

We therefore performed a Kruskal-Wallis One-way ANOVA test on all three variants of XD-TA consisting of the best configurations for no control of $\epsilon$ (XD-TA-1), control of $\epsilon$ using a crisp rule-based system (XD-TA-2), and control of $\epsilon$ using a fuzzy control system (XD-TA-3). This test would show us which control mechanism, if any, performed better than the remaining mechanisms for each problem instance. The null hypothesis, $h_0$, of the KW-ANOVA test states that the data comes from the same distribution. In terms of our analysis acceptance of $h_0$ would show that XD-TA-1, XD-TA-2, and XD-TA-3 do not perform statistically significantly different from one another, thus the control mechanism has no effect on performance. If $h_0$ is however rejected in favour of $h_a$, then at least one of the control mechanisms performs statistically significantly better than the remaining mechanisms. In this case, a post-hoc test is used to compare the significance between the differences of each mechanism. The results of the KW-ANOVA test are shown in Table VI where the best approach, or group of approaches, for each instance are shaded in grey. The best approach for solving each instance in terms of their mean rank test statistic is shaded in dark grey. Note however that there is no statistical significance between those shaded dark grey and those in light grey.

The KW-ANOVA test showed that the crisp controlled version of XD-TA (XD-TA-2) was by far the best for cross-domain search. The difference between the performance of all three variants of XD-TA was not significant for 3 of the 30 instances with the test failing to reject $h_0$ that the mean ranks are from the same distribution. Moreover, these 3 instances were from the same problem domain, TSP, for which XD-TA-1 consistently appeared in the best group of approaches, showing that the control of $\epsilon$ is generally ineffective for this problem. For the remaining 27 instances, the test statistic rejected $h_0$ showing that at least one method performed statistically significantly better than the other(s). XD-TA-2 was present in the best performing group of XD-TA variants 23 times out of 30 compared to just 12 and 8 for XD-TA-3 and XD-TA-1 respectively. Moreover, XD-TA-2 performed statistically significantly better than both XD-TA-1 and XD-TA-3 for 13 of these instances. This shows that while the fuzzy control of $\epsilon$ was able to improve the cross-domain performance of XD-TA, control in general was a confounding factor of the improvement. In this case, a crisp rule-based control mechanism was able to perform much better than the fuzzy controlled algorithm. Additionally, XD-TA-3 showed promising results for solving Bin Packing problems appearing one of the best more frequently than other approaches, suggesting that fuzzy control can improve the performance of a search method, albeit not the cross-domain performance.

TABLE VI
KRUSKAL-WALLIS ONE-WAY ANOVA COMPARING THE PERFORMANCE OF XD-TA-1, XD-TA-2, AND XD-TA-3 WITH $n_0$ THAT ALL RESULTS ARE FROM THE SAME DISTRIBUTION AT CI = 95%.

| Instance | | XD-TA-1 | XD-TA-2 | XD-TA-3 | $\chi^2(2)$ | p |
|---|---|---|---|---|---|---|
| BP | 7 | 78.00 | 22.26 | 40.74 | 68.61 | $1.26\times10^{-15}$ |
| | 1 | 20.58 | 66.39 | 54.03 | 47.80 | $4.17\times10^{-11}$ |
| | 9 | 78.00 | 47.00 | 16.00 | 81.79 | $1.74\times10^{-18}$ |
| | 10 | 78.00 | 37.13 | 25.87 | 64.04 | $1.24\times10^{-14}$ |
| | 11 | 78.00 | 47.00 | 16.00 | 81.79 | $1.74\times10^{-18}$ |
| SAT | 3 | 77.08 | 20.29 | 43.63 | 70.01 | $6.29\times10^{-16}$ |
| | 5 | 76.10 | 23.44 | 41.47 | 61.20 | $5.14\times10^{-14}$ |
| | 4 | 74.45 | 20.39 | 46.16 | 62.51 | $2.66\times10^{-14}$ |
| | 10 | 78.00 | 25.00 | 38.00 | 70.65 | $4.56\times10^{-16}$ |
| | 11 | 73.52 | 33.58 | 33.90 | 47.07 | $6.01\times10^{-11}$ |
| PS | 5 | 42.81 | 28.56 | 69.63 | 37.18 | $8.43\times10^{-9}$ |
| | 9 | 50.82 | 67.26 | 22.92 | 42.76 | $5.18\times10^{-10}$ |
| | 8 | 36.00 | 45.48 | 59.52 | 11.92 | $2.60\times10^{-3}$ |
| | 10 | 49.89 | 25.87 | 65.24 | 33.52 | $5.27\times10^{-8}$ |
| | 11 | 44.50 | 20.31 | 76.19 | 67.02 | $2.80\times10^{-15}$ |
| FS | 1 | 45.29 | 22.53 | 73.18 | 54.83 | $1.24\times10^{-12}$ |
| | 8 | 69.52 | 29.92 | 41.56 | 35.27 | $2.20\times10^{-8}$ |
| | 3 | 53.29 | 26.08 | 61.63 | 30.99 | $1.86\times10^{-7}$ |
| | 10 | 61.11 | 32.61 | 47.27 | 17.30 | $1.75\times10^{-4}$ |
| | 11 | 56.11 | 30.98 | 53.90 | 16.49 | $2.63\times10^{-4}$ |
| TSP | 0 | 47.52 | 37.90 | 55.58 | 6.67 | $3.57\times10^{-2}$ |
| | 8 | 48.76 | 49.31 | 42.94 | 1.06 | $5.88\times10^{-1}$ |
| | 2 | 26.90 | 62.26 | 51.84 | 28.09 | $7.95\times10^{-7}$ |
| | 7 | 51.35 | 46.74 | 42.90 | 1.52 | $4.67\times10^{-1}$ |
| | 6 | 46.63 | 48.27 | 46.10 | 0.11 | $9.47\times10^{-1}$ |
| VRP | 6 | 70.42 | 54.58 | 16.00 | 66.68 | $3.32\times10^{-15}$ |
| | 2 | 78.00 | 16.19 | 46.81 | 81.28 | $2.24\times10^{-18}$ |
| | 5 | 78.00 | 46.45 | 16.55 | 80.37 | $3.54\times10^{-18}$ |
| | 1 | 49.90 | 16.00 | 75.10 | 74.85 | $5.59\times10^{-17}$ |
| | 9 | 78.00 | 16.61 | 46.39 | 80.20 | $3.84\times10^{-18}$ |

## V. CONCLUSIONS AND FUTURE WORK

Three variants of a Threshold Accepting algorithm modified for cross-domain search were developed and tuned. The three variants, known as XD-TA-1, XD-TA-2, and XD-TA-3, used different strategies for controlling a threshold parameter of XD-TA. XD-TA-1 used a fixed setting for $\epsilon$ throughout the search representing a baseline XD-TA with no control. Two further algorithms, XD-TA-2 and XD-TA-3 were designed using a crisp rule-based control mechanism and a fuzzy control mechanism respectively. The different variants were compared across a total of 30 problem instances taken from 6 different problem domains to represent the cross-domain search problem. Analysis showed that while a fuzzy control system can be used to improve the cross-domain performance of XD-TA(-1), a much simpler crisp control system, requiring much less time to develop and tune, was able to outperform both XD-TA-1 and XD-TA-3.

Fuzzy logic has been used in the past for controlling the parameters of population-based metaheuristics for solving a variety of computationally hard problems, but were only used for solving a particular type of problem. Our results also indicate potential for fuzzy logic to be used to improve the performance of single-point based local search metaheuristics by controlling their parameters for solving a single type of problem. Fuzzy logic in these experiments has not performed as expected, failing to improve over the crisp control system at the cross-domain level. Our results indicate that fuzzy logic on its own is no more suitable for cross-domain search than other control methods. Within cross-domain search, local search metaheuristics that perform well often employ strategies for tuning and/or controlling their parameters based on the features of the problem being solved. Fuzzy systems themselves introduce a set of parameters. In retrospect, fine tuning of the membership functions through learning mechanisms may have produced different results and is an area for future research.

Whilst fuzzy logic was not able to improve the cross-domain performance of XD-TA by controlling its *parameters*, this does not rule out other research directions for using fuzzy logic in other ways. Fuzzy logic has been previously applied to the problem models themselves such as in the Vehicle Routing problem where fuzzy sets were used to model imprecise variables such as travel time and transportation costs [20]. Cross-domain search methods however operate at a higher level than the heuristic level meaning fuzzy logic cannot be used in this way. Guided Local Search (GLS) works by augmenting the objective function using crisp penalty functions. The cross-domain performance of GLS could potentially be improved by replacing these with fuzzy penalty functions.

## REFERENCES

[1] M. R. Garey and D. S. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness.* New York, NY, USA: W. H. Freeman & Co., 1990.

TABLE V

MEAN AVERAGE, BEST RESULTS, AND STANDARD DEVIATION OF RESULTS OBTAINED BY EACH VARIANT OF XD-TA OVER 31 TRIALS.

| Problem | | XD-TA-1 | | | XD-TA-2 | | | XD-TA-3 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Instance | | Mean | Best | SD | Mean | Best | SD | Mean | Best | SD |
| BP | 7 | 0.1860 | 0.1761 | 0.0049 | 0.0474 | 0.0413 | 0.0049 | 0.0531 | 0.0490 | 0.0021 |
| | 1 | 0.0074 | 0.0067 | 0.0006 | 0.0107 | 0.0080 | 0.0014 | 0.0102 | 0.0076 | 0.0018 |
| | 9 | 0.0298 | 0.0270 | 0.0010 | 0.0141 | 0.0126 | 0.0010 | 0.0083 | 0.0069 | 0.0007 |
| | 10 | 0.1256 | 0.1250 | 0.0003 | 0.1110 | 0.1104 | 0.0003 | 0.1109 | 0.1105 | 0.0004 |
| | 11 | 0.0541 | 0.0470 | 0.0024 | 0.0356 | 0.0340 | 0.0011 | 0.0156 | 0.0127 | 0.0011 |
| SAT | 3 | 17.10 | 10.00 | 4.06 | 7.29 | 4.00 | 1.70 | 10.00 | 8.00 | 1.65 |
| | 5 | 38.32 | 13.00 | 14.33 | 10.74 | 6.00 | 3.50 | 13.42 | 6.00 | 2.46 |
| | 4 | 25.19 | 8.00 | 11.55 | 6.00 | 1.00 | 2.54 | 9.90 | 6.00 | 2.53 |
| | 10 | 19.06 | 12.00 | 3.70 | 5.58 | 2.00 | 1.23 | 6.39 | 6.00 | 0.67 |
| | 11 | 11.87 | 8.00 | 1.78 | 8.81 | 7.00 | 1.01 | 8.87 | 7.00 | 0.96 |
| PS | 5 | 23.35 | 17.00 | 4.18 | 20.77 | 13.00 | 2.89 | 28.55 | 18.00 | 4.99 |
| | 9 | 9694.61 | 9516.00 | 114.32 | 9794.94 | 9444.00 | 130.97 | 9524.74 | 9300.00 | 123.76 |
| | 8 | 3205.48 | 3145.00 | 49.71 | 3213.74 | 3167.00 | 39.26 | 3263.68 | 3147.00 | 77.78 |
| | 10 | 1774.77 | 1460.00 | 225.45 | 1612.03 | 1470.00 | 94.27 | 1878.19 | 1550.00 | 183.39 |
| | 11 | 365.48 | 325.00 | 24.40 | 337.13 | 310.00 | 13.11 | 858.71 | 370.00 | 725.12 |
| FS | 1 | 6295.61 | 6278.00 | 6.58 | 6286.35 | 6270.00 | 7.86 | 6305.42 | 6289.00 | 6.05 |
| | 8 | 26885.19 | 26838.00 | 19.01 | 26849.77 | 26784.00 | 22.35 | 26859.26 | 26788.00 | 24.99 |
| | 3 | 6365.26 | 6345.00 | 6.59 | 6359.55 | 6339.00 | 6.62 | 6366.97 | 6344.00 | 6.18 |
| | 10 | 11473.94 | 11435.00 | 14.15 | 11454.26 | 11418.00 | 17.81 | 11464.90 | 11432.00 | 14.05 |
| | 11 | 26688.32 | 26658.00 | 17.73 | 26661.65 | 26608.00 | 27.39 | 26684.61 | 26619.00 | 27.06 |
| TSP | 0 | 48563.02 | 48271.93 | 250.98 | 48476.11 | 48286.15 | 86.86 | 48553.41 | 48427.38 | 63.73 |
| | 8 | $2.153\times10^7$ | $2.123\times10^7$ | $2.608\times10^5$ | $2.154\times10^7$ | $2.112\times10^7$ | $3.052\times10^5$ | $2.145\times10^7$ | $2.120\times10^7$ | $2.111\times10^5$ |
| | 2 | 6933.16 | 6900.65 | 14.16 | 6954.06 | 6921.64 | 11.97 | 6948.04 | 6919.42 | 12.33 |
| | 7 | 71379.20 | 69738.05 | 1184.51 | 71079.73 | 69680.55 | 704.88 | 70984.70 | 69822.71 | 671.66 |
| | 6 | 57984.77 | 54406.39 | 1868.53 | 57976.49 | 54641.26 | 1821.25 | 57846.79 | 54343.18 | 1727.60 |
| VRP | 6 | 86276.57 | 84650.78 | 1027.23 | 84219.72 | 79515.17 | 2584.29 | 67553.28 | 63319.71 | 1725.80 |
| | 2 | 13444.56 | 13425.25 | 7.34 | 12315.08 | 12292.65 | 21.83 | 13285.96 | 12330.76 | 247.88 |
| | 5 | 272648.03 | 262320.73 | 4370.68 | 187362.40 | 167350.31 | 8805.09 | 167487.06 | 163724.28 | 2176.62 |
| | 1 | 20667.89 | 20661.99 | 3.06 | 20655.87 | 20652.63 | 1.08 | 20678.50 | 20665.04 | 6.86 |
| | 9 | 194763.76 | 191072.01 | 1618.57 | 147329.18 | 145771.29 | 665.82 | 150292.27 | 147740.46 | 1155.02 |

[2] G. Ochoa and M. Hyde. The cross-domain heuristic search challenge (chesc 2011). [Online]. Available: http://www.asap.cs.nott.ac.uk/chesc2011/

[3] E. Özcan, B. Bilgin, and E. E. Korkmaz, "Hill climbers and mutational heuristics in hyperheuristics," in *Parallel Problem Solving from Nature - PPSN IX*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2006, vol. 4193, pp. 202–211.

[4] 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.

[5] L. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338 – 353, 1965.

[6] 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. Springer Berlin Heidelberg, 2000, vol. 1917, pp. 335–344.

[7] 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 vol. 1.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] W. G. Jackson, E. Özcan, and R. I. John, "Fuzzy adaptive parameter control of a late acceptance hyper-heuristic," in *Computational Intelligence (UKCI), 2014 14th UK Workshop on*, Sept 2014, pp. 1–8.

[13] K. Sörensen and F. W. Glover, "Metaheuristics," in *Encyclopedia of Operations Research and Management Science*, S. Gass and M. C. Fu, Eds. Springer US, 2013, pp. 960–970.

[14] G. Dueck and T. Scheuer, "Threshold accepting: A general purpose optimization algorithm appearing superior to simulated annealing," *Journal of Computational Physics*, vol. 90, no. 1, pp. 161 – 175, 1990.

[15] G. Dueck, "New optimization heuristics: The great deluge algorithm and the record-to-record travel," *Journal of Computational Physics*, vol. 104, no. 1, pp. 86 – 92, 1993.

[16] V. K. Mısır M, Wauters T and B. G. V, "A new learning hyper-heuristic for the traveling tournament problem," in *Proceedings of the 8th Metaheuristic International Conference (MIC09)*, 2009.

[17] A. Kheiri and E. Özcan, "An iterated multi-stage selection hyper-heuristic," *European Journal of Operational Research*, vol. 250, no. 1, pp. 77 – 90, 2016.

[18] G. Ochoa, M. Hyde, T. Curtois, J. A. Vazquez-Rodriguez, J. Walker, M. Gendreau, G. Kendall, B. McCollum, A. J. Parkes, S. Petrovic, and E. K. Burke, "Hyflex: A benchmark framework for cross-domain heuristic search," in *Evolutionary Computation in Combinatorial Optimization*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, vol. 7245, pp. 136–147.

[19] J. Rada-Vilela, "fuzzylite: a fuzzy logic control library," 2014. [Online]. Available: http://www.fuzzylite.com

[20] D. Teodorovic and S. Kikuchi, "Application of fuzzy sets theory to the saving based vehicle routing algorithm," *Civil Engineering Systems*, vol. 8, no. 2, pp. 87–93, 1991.