# Heuristic Space Diversity Management in a Meta-Hyperheuristic Framework

Jacomine Grobler, Andries P. Engelbrecht, Graham Kendall, and V.S.S. Yadavalli

***Abstract*—This paper introduces the concept of heuristic space diversity and investigates various strategies for the management of heuristic space diversity within the context of a meta-hyperheuristic algorithm. Evaluation on a diverse set of floating-point benchmark problems show that heuristic space diversity has a significant impact on hyperheuristic performance. The increasing heuristic space diversity strategies performed the best out of all strategies tested. Good performance was also demonstrated with respect to another popular multi-method algorithm and the best performing constituent algorithm.**

## I. Introduction

OVER the last decade, research into hyper-heuristics have made an increasing impact on how optimization problems are approached. In contrast to traditional single method optimization algorithms, which search through a space of decision variables for solutions, hyper-heuristics search through a heuristic space of available heuristics or heuristic components [1]. The idea is to either find an "optimal" selection of heuristics, or to construct a heuristic from available heuristic components, to address a specific problem at hand. A meta-hyper-heuristic can be defined as a hyperheuristic where the constituent or low level algorithms consist of metaheuristic algorithms.

Diversity management is another important concept that has received increasing attention recently. Traditionally, the ability of an optimization algorithm to balance exploration and exploitation has been shown to have a significant impact on its performance. If the algorithm converges too quickly, it is more likely to become stuck in a local optimum. If the algorithm focuses too much on exploring new areas of the search space near the end of the optimization run, time is wasted on exploring the search space which could have been used to further refine promising solutions.

Based on the importance of effective management of solution space diversity in traditional optimization algorithms, it is not a major stretch to think that the diversity of the heuristic space and how it is managed throughout the optimization run, could have an important impact on hyperheuristic performance.

This paper proposes a measurement for quantitatively defining heuristic space diversity (HSD). A number of strategies for managing HSD are also proposed. Algorithm performance was evaluated on a set of varied floating-point benchmark problems and the most promising results were obtained by the hyperheuristics utilizing an increasing HSD strategy. Good performance was also shown against the population based algorithm portfolio algorithm [2], which is a well known multi-method algorithm, and the covariance matrix adapting evolutionary strategy algorithm (CMAES) [3], the best performing constituent algorithm.

To the best of the authors' knowledge, this is the first paper that explicitly introduces the concept of heuristic space diversity and the control of HSD to influence algorithm performance in a meta-hyperheuristic framework.

The rest of the paper is organized as follows: Section II provides some background with regards to diversity management. Section III provides a brief overview of the HMHH algorithm used as basis for the investigation, while Section IV describes the HSD control strategies which are evaluated. The results are documented in Section V before the paper is concluded in Section VI.

## II. Diversity management in multi-method algorithms

Although diversity management is not a new concept and is actually relatively common in single method literature, its use in the multi-method algorithm world is relatively limited. Furthermore, if diversity management is considered at all, the focus is mostly on managing solution space diversity (SSD) and not heuristic space diversity. The rest of this section gives a brief overview of diversity management in multi-method algorithms and also discusses a number of related issues.

Examples of controlling SSD to influence hyperheuristic performance includes Vrugt *et al.*'s AMALGAM [4], Grobler *et al.*'s investigation into the use of local search in a meta-hyperheuristic framework [5], and Grobler *et al.*'s adaptive local search algorithm [6]. AMALGAM makes use of a species selection mechanism to maintain SSD. In [6] various solution space diversity control strategies based on both adaptive and constant local search and AMALGAM's species selection mechanism were used to evaluate the impact of different SSD profiles on algorithm performance. Both papers showed that multi-method algorithm performance improvements can be obtained by managing SSD effectively.

The use of local search in hyperheuristics is so closely related to SSD that it is also worth mentioning here. Qu and Burke's graph based hyperheuristic framework [7] uses a local search algorithm to operate directly on the solution space in conjunction with a hyperheuristic strategy which operates in heuristic space. Local search algorithms can

Jacomine Grobler (email: jacomine.grobler@gmail.com) and V.S.S. Yadavalli are with the Department of Industrial and Systems Engineering and Andries P. Engelbrecht is with the Department of Computer Science, University of Pretoria, Pretoria, South Africa. Graham Kendall is with the School of Computer Science, University of Nottingham, UK and University of Nottingham Malaysia Campus.

also be incorporated into the set of available low-level heuristics [8]. This option can be considered an intervention in heuristic space diversity, especially when metaheuristics are utilized as low-level heuristics, since a more diverse set of algorithms are made available to the high-level strategy.

More closely related to heuristic space diversity is the issue of selecting complementing constituent algorithms. Peng *et al.* [2] proposed a pairwise metric which can be used to determine the risk associated with an algorithm failing to solve the problem in question. Engelbrecht [9] selected complementary swarm behaviours in a heterogeneous PSO by analyzing the exploration-exploitation finger prints of the different PSO updates.

It is clear that a number of authors have considered SSD management and algorithm selection to ensure complementary diverse algorithms. However, to the best of the authors' knowledge, this paper is the first to actively try to influence HSD to improve hyperheuristic performance.

## III. The heterogeneous meta-hyperheuristic algorithm

Due to its excellent performance against other popular multi-method algorithms, the tabu-search based HMHH algorithm of [10] was used as basis for investigating the management of heuristic space diversity. The various algorithmic elements of the HMHH algorithm, including a common population of entities each representing a candidate solution which is evolved over time, a set of constituent algorithms, an acceptance strategy, and a selection strategy is indicated in Figure 1.

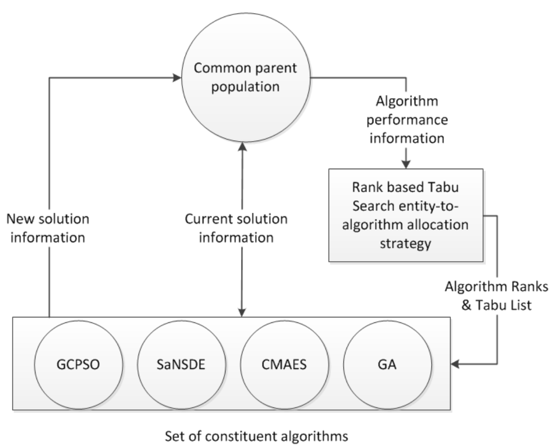


Fig. 1. The heterogeneous meta-hyperheuristic.

The HMHH algorithm divides the population of entities into a number of subpopulations which are evolved in parallel by a set of constituent algorithms. Each entity is able to access the genetic material of other subpopulations, as if part of a common population of entities. The allocation of entities to constituent algorithms is updated on a dynamic basis throughout the optimization run. The idea is that an intelligent algorithm can be evolved which selects the appropriate constituent algorithm at each $k^{th}$ iteration to be applied to each entity within the context of the common parent population, to ensure that the population of entities converge to a high quality solution. The constituent algorithm allocation is maintained for $k$ iterations, while the common parent population is continuously updated with new information and better solutions. Throughout this process, the various constituent algorithms are ranked based on their previous performance as defined by $Q_{\delta m}(t)$ in Algorithm 1. More specifically,

$$Q_{\delta m}(t) = \sum_{i=1}^{|\boldsymbol{I}_m(t)|} \left(f(\boldsymbol{x}_i(t-k)) - f(\boldsymbol{x}_i(t))\right) \quad \forall i \in \boldsymbol{I}_m(t) \tag{1}$$

where $f(\boldsymbol{x}_i(t)$ denotes the fitness function value of entity $i$ at time $t$ and $\boldsymbol{I}_m(t)$ is the set of entities allocated to algorithm $m$ at time $t$. A tabu list is used to prevent the algorithm from repeatedly using the same poorly performing constituent algorithms. The highest ranking non-tabu operator is then selected for each entity during re-allocation of entities to algorithms as described in [11].

This algorithm uses four common meta-heuristic algorithms as the set of constituent algorithms:

- A genetic algorithm (GA) with a floating-point representation, tournament selection, blend crossover [12], [13], and self-adaptive Gaussian mutation [14].
- The guaranteed convergence particle swarm optimization algorithm (GCPSO) [15].
- The self-adaptive (SaNSDE) algorithm of [16].
- The covariance matrix adapting evolutionary strategy algorithm (CMAES) [3].

## IV. Investigating alternative heuristic space diversity management strategies

The concept of heuristic space diversity is best illustrated by means of an example. In Figure 2 the entities in the population to the left were divided relatively equally between all of the available constituent algorithms during entity to algorithm allocation. This population can be described as having a high HSD. On the other hand, most of the entities in the population of the right were allocated to the genetic algorithm with only one entity each allocated to PSO and ES. This population can be described as having a low HSD.

A more quantitative metric for heuristic space diversity, $D_h(t)$, the heuristic space diversity at time $t$, can be defined as follows:

$$D_h(t) = UB_{D_h(t)} \left(1 - \frac{\sum_{i=1}^{I} |T - n_i(t)|}{1.5 n_s}\right) \tag{2}$$

with

$$T = \frac{n_s}{n_a}, \tag{3}$$

where $n_a$ is the number of algorithms available for selection by the hyper-heuristics, $n_s$ is the number of entities in

**Algorithm 1:** The heterogeneous meta-hyper-heuristic.

Initialize the parent population $\boldsymbol{X}$
$A_i(t)$ denotes the algorithm applied to entity $i$ at iteration $t$
**for** *All entities* $i \in \boldsymbol{X}$ **do**
  Randomly select an initial algorithm $A_i(1)$ from the set of constituent algorithms to apply to entity $i$
**end**
$t = 0$
$k = 5$
**while** *A stopping condition is not met* **do**
  **for** *All entities* $i$ **do**
    Apply constituent algorithm $A_i(t)$ to entity $i$ for $k$ iterations
    $t = t + k$
    Calculate $Q_{\delta m}(t)$, the total improvement in fitness function value of all entities assigned to algorithm $m$ from iteration $t - k$ to iteration $t$.
  **end**
  **for** *All entities* $i$ **do**
    Use $Q_{\delta m}(t)$ as input to select constituent algorithm $A_i(t)$ according to the rank based tabu search mechanism described in [11]
  **end**
**end**

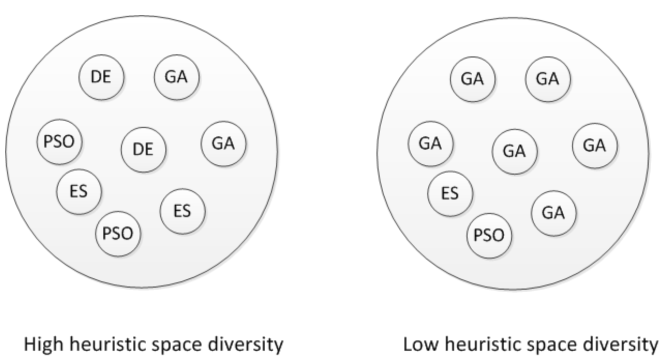


Fig. 2. An example of a population with a high HSD and a population with a low HSD.

the population, $n_i(t)$ is the number of entities allocated to algorithm $i$ at time $t$, and $UB_{D_h(t)}$ is the upper bound of the HSD measure. For the purposes of this paper, $UB_{D_h(t)}$ was set to 100 so that $D_h(t) \in [0, 100]$.

Five strategies for controlling HSD throughout the optimization process are explored in this paper:

- **The baseline HMHH algorithm** - This algorithm is the standard HMHH algorithm implemented as described in Section III. No effort is made to manipulate the HSD in this algorithm.
- **Linearly decreasing HSD hyperheuristic (LDHH)** - This algorithm is characterized by a linearly decreasing HSD. At the start of the optimization run all four constituent algorithms are available for selection. During the optimization run, the worst performing constituent algorithm is removed from the set of available algorithms at predefined constant time intervals. As an example, if the maximum allowable function evaluations are 100,000, the worst performing algorithm at that time will be removed respectively at 25,000, 50,000 and 75,000 function evaluations. The idea is to force the hyperheuristic to explore the heuristic space at the start of the optimization run and exploit the best performing algorithm towards the end of the optimization run.
- **Exponentially decreasing HSD hyperheuristic (EDHH)** - This algorithm is characterized by an exponentially decreasing HSD. All constituent algorithms are again available for allocation to entities at the start of the optimization run and algorithms are again removed according to their performance at predetermined time intervals. This time, however, the algorithms are removed at exponential time intervals. The result is a slower changeover from exploration to exploitation.
- **Linearly increasing HSD hyperheuristic (LIHH)** - This algorithm assumes apriori knowledge of the constituent algorithm performance on the benchmark problem set being solved. The constituent algorithms are ranked from best performing to worst performing. Only the best performing algorithm is made available to the HH at the start of the optimization run. As the optimization process progresses, additional algorithms are made available according to their ranking at predetermined constant time intervals. Here the hyperheuristic is forced to move from exploitation to exploration. The idea is to obtain maximum gain from the highest ranked algorithm and as the performance gains decrease, the rest of the constituent algorithms become available to diversify the heuristic space and improve the overall algorithm performance.
- **Exponentially increasing HSD hyperheuristic (EIHH)** - This algorithm is similar to the LIHH algorithm, the only difference being that exponential time intervals are used to add algorithms to the set of available algorithms. The use of exponential time intervals increases the rate of change of HSD leading to a faster changeover from exploitation to exploration.

## V. Empirical evaluation

The various HSD control strategies were evaluated on the first 14 problems of the 2005 IEEE Congress of Evolutionary Computation benchmark problem set [17] in both 10 and 30 dimensions. This benchmark problem set enables algorithm performance evaluation on both unimodal and multimodal functions and includes various expanded and hybridized problems, some with noisy fitness functions. The algorithm control parameter values listed in Table I were found to work well for the algorithms under study during previous research by the authors. $m \longrightarrow n$ indicates that the associated parameter is decreased linearly from $m$ to $n$ over 95% of the maximum number of iterations, $I_{max}$.

TABLE I
HMHH ALGORITHM PARAMETERS.

| **Parameter** | **Value used** |
|---|---|
| Number of entities in common population ($n_s$) | 100 |
| Number of iterations between re-allocation ($k$) | 5 |
| Size of tabu list ($n_a = 4$) | 2 |
| Size of tabu list ($n_a = 3$) | 1 |
| Size of tabu list ($n_a \leq 2$) | 0 |
| **PSO parameters** | |
| Acceleration constant ($c_1$) | $2.0 \longrightarrow 0.7$ |
| Acceleration constant ($c_2$) | $0.7 \longrightarrow 2.0$ |
| Inertia weight ($w$) | $0.9 \longrightarrow 0.4$ |
| **SaNSDE parameters** | As specified in [16]. |
| **GA parameters** | |
| Probability of crossover ($p_c$) | $0.6 \longrightarrow 0.4$ |
| Probability of mutation ($p_m$) | 0.1 |
| Blend crossover parameter ($\alpha$) | 0.5 |
| GA tournament size ($N_t$) | 13 |
| **CMAES parameters** | As specified in [3]. |

The results of the heuristic space diversity management technique comparison are presented in Table IV, where the results for each algorithm were recorded over 30 independent simulation runs. $\mu$ and $\sigma$ denote the mean and standard deviation associated with the corresponding performance measure and #$FEs$ denotes the number of function evaluations which were needed to reach the global optimum within a specified accuracy. If the global optimum was reached within the specified accuracy, the run was stopped and the difference between the global optimum and the final fitness function obtained, denoted by $FFV$, was recorded. Where the global optimum could not be found within the maximum number of iterations, the difference between the final solution at $I_{max}$ and the global optimum, also denoted by $FFV$, was recorded.

Mann-Whitney U tests were used to evaluate the various strategies according to the number of iterations required to obtain the final fitness function value, as well as the quality of the actual fitness function value. Statistical tests were also used to evaluate the significance of the results. The results in Table II were obtained by comparing each dimension-problem-combination of the strategy under evaluation, to all of the dimension-problem-combinations of the other strategies. For every comparison, a Mann-Whitney U test at 95% significance was performed (using the two sets of 30 data points of the two strategies under comparison) and if the first strategy statistically significantly outperformed the second strategy, a win was recorded. If no statistical difference could be observed a draw was recorded. If the second strategy outperformed the first strategy, a loss was recorded for the first strategy. The total number of wins, draws and losses were then recorded for all combinations of the strategy under evaluation. To illustrate, (11-15-2) in row 1 column 2, indicates that the LDHH strategy outperformed the baseline HMHH algorithm 11 times over the benchmark problem set. Fifteen draws and two losses were recorded.

TABLE II
HYPOTHESES ANALYSIS OF ALTERNATIVE HEURISTIC SPACE DIVERSITY CONTROL MECHANISMS.

| **Algorithm** | **HMHH** |
|---|---|
| **LDHH** | $11-15-2$ |
| **EDHH** | $11-13-4$ |
| **LIHH** | $19-5-4$ |
| **EIHH** | $19-8-1$ |
| **TOTAL** | $60-41-11$ |

From the results it is clear that managing the HSD leads to statistically significantly improved hyperheuristic performance. Table II shows that for 101 cases out of 112, the strategies where the HSD was controlled performed statistically similar or better than the baseline HMHH algorithm. Interestingly the hyperheuristic's performance was relatively insensitive to the rate of change of diversity. Finally, the increasing HSD strategies outperformed the decreasing HSD strategies quite significantly. It is suspected that the apriori constituent algorithm performance knowledge is largely responsible for these good results. Unfortunately, this knowledge is not always readily available and then an alternative strategy, for example, randomly selecting the sequence of algorithms to be made available, needs to be used.

In an attempt to further verify the performance of the HSD management strategies, the two best performing solution diversity management strategy from the previous analysis, LDHH and EIHH, were also compared under similar conditions to PAP [2], which was found in a previous study [18] to be the best performing multi-method algorithm currently available (after the HMHH algorithm). Finally, the best performing constituent algorithm (CMAES) [3], was also added for comparison purposes. The results are recorded in Table V. In Table III Mann-Whitney U tests were used to compare the performance of PAP and CMAES to the HSD management strategies. This time each HSD control strategy was compared to both CMAES and PAP and the "number of wins-draws-losses" were obtained.

TABLE III
FURTHER HYPOTHESES ANALYSIS OF THE BEST PERFORMING HSD CONTROL MECHANISMS.

| | **PAP** | **CMAES** |
|---|---|---|
| **LDHH** | $10-11-7$ | $7-4-17$ |
| **EIHH** | $21-7-0$ | $9-17-2$ |

From Table III it can be seen that the HMHH algorithms outperforms PAP in a large number of cases. LDHH performs similar or better to PAP 75% of the time and EIHH performs similar or better 100% of the time. LDHH does not perform quite as well when compared to CMAES. This can, however, be expected since a portion of the function evaluation budget needs to be allocated to solve the algorithm selection problem. This is in contrast to CMAES which can use the whole function evaluation budget on optimization

TABLE IV
COMPARISON RESULTS OF THE DIFFERENT HSD MANAGEMENT STRATEGIES ON THE 2005 IEEE CEC BENCHMARK PROBLEM SET.

| Prob | HMHH | | | | LDHH | | | | EDHH | | | | LIHH | | | | EIHH | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| (Dims) | FFV | | # FEs | | FFV | | # FEs | | FFV | | # FEs | | FFV | | # FEs | | FFV | | # FEs | |
| | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ |
| 1(10) | $1E-06$ | 0 | 13310 | 694.98 | $1E-06$ | 0 | 12017 | 642.78 | $1E-06$ | 0 | 12090 | 579.15 | $1E-06$ | 0 | 8623.3 | 244.5 | $1E-06$ | 0 | 8563.3 | 277.28 |
| 1(30) | $1E-06$ | 0 | 45040 | 1783.8 | $1E-06$ | 0 | 43653 | 1500.7 | $1E-06$ | 0 | 46723 | 2038.7 | $1E-06$ | 0 | 19253 | 599.27 | $1E-06$ | 0 | 19193 | 549.57 |
| 2(10) | $1E-06$ | 0 | 31577 | 2659 | $1E-06$ | 0 | 13743 | 959.41 | $1E-06$ | 0 | 14393 | 902.84 | $1E-06$ | 0 | 9173.3 | 294.7 | $1E-06$ | 0 | 9106.7 | 398.21 |
| 2(30) | $1E-06$ | 0 | $2.204E+05$ | 49695 | $1E-06$ | 0 | 97973 | 14510 | $1E-06$ | 0 | $2.018E+05$ | 95248 | $1E-06$ | 0 | 26760 | 766.36 | $1E-06$ | 0 | 26410 | 806.59 |
| 3(10) | 238.87 | 910.04 | 84027 | 11776 | $1E-06$ | 0 | 20120 | 2020.8 | $1E-06$ | 0 | 19550 | 1444.1 | $1E-06$ | 0 | 13350 | 483.34 | $1E-06$ | 0 | 13277 | 397.13 |
| 3(30) | $1.352E+05$ | 73374 | $3E+05$ | 0 | 55462 | 48441 | $2.970E+05$ | 16395 | 70567 | 75788 | $3E+05$ | 0 | $1E-06$ | 0 | 61567 | 1265.3 | $1E-06$ | 0 | 61603 | 1201.9 |
| 4(10) | $1E-06$ | 0 | 43940 | 9759.9 | $1E-06$ | 0 | 15550 | 1344.9 | $1E-06$ | 0 | 16003 | 1023.7 | $1E-06$ | 0 | 9510 | 347.75 | $1E-06$ | 0 | 9573.3 | 374.1 |
| 4(30) | 0.945 | 2.854 | $3E+05$ | 0 | 0.003 | 0.0183 | $1.502E+05$ | $1.017E+05$ | 1.147 | 3.781 | $1.641E+05$ | $1E+05$ | $1E-06$ | 0 | 29330 | 927.79 | $1E-06$ | 0 | 29267 | 986.58 |
| 5(10) | $1E-06$ | 0 | 17980 | 1572 | $1E-06$ | 0 | 17313 | 1044.1 | $1E-06$ | 0 | 17077 | 1054 | $1E-06$ | 0 | 17520 | 608.79 | $1E-06$ | 0 | 17447 | 567.35 |
| 5(30) | 1360.6 | 728.2 | $3E+05$ | 0 | 1086.8 | 849.77 | $3E+05$ | 0 | 1200.2 | 662.45 | $3E+05$ | 0 | $1E-06$ | 0 | $2.572E+05$ | 68420 | 0.000 | 0.002 | $2.720E+05$ | 70931 |
| 6(10) | 0.119 | 0.418 | 65910 | 19543 | 0.265 | 1.010 | 37347 | 18132 | 0.016 | 0.084 | 35933 | 13900 | 0.001 | 0.003 | 18940 | 744.91 | 0.000 | 0.002 | 19210 | 716 |
| 6(30) | 4.275 | 22.125 | $2.510E+05$ | 45175 | 1.165 | 1.906 | $2.394E+05$ | 54813 | 1.076 | 1.730 | $2.759E+05$ | 41820 | 0.133 | 0.727 | $1.229E+05$ | 46644 | 0 | 0 | $2.087E+05$ | 49555 |
| 7(10) | 0.445 | 0.342 | $1E+05$ | 0 | 0.155 | 0.127 | $1E+05$ | 0 | 0.106 | 0.093 | 93167 | 17726 | 0.001 | 0.003 | 7533.3 | 339.71 | 0.001 | 0.003 | 7640 | 439.91 |
| 7(30) | 0.005 | 0.008 | $1.615E+05$ | $1.153e+05$ | 0.004 | 0.010 | $1.495E+05$ | $1.166E+05$ | 0.007 | 0.016 | $1.470E+05$ | $1.187E+05$ | 0.000 | 0.002 | 16603 | 673.38 | 0 | 0 | 16657 | 622.94 |
| 8(10) | 20.061 | 0.106 | $1E+05$ | 0 | 20.034 | 0.077 | 99290 | 3888.8 | 20.114 | 0.166 | $1E+05$ | 0 | 20.07 | 0.114 | $1E+05$ | 0 | 20.054 | 0.097 | $1E+05$ | 0 |
| 8(30) | 20.166 | 0.110 | $3E+05$ | 0 | 20.15 | 0.130 | $3E+05$ | 0 | 20.239 | 0.197 | $3E+05$ | 0 | 20.209 | 0.158 | $3E+05$ | 0 | 20.122 | 0.106 | $3E+05$ | 0 |
| 9(10) | 0.038 | 0.178 | 38270 | 19385 | 0.005 | 0.005 | 33440 | 15441 | 0.529 | 0.888 | 54163 | 35998 | 0.495 | 0.766 | 79563 | 30822 | 0.233 | 0.419 | 58827 | 27886 |
| 9(30) | 2.245 | 1.806 | $2.896E+05$ | 30956 | 9.861 | 5.367 | $2.973E+05$ | 14953 | 18.916 | 8.090 | $3E+05$ | 0 | 7.852 | 4.594 | $3E+05$ | 0 | 3.455 | 3.710 | $2.758E+05$ | 42827 |
| 10(10) | 16.157 | 7.026 | $1E+05$ | 0 | 19.136 | 9.445 | $1E+05$ | 0 | 15.582 | 9.623 | 98393 | 8800.1 | 1.6145 | 1.091 | 88090 | 30887 | 1.879 | 1.205 | 88200 | 30602 |
| 10(30) | 72.135 | 30.203 | $3E+05$ | 0 | 73.098 | 31.88 | $3E+05$ | 0 | 73.336 | 28.963 | $3E+05$ | 0 | 12.719 | 5.454 | $3E+05$ | 0 | 10.553 | 4.577 | $3E+05$ | 0 |
| 11(10) | 6.876 | 1.702 | $1E+05$ | 0 | 6.914 | 1.313 | $1E+05$ | 0 | 6.112 | 2.0343 | 98277 | 9439.1 | 1.223 | 1.279 | 74537 | 39566 | 1.021 | 1.149 | 66540 | 41709 |
| 11(30) | 27.311 | 4.447 | $3E+05$ | 0 | 21.74 | 7.740 | $3E+05$ | 0 | 19.711 | 5.236 | $3E+05$ | 0 | 8.902 | 3.282 | $3E+05$ | 0 | 10.785 | 6.415 | $3E+05$ | 0 |
| 12(10) | 466.17 | 620.43 | 88740 | 19820 | 156.24 | 420.61 | 69417 | 35661 | 131.04 | 401.7 | 60357 | 37188 | 174.03 | 416.19 | 72990 | 41976 | 317.66 | 627.11 | 63770 | 45137 |
| 12(30) | 4489.4 | 5300.9 | $3E+05$ | 0 | 4646.4 | 4889 | $3E+05$ | 0 | 10348 | 13200 | $3E+05$ | 0 | 5850.1 | 3788.7 | $2.912E+05$ | 48127 | 4822.2 | 4782.6 | $3E+05$ | 0 |
| 13(10) | 0.502 | 0.230 | $1E+05$ | 0 | 0.547 | 0.194 | $1E+05$ | 0 | 0.585 | 0.227 | $1E+05$ | 0 | 0.643 | 0.220 | $1E+05$ | 0 | 0.442 | 0.143 | $1E+05$ | 0 |
| 13(30) | 1.873 | 0.441 | $3E+05$ | 0 | 2.794 | 0.879 | $3E+05$ | 0 | 2.256 | 0.632 | $3E+05$ | 0 | 2.415 | 0.677 | $3E+05$ | 0 | 1.836 | 0.555 | $3E+05$ | 0 |
| 14(10) | 3.653 | 0.285 | $1E+05$ | 0 | 3.457 | 0.545 | $1E+05$ | 0 | 3.511 | 0.425 | $1E+05$ | 0 | 2.826 | 0.643 | $1E+05$ | 0 | 2.695 | 0.379 | $1E+05$ | 0 |
| 14(30) | 13.14 | 0.443 | $3E+05$ | 0 | 13.204 | 0.318 | $3E+05$ | 0 | 13.185 | 0.381 | $3E+05$ | 0 | 10.059 | 0.823 | $3E+05$ | 0 | 10.288 | 0.965 | $3E+05$ | 0 |

TABLE V

COMPARISON RESULTS OF THE BEST PERFORMING HSD MANAGEMENT STRATEGIES VERSUS THE BEST PERFORMING CONSTITUENT ALGORITHM AND MULTI-METHOD ALGORITHM ON THE 2005 IEEE CEC BENCHMARK PROBLEM SET.

| Prob (Dims) | LDHH | | | | EIHH | | | | PAP | | | | CMAES | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | FFV | | # FEs | | FFV | | # FEs | | FFV | | # FEs | | FFV | | # FEs | |
| | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ |
| 1(10) | $1E-06$ | 0 | 12017 | 642.78 | $1E-06$ | 0 | 8563,3 | 277.28 | $1E-06$ | 0 | 13857 | 630.64 | $1E-06$ | 0 | 8526,7 | 302.78 |
| 1(30) | $1E-06$ | 0 | 43653 | 1500.7 | $1E-06$ | 0 | 19193 | 549.57 | $1E-06$ | 0 | 39190 | 5945.1 | $1E-06$ | 0 | 19110 | 447.48 |
| 2(10) | $1E-06$ | 0 | 13743 | 959.41 | $1E-06$ | 0 | 9106,7 | 398.21 | $1E-06$ | 0 | 18760 | 1030.4 | $1E-06$ | 0 | 9156,7 | 286.1 |
| 2(30) | $1E-06$ | 0 | 97973 | 14510 | $1E-06$ | 0 | 26410 | 806.59 | $1E-06$ | 0 | 90063 | 2729.3 | $1E-06$ | 0 | 26783 | 739.1 |
| 3(10) | $1E-06$ | 0 | 20120 | 2020.8 | $1E-06$ | 0 | 13277 | 397.13 | $1E-06$ | 0 | 46067 | 2040.3 | $1E-06$ | 0 | 13320 | 379.11 |
| 3(30) | 55462 | 48441 | $2,97E+05$ | 16395 | $1E-06$ | 0 | 61603 | 1201.9 | $1E-06$ | 0 | $2,88E+05$ | 5481.5 | $1E-06$ | 0 | 61173 | 1387.4 |
| 4(10) | $1E-06$ | 0 | 15550 | 1344.9 | $1E-06$ | 0 | 9573,3 | 374.1 | $1E-06$ | 0 | 20483 | 1424.7 | $1E-06$ | 0 | 9590 | 283.27 |
| 4(30) | 0.003 | 0.018 | $1,50E+05$ | $1,02E+05$ | $1E-06$ | 0 | 29267 | 986.58 | 4448.8 | 5562.2 | $2,87E+05$ | 39164 | $1E-06$ | 0 | 29357 | 570.35 |
| 5(10) | $1E-06$ | 0 | 17313 | 1044.1 | $1E-06$ | 0 | 17447 | 567.35 | $1E-06$ | 0 | 25010 | 2790.5 | $1E-06$ | 0 | 17433 | 546.04 |
| 5(30) | 1086.8 | 849.77 | $3E+05$ | 0 | 0.000 | 0.002 | $2,72E+05$ | 70931 | 1115.5 | 1446.5 | $3E+05$ | 0 | $1E-06$ | 0 | $1,15E+05$ | 3960.1 |
| 6(10) | 0.265 | 1.010 | 37347 | 18132 | 0.000 | 0.002 | 19210 | 716 | 0.133 | 0.727 | 50613 | 10870 | 0.001 | 0.003 | 18950 | 744.52 |
| 6(30) | 1.165 | 1.906 | $2,39E+05$ | 54813 | 0 | 0 | $2,09E+05$ | 49555 | 0.519 | 1.346 | $2,75E+05$ | 27166 | 0.133 | 0.727 | $1,20E+05$ | 37870 |
| 7(10) | 0.155 | 0.127 | $1E+05$ | 0 | 0.001 | 0.003 | 7640 | 439.91 | 0.003 | 0.006 | 62867 | 35481 | 1267 | $4,63E-13$ | $1E+05$ | 0 |
| 7(30) | 0.004 | 0.010 | $1,50E+05$ | $1,17E+05$ | 0 | 0 | 16657 | 622.94 | 0 | 0 | 82047 | 41811 | 4696.3 | $2,78E-12$ | $3E+05$ | 0 |
| 8(10) | 20.034 | 0.077 | 99290 | 3888.8 | 20.054 | 0.097 | $1E+05$ | 0 | 20.058 | 0.086 | $1E+05$ | 0 | 20.312 | 0.113 | $1E+05$ | 0 |
| 8(30) | 20.15 | 0.130 | $3E+05$ | 0 | 20.122 | 0.106 | $3E+05$ | 0 | 20.264 | 0.163 | $3E+05$ | 0 | 20.892 | 0.175 | $3E+05$ | 0 |
| 9(10) | 0.005 | 0.005 | 33440 | 15441 | 0.233 | 0.419 | 58827 | 27886 | 0.220 | 0.401 | 68493 | 26929 | 1.946 | 1.511 | 88203 | 30601 |
| 9(30) | 9.861 | 5.367 | $2,97E+05$ | 14953 | 3.455 | 3.710 | $2,76E+05$ | 42827 | 2.676 | 1.92 | $2,81E+05$ | 55032 | 39.564 | 6.454 | $3E+05$ | 0 |
| 10(10) | 19.136 | 9.445 | $1E+05$ | 0 | 1.8793 | 1.205 | 88200 | 30602 | 12.857 | 6.265 | $1E+05$ | 0 | 1.647 | 1.177 | 90947 | 27625 |
| 10(30) | 73.098 | 31.88 | $3E+05$ | 0 | 10.553 | 4.577 | $3E+05$ | 0 | 70.555 | 19.764 | $3E+05$ | 0 | 9.391 | 3.282 | $3E+05$ | 0 |
| 11(10) | 6.914 | 1.313 | $1E+05$ | 0 | 1.021 | 1.149 | 66540 | 41709 | 4.085 | 1.262 | $1E+05$ | 0 | 1.299 | 1.365 | 30310 | 10496 |
| 11(30) | 21.74 | 7.740 | $3E+05$ | 0 | 10.785 | 6.415 | $3E+05$ | 0 | 20.034 | 2.645 | $3E+05$ | 0 | 9.026 | 3.055 | $3E+05$ | 0 |
| 12(10) | 156.24 | 420.61 | 69417 | 35661 | 317.66 | 627.11 | 63770 | 45137 | 231.53 | 539.16 | 71737 | 28064 | 1546.1 | 2735.5 | 70053 | 43090 |
| 12(30) | 4646.4 | 4889 | $3E+05$ | 0 | 4822.2 | 4782.6 | $3E+05$ | 0 | 3573.5 | 3310.2 | $3E+05$ | 0 | 20324 | 19261 | $3E+05$ | 0 |
| 13(10) | 0.547 | 0.194 | $1E+05$ | 0 | 0.442 | 0.143 | $1E+05$ | 0 | 0.498 | 0.159 | $1E+05$ | 0 | 0.897 | 0.253 | $1E+05$ | 0 |
| 13(30) | 2.794 | 0.879 | $3E+05$ | 0 | 1.836 | 0.555 | $3E+05$ | 0 | 1.74 | 0.421 | $3E+05$ | 0 | 3.179 | 0.561 | $3E+05$ | 0 |
| 14(10) | 3.457 | 0.545 | $1E+05$ | 0 | 2.695 | 0.379 | $1E+05$ | 0 | 3.407 | 0.317 | $1E+05$ | 0 | 2.585 | 0.530 | $1E+05$ | 0 |
| 14(30) | 13.204 | 0.318 | $3E+05$ | 0 | 10.288 | 0.965 | $3E+05$ | 0 | 13.026 | 0.319 | $3E+05$ | 0 | 10.394 | 0.810 | $3E+05$ | 0 |

of the actual problem. It is encouraging to note that IEHH significantly outperforms CMAES. Allocating part of the function evaluation budget to other algorithms later during the optimization run clearly has a positive impact on hyper-heuristic performance.

## VI. Conclusion

This paper defined the concept of heuristic space diversity and investigated the impact of different heuristic space diversity management strategies on multi-method optimization algorithm performance. The results indicated that a significant performance improvement can be obtained by controlling the HSD of the HMHH algorithm. The control strategies were found to be relatively insensitive to the rate of change of HSD and the increasing HSD strategies were shown to outperform the decreasing and uncontrolled HSD strategies. Finally, the best performing HSD control strategies were shown to perform well against a popular multi-method algorithm and the best performing constituent algorithm.

Future research opportunities exist in expanding the analysis to a larger set of benchmark problems and investigating the resulting HSD profiles of popular existing multi-method algorithms such as AMALGAM, PAP and other bandit-based approaches [19] and the subsequent impact these profiles have on algorithm performance. Furthermore, EIHH and LIHH can be modified to avoid the necessity of apriori knowledge about constituent algorithm performance on the problem being solved.

## References


[1] E. K. Burke, M. Gendreau, M. Hyde, G. Kendall, G. Ochoa, E. Ozcan, and R. Qu, "Hyper-heuristics: A survey of the state of the art," *Journal of the Operational Research Society*, vol. 64, pp. 1695–1724, 2013.

[2] F. Peng, K. Tang, G. Chen, and X. Yao,"Population-Based Algorithm Portfolios for Numerical Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 5, pp. 782–800, 2010.

[3] A. Auger, and N. Hansen, "A Restart CMA evolution strategy With increasing population size," *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, pp. 1769-1776, 2005.

[4] J. A. Vrugt, B. A. Robinson, and J. M. Hyman, "Self-Adaptive Multimethod Search for Global Optimization in Real-Parameter Spaces," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 243–259, 2009.

[5] J. Grobler, A. P. Engelbrecht, G. Kendall, and V. S. S. Yadavalli, "Investigating the Use of Local Search for Improving Meta-Hyper-Heuristic Performance," *Proceedings of the 2012 IEEE Congress on Evolutionary Computation*, pp. 1-8, 2012.

[6] J. Grobler and A. P. Engelbrecht, "Solution space diversity management in a meta-hyperheuristic framework," *Proceedings of the 2013 BRICS Congress on Computational Intelligence*, pp. 1-6, 2013.

[7] R. Qu and E. K. Burke, "Hybridisations within a graph based hyper-heuristic framework for university timetabling problems," *Journal of the Operational Research Society*, vol. 60, pp. 1273–1285, 2009.

[8] E. Ozcan, B. Bilgin, and E. E. Korkmaz, "A comprehensive survey of hyperheuristics," *Intelligent Data Analysis*, vol. 12, no. 1, pp. 1–21, 2008.

[9] A. P. Engelbrecht, "Scalability of a heterogeneous particle swarm optimizer," *Proceedings of the 2011 Symposium on Swarm Intelligence*, pp. 1–8, 2011.

[10] J. Grobler, A. P. Engelbrecht, G. Kendall, and V. S. S. Yadavalli, "Alternative hyper-heuristic strategies for multi-method global optimization," *Proceedings of the 2010 IEEE World Congress on Computational Intelligence*, pp. 826–833, 2010.

[11] E. K. Burke, G. Kendall, and E. Soubeiga, "A Tabu-Search Hyper-heuristic for Timetabling and Rostering," *Journal of Heuristics*, vol. 9, no. 6, pp. 451–470, 2003.

[12] L. J. Eshelman and J. D. Schaffer, "Real-coded genetic algorithms and interval schemata," In D. Whitley, editor, *Foundations of Genetic Algorithms*, vol. 2, pp. 187–202, 1993.

[13] O. Olorunda and A. P. Engelbrecht, "An Analysis of Heterogeneous Cooperative Algorithms," *Proceedings of the 2009 IEEE Congress on Evolutionary Computation*, pp. 1562–1569, 2009.

[14] J. Grobler, A. P. Engelbrecht, G. Kendall, and V. S. S. Yadavalli, "Investigating the impact of alternative evolutionary selection strategies on multi-method global optimization," *Proceedings of the 2011 IEEE Congress on Evolutionary Computation*, pp. 2337-2344, 2011.

[15] F. Van den Bergh and A. P. Engelbrecht, "A new locally convergent particle swarm optimiser," *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, vol. 3, pp. 6–12, 2002.

[16] Z. Yang, K. Tang, and X. Yao, "Self-adaptive Differential Evolution with Neighbourhood Search," *Proceedings of the 2008 IEEE Congress on Evolutionary Computation*, pp. 1110-1116, 2008.

[17] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y. P. Chen, A. Auger, S. Tiwari, "Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization," *Technical Report, Nanyang Technological University, Singapore and KanGAL Report Number 2005005 (Kanpur Genetic Algorithms Laboratory, IIT Kanpur)*, 2005.

[18] J. Grobler, A. P. Engelbrecht, G. Kendall, and V. S. S. Yadavalli, "Multi-Method Algorithms: Investigating the Entity-to-Algorithm Allocation Problem," *Proceedings of the 2013 IEEE Congress on Evolutionary Computation*, pp. 570-577, 2013.

[19] A. Fialho, M. Schoenauer, and M. Sebag, "Fitness-AUC bandit adaptive strategy selection vs. the probability matching one within differential evolution: an empirical comparison on the BBOB-2010 noiseless testbed," *Proceedings of the GECCO 2010 Workshop on Black-Box Optimization Benchmarking*, 2010.