Metric-based heuristic space diversity management in a meta-hyper-heuristic framework

Jacomine Grobler
Department of Industrial and Systems Engineering
University of Pretoria
Pretoria, South Africa
Email: jacomine.grobler@gmail.com

Andries P. Engelbrecht
Department of Computer Science
University of Pretoria
Pretoria, South Africa

***Abstract*—This paper investigates various strategies for the management of heuristic space diversity within the context of a meta-hyper-heuristic algorithm. In contrast to all previously developed heuristic space diversity management strategies, this paper makes use of a heuristic space diversity metric to monitor heuristic space diversity throughout the optimization run and trigger the need for increased or decreased heuristic space diversity. Three different heuristic space diversity management strategies are evaluated. Maintaining a high level of heuristic space diversity throughout the optimization run is shown to be the best performing strategy. Good performance is also demonstrated with respect to a state-of-the-art multi-method algorithm, another successful diversity controlling meta-hyper-heuristic and the best-performing constituent algorithm.**

I. Introduction

Effective management of diversity has been repeatedly shown to have a significant impact on the quality of solutions found by an optimization algorithm. Traditionally, management of the diversity of the solution or decision space prevents algorithms from converging too quickly to suboptimal solutions and also ensures effective exploration of a large part of the search space. However, with the increase in popularity of multi-method algorithms, the idea of diversity management has changed. Multi-method algorithms also have a heuristic space which needs to be searched effectively to ensure effective combinations of constituent algorithms at different stages of the optimization process. Recently, the concept of heuristic space diversity (HSD) was defined and it was shown that managing the diversity of the heuristic space through managing the set of available constituent algorithms, can have a significant effect on the performance of multi-method algorithms.

In this paper an adaptive HSD management strategy is proposed based on the HSD metric defined by Grobler *et al.* [1]. Three variations of the strategy are investigated and performance was evaluated on a set of varied floating-point benchmark problems. The most promising results were obtained by the HSD(1) strategy which attempts to maintain a pre-defined high level of HSD throughout the optimization run. In this strategy, HSD is increased by randomly selecting an entity and re-allocating the entity to a randomly selected constituent algorithm. Good performance was also obtained when the HSD(1) strategy was compared to a meta-hyper-heuristic algorithm which manages HSD through modification of the set of constituent algorithms (EIHH2) and the population based algorithm portfolio algorithm [2], which is a well known multi-method algorithm.

Effective control of the diversity of the heuristic space has already been considered in the past. However, to the best of the authors knowledge, all of these studies focus on the characteristics and modification of the set of available constituent algorithms. This algorithm is the first algorithm to make use of a HSD metric to self-adaptively determine when to adjust the allocation of entities-to-algorithms to control the HSD.

The rest of the paper is organized as follows: Section II provides an overview of existing literature. 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 were evaluated. The results are documented in Section V before the paper is concluded in Section VI.

II. Diversity management

The relationship between heuristic space and solution space was considered in detail for the first time in [3]. Since then, further efforts to analyze the heuristic space by means of landscape analysis were also conducted [4], [5]. From these studies it became evident that additional effort in managing the diversity of the solution space could result in important hyper-heuristic performance benefits.

Various examples of algorithms which attempt to either further exploit the solution space around good performing solutions [19], [21], [31], or improve the overall exploration ability of the hyper-heuristic by applying diversity management mechanisms directly to the solution space can be found in the hyper-heuristic literature [6]–[10].

Solution space diversity management is also an important consideration in a field closely related to hyper-heuristics, namely memetic computing [11]. Crepinsek *et al.* [12] provide a detailed review of diversity management in memetic computing and other fields. Notable examples of using solution space diversity to control the exploration-exploitation trade-off of memetic algorithms are the fitness-diversity adaptive local search algorithms [24].

978-1-4799-7560-0/15 $31.00 © 2015 IEEE
DOI 10.1109/SSCI.2015.234

More closely related to heuristic space diversity is the issue of selecting the set of low level heuristics. Montazeri *et al.* [14] ensured that their set of low level heuristics contains both exploiter heuristics, designed for intensification, and explorer heuristics, aimed at diversification. Yao *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 [15] selected complementary swarm behaviours in a heterogeneous particle swarm optimization (PSO) algorithm by analyzing the exploration-exploitation finger prints of the different PSO updates.

Recently, researchers have started to dynamically update the set of low level heuristics during the optimization run. Sim *et al.* [16] made use of a self-organizing network of low level heuristics to ensure that different heuristics were available to cover different areas of the search space. Random heuristics were added at fixed time intervals and an affinity measure related to the difference in performance between the different low level heuristics was used to determine when an under-performing heuristic should be removed. The evolutionary selection hyper-heuristic of Misir *et al.* [17] makes use of an adaptive dynamic heuristic set strategy, a move acceptance strategy, and a re-initialisation mechanism to manage the exploration-exploitation trade-off. The first explicit metric focused on measuring and monitoring HSD, was defined by Grobler *et al.* [1]. Various strategies for increasing and decreasing the size of the set of available constituent algorithms at predefined time intervals were also developed and showed promising results.

It is clear that a number of researchers have considered techniques to improve the exploration-exploitation trade off in a hyper-heuristics context. The selection of low level heuristics with regards to diversity management and the effective management of the set of low level heuristics over time have also been studied. However, to the best of the authors' knowledge, this paper is the first to attempt to manage HSD based on the performance of a specific HSD metric.

III. THE HETEROGENEOUS META-HYPER-HEURISTIC ALGORITHM

Due to its excellent performance against other popular multi-method algorithms, the tabu-search based HMHH algorithm of [22], illustrated in Figure 1, was used as a basis for investigating the management of HSD in this paper. The HMHH algorithm divides a 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 converges 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,

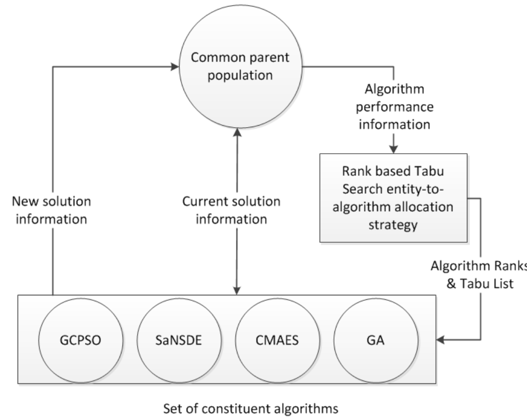

Fig. 1: The heterogeneous meta-hyper-heuristic (HMHH).

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

where $f(\boldsymbol{x}_i(t))$ denotes the fitness function value of entity i at iteration t and $\boldsymbol{I}_m(t)$ is the set of entities allocated to algorithm m at iteration 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 [13].

The HMHH 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 [23], [27], and self-adaptive Gaussian mutation [18].
- The guaranteed convergence particle swarm optimization algorithm (GCPSO) [28].
- The self-adaptive differential evolution algorithm with neighborhood search (SaNSDE) [29].
- The covariance matrix adapting evolutionary strategy algorithm (CMAES) [30].

IV. INVESTIGATING ALTERNATIVE HEURISTIC SPACE DIVERSITY MANAGEMENT STRATEGIES

In this paper three HSD control strategies based on the diversity metric of Grobler *et al.* were evaluated. For all three strategies, certain predetermined bounds within which the HSD should remain, are defined.

HSD(1) attempts to maintain a high HSD throughout the optimization run. As soon as HSD falls below a pre-defined

Algorithm 1: The heterogeneous meta-hyper-heuristic.

Initialize the parent population $\boldsymbol{X}$
$\boldsymbol{M}(t)$ denotes the set of constituent algorithms available at iteration t
$A_i(t)$ denotes the algorithm applied to entity i at iteration t
k denotes the number of iterations between entity-to-algorithm allocation
Initialize $\boldsymbol{M}(t)$
$t = 0$
for *All entities* $i \in \boldsymbol{X}(0)$ **do**
    Randomly select an initial algorithm $A_i(0)$ from $\boldsymbol{M}(t)$ to apply to entity i
end
while *A stopping condition is not met* **do**
    for *All entities* $i \in \boldsymbol{X}(t)$ **do**
        Apply constituent algorithm $A_i(t)$ to entity i for k iterations
    end
    $t = t + k$
    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 using Equation (1).
    for *All entities* $i \in \boldsymbol{X}(t)$ **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 [13]
    end
    if *HSD falls outside of predefined bounds* **then**
        Update $\boldsymbol{A}(t)$ according to the selected HSD strategy.
    end
end

threshold, β, an entity is randomly selected and re-allocated to another randomly selected available constituent algorithm to increase the HSD above the threshold. An example is provided in Figure 2.

HSD(2) is based on the assumption that a desirable heuristic space is diverse at the start of the optimization process and converges towards the end of the optimization run to a less diverse heuristic space where most entities are allocated to a single good performing constituent algorithm taking into account the learning process of the hyper-heuristic algorithm and specific characteristics of the problem being solved. In other words, the HSD(2) strategy attempts to obtain a decreasing HSD throughout the optimization run. This objective is achieved by defining two bounds, similar to the bounds defined in [31] to control solution space diversity.

A linearly decreasing upper, UB_{D_h}, and a lower bound, LB_{D_h}, was defined as follows:

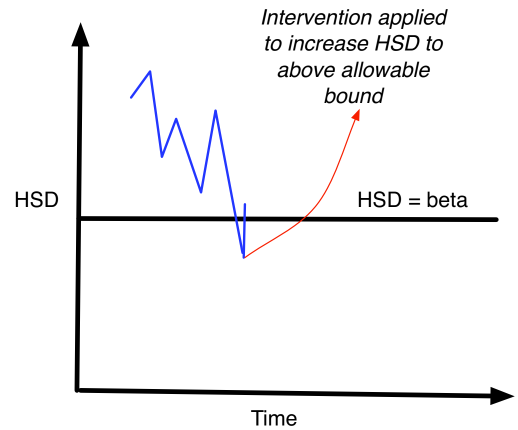

Fig. 2: A conceptual example of the operation of HSD(1).

$$UB_{D_h}(1) = D_h(1) + \gamma D_h(1) \tag{2}$$
$$UB_{D_h}(I_{max}) = \gamma D_h(1) \tag{3}$$
$$LB_{D_h}(1) = D_h(1) - \gamma D_h(1) \tag{4}$$
$$LB_{D_h}(I_{max}) = 0 \tag{5}$$

where γ is a positive constant between 0 and 1, I_{max} is the maximum number of iterations allowed, and HSD, $HSD(t)$, at time t is defined as:

$$D_h(t) = Max_{D_h(t)} \left(1 - \frac{\sum_{i=1}^{I} |T - n_i(t)|}{1.5 n_s} \right) \tag{6}$$

with

$$T = \frac{n_s}{n_a}, \tag{7}$$

where n_a is the number of algorithms available for selection by the hyper-heuristic, n_s is the number of entities in the population, $n_i(t)$ is the number of entities allocated to algorithm i at iteration t, and $UB_{D_h(t)}$ is the upper bound of the HSD measure. For the purposes of this paper, $Max_{D_h(t)}$ was set to 100 so that $D_h(t) \in$ [0, 100].

As soon as the HSD exceeds the upper bound, a HSD decreasing mechanism is applied to a randomly selected individual to decrease the overall HSD of the algorithm. The decreasing mechanism identifies the best performing constituent algorithm and randomly selects an entity which is not currently assigned to this best performing constituent algorithm and assigns it to the best performing algorithm to reduce the overall HSD of the population. When the HSD decreases to below the lower diversity bound, then the HSD increasing mechanism of HSD(1) is applied to increase HSD to fall within the defined bounds. An example is provided in Figure 3.

HSD(3) is similar to HSD(2) apart from the fact that an increasing HSD profile is desirable. The motivation for this strategy was the excellent performance of the EIHH2 algorithm where the best performing constituent algorithm was first allowed to work on the problem until minimal

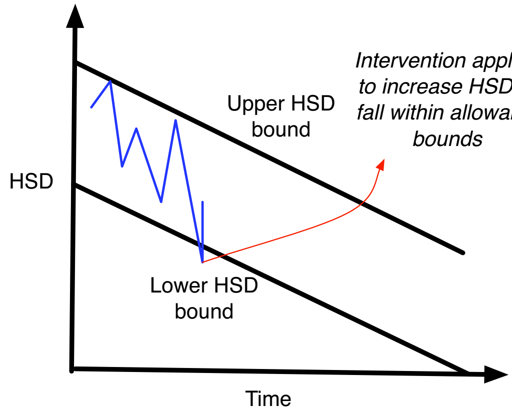

Fig. 3: A conceptual example of the operation of HSD(2).

improvement was obtained and other constituent algorithms could be added. A linearly increasing upper, UB_{D_h}, and a lower bound, LB_{D_h}, was defined for heuristic space diversity as follows:

$$UB_{D_h}(1) = \gamma D_h(1) \tag{8}$$
$$UB_{D_h}(I_{max}) = D_h(1) + \gamma D_h(1) \tag{9}$$
$$LB_{D_h}(1) = 0 \tag{10}$$
$$LB_{D_h}(I_{max}) = D_h(1) - \gamma D_h(1) \tag{11}$$

An example is provided in Figure 4.

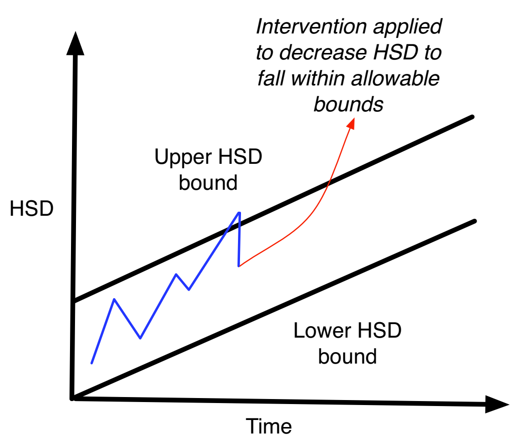

Fig. 4: A conceptual example of the operation of HSD(3).

V. Empirical Evaluation

The three metric-based HSD control strategies and the HMHH algorithm with no HSD control were evaluated on the first 14 problems of the 2005 IEEE Congress of Evolutionary Computation benchmark problem set [32] in 10 and 30 dimensions. The algorithm control parameters, such as the HSD threshold and bounds parameters, will have an influence on algorithm performance and will thus need to be tuned more thoroughly in future. The initial control parameters used are, however, listed in Table I. All the constituent algorithm control parameters were used as specified in [1].

TABLE I: HMHH algorithm parameters.

Parameter	Value used
Entities in common population (n_s)	100
Iterations between re-allocation (k)	5
Size of tabu list	2
HSD threshold (HSD(1)) (β)	0.5
HSD bounds parameter (γ)	0.2

The results of the first comparison between the various heuristic space diversity management techniques are presented in Table II. For all the experiments conducted in this paper, results for each algorithm were recorded over 30 independent simulation runs. The notation, μ and σ, 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. Where the global optimum could not be found within the maximum number of iterations, the final solution at I_{max}, denoted by FFV, was recorded.

Statistical tests were also used to evaluate the significance of the results. The results in Table III 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. As an example, (1-22-5) in row 1 column 2, indicates that the HMHH strategy significantly outperformed HSD(1) only once over the benchmark problem set. Furthermore, 22 draws and 5 losses were recorded.

From the results it is clear that attempting to manage HSD through an HSD metric as done in strategy HSD(1) and HSD(2) does lead to a statistically significant difference in hyper-heuristic performance when compared to the baseline HMHH algorithm where no HSD manipulation is used. The best performing HSD control strategy is HSD(1). In contrast, HSD(3) performed quite poorly.

The best performing solution diversity management strategy from Table III, HSD(1), was also compared under similar conditions to the PAP algorithm of Yao *et al.* [2] and the EIHH2 algorithm of Grobler *et al.* [1]. PAP was found in a previous study [33] to be one of the better performing multi-method algorithms currently available. In the same study, the EIHH2 algorithm was found to be the best performing algorithm. The EIHH2 algorithm also attempts to manage heuristic space diversity for better algorithm performance.

TABLE II: Comparison results of the different HSD management strategies on the 2005 IEEE CEC benchmark problem set.

Prob (Dims)	HMHH				HSD(1)				HSD(2)				HSD(3)			
	FFV		# FEs		FFV		# FEs		FFV		# FEs		FFV		# FEs	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
1(10)	$1.00E-06$	0	11870	538.93	$1.00E-06$	0	12103	605.99	$1.00E-06$	0	11980	636.48	$1.00E-06$	0	13370	1755
1(30)	$1.00E-06$	0	52983	2723.3	$1.00E-06$	0	44767	2035.1	$1.00E-06$	0	44873	1848.6	$1.00E-06$	0	39780	1835.2
2(10)	$1.00E-06$	0	14013	1246.7	$1.00E-06$	0	14027	908.55	$1.00E-06$	0	13717	914.29	$1.00E-06$	0	17810	2975.3
2(30)	$1.00E-06$	0	90727	15876	$1.00E-06$	0	82073	19964	$1.00E-06$	0	80313	17887	$1.00E-06$	0	90380	14391
3(10)	$1.00E-06$	0	22750	3003	$1.00E-06$	0	21077	1673.7	$1.00E-06$	0	21810	1921.4	$1.00E-06$	0	24763	7571.2
3(30)	295.35	998.74	$2.79E+05$	27782	$1.00E-06$	0	$2.19E+05$	45887	12.367	64.564	$2.30E+05$	59316	413.7	1380.7	$2.60E+05$	49508
4(10)	$1.00E-06$	0	15853	1341.8	$1.00E-06$	0	15530	934.46	$1.00E-06$	0	15903	1486.4	$1.00E-06$	0	27937	9077
4(30)	$1.00E-06$	0	$1.50E+05$	27962	$1.00E-06$	0	$1.27E+05$	41055	$1.00E-06$	0	$1.36E+05$	49203	$1.00E-06$	0	$1.13E+05$	38927
5(10)	$1.00E-06$	0	17110	883.31	$1.00E-06$	0	16893	910.07	$1.00E-06$	0	16957	863.71	$1.00E-06$	0	20730	2566.8
5(30)	174.5	278.17	$3.00E+05$	0	18.572	56.955	$3.00E+05$	0	121.87	163.19	$3.00E+05$	0	145.68	225.96	$3.00E+05$	0
6(10)	0	0	33417	7388.8	0.00033333	0.0018257	30357	5412	0.0013333	0.0034575	36953	15182	0.13333	0.72652	38320	15085
6(30)	0.13267	0.72665	$2.43E+05$	33935	0.163	0.73937	$2.26E+05$	38802	0.41167	1.2114	$2.22E+05$	52313	0.53067	1.3761	$2.22E+05$	54578
7(10)	0.162	0.13632	$1.00E+05$	0	0.12167	0.094288	$1.00E+05$	0	0.15767	0.17378	$1.00E+05$	0	0.40267	0.60195	94610	20515
7(30)	0.0036667	0.0080872	$1.30E+05$	$1.13E+05$	0.0046667	0.0068145	$1.71E+05$	$1.23E+05$	0.03	0.02	$3.00E+05$	0	0.009	0.0099481	$2.21E+05$	$1.14E+05$
8(10)	20.06	0.086681	$1.00E+05$	0	20.075	0.10602	$1.00E+05$	0	20.028	0.075113	$1.00E+05$	0	20.052	0.10447	$1.00E+05$	0
8(30)	20.169	0.12041	$3.00E+05$	0	20.227	0.13817	$3.00E+05$	0	20.22	0.14306	$3.00E+05$	0	20.14	0.12029	$3.00E+05$	0
9(10)	0.005	0.0050855	43320	19202	0.006	0.0049827	48507	19386	0.040333	0.17752	44070	19095	0.038667	0.17786	45710	24776
9(30)	2.3763	1.3964	$2.98E+05$	10079	2.2783	1.2808	$2.95E+05$	21120	2.408	1.0675	$3.00E+05$	0	2.6023	1.7112	$2.97E+05$	15370
10(10)	15.556	9.1746	$1.00E+05$	0	15.256	7.3406	$1.00E+05$	0	16.37	9.6037	$1.00E+05$	0	20.716	12.239	$1.00E+05$	0
10(30)	55.768	19.838	$3.00E+05$	0	65.178	23.908	$3.00E+05$	0	62.821	21.911	$3.00E+05$	0	65.474	40.817	$3.00E+05$	0
11(10)	5.0464	2.2525	97283	14880	5.6651	2.0562	$1.00E+05$	0	6.3133	2.1066	$1.00E+05$	0	6.8019	1.3304	$1.00E+05$	0
11(30)	24.378	5.1224	$3.00E+05$	0	28.237	5.5627	$3.00E+05$	0	28.528	5.1712	$3.00E+05$	0	26.981	5.9147	$3.00E+05$	0
12(10)	302.86	546.06	69543	39317	500.86	719.27	65240	40969	284.68	588.11	60770	39431	355.41	620.41	66343	37607
12(30)	2611.5	3622.3	$2.95E+05$	20162	1882.6	2790.8	$2.88E+05$	44191	2767	2840.5	$2.99E+05$	6098	1894.2	2285.1	$2.95E+05$	25524
13(10)	0.43267	0.16282	99050	5203.4	0.46433	0.16733	$1.00E+05$	0	0.529	0.16979	$1.00E+05$	0	0.50233	0.19537	$1.00E+05$	0
13(30)	2.0187	0.44262	$3.00E+05$	0	2.0017	0.47695	$3.00E+05$	0	1.8873	0.45001	$3.00E+05$	0	1.9227	0.81506	$3.00E+05$	0
14(10)	3.6397	0.29122	$1.00E+05$	0	3.5657	0.37704	$1.00E+05$	0	3.6007	0.31844	$1.00E+05$	0	3.686	0.29084	$1.00E+05$	0
14(30)	13.14	0.44011	$3.00E+05$	0	13.072	0.45905	$3.00E+05$	0	13.182	0.40298	$3.00E+05$	0	13.284	0.31641	$3.00E+05$	0

A priori knowledge of the performance of the set of constituent algorithms is, however, required. The optimization process commences with only the best performing constituent algorithm available to the EIHH2 algorithm. At exponential time intervals throughout the rest of the optimization process, additional constituent algorithms are inserted into the set of available constituent algorithms in sequence of better performance. The resulting exponentially increasing HSD was previously shown to improve algorithm performance. Finally, the best performing constituent algorithm (CMAES) [30], was also added for comparison purposes. The results are recorded in Table IV. The number of "Mann-Whitney U wins-draws-losses" obtained by HSD(1) when compared to PAP, EIHH2, and CMAES is recorded in Table V. The complete results of all the constituent algorithms can be found in [34].

TABLE III: Hypotheses analysis of alternative heuristic space diversity control mechanisms.

	HMHH	**HSD(1)**	**HSD(2)**	**HSD(3)**	**TOTAL**
HMHH	NA	1-22-5	3-20-5	8-18-2	12-60-12
HSD(1)	5-22-1	NA	1-25-2	10-16-2	16-63-5
HSD(2)	5-20-3	2-25-1	NA	9-16-3	16-61-7
HSD(3)	2-18-8	2-16-10	3-16-9	NA	7-50-27

Table V shows that the HSD(1) algorithm outperformed PAP and EIHH2 in a number of cases. HSD(1) performed better than PAP 12 times out of 28 instances (43% of the time) and performed better than EIHH2 6 times and equally well 17 times out of 28 instances. HSD(1) did not perform quite as well when compared to CMAES, only performing better 9 times out of 28 instances. 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 entire function evaluation budget on optimization of the actual problem.

VI. Conclusion

This paper has investigated the impact of different heuristic space diversity (HSD) 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 by means of the HSD metric of Grobler *et al.*. The HSD(1) strategy which attempts to maintain a high HSD throughout the optimization run, was shown to outperform the decreasing (HSD(2)) and increasing (HSD(3)) HSD management strategies. Finally, the best performing HSD control strategy was shown to perform well against a popular multi-method algorithm, another meta-hyper-heuristic with embedded HSD control, and the best performing constituent algorithm.

Future research opportunities exist in expanding the experiments to a wider range of benchmark problems, conducting a sensitivity analysis on the algorithm control parameters, a more detailed analysis of the characteristics of the HSD metric and developing additional HSD control strategies. The performance of the HSD(1) strategy can also be evaluated on large-scale real-world applications.

Acknowledgment

This material is based on work funded by the Council for Scientific and Industrial Engineering through their Parliamentary Grant.

References

[1] J. Grobler, A.P. Engelbrecht, G. Kendall, and V.S.S. Yadavalli, "Heuristic space diversity control for improved meta-hyper-heuristic performance," *Information Sciences*, vol. 300, pp. 49–62, 2015.

[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] E. Burke, B. McCollum, A. Meisels, S. Petrovic, and R. Qu, "A graph-based hyper-heuristic for educational timetabling problems," *European Journal of Operational Research*, vol. 176, pp. 177–192, 2007.

[4] G. Ochoa, J. Vázquez-Rodríguez, S. Petrovic, and E. Burke, "Dispatching rules for production scheduling: a hyper-heuristic landscape analysis," *Proceedings of the 2009 Congress on Evolutionary Computation*, pp. 1873–1880, 2009.

[5] G. Ochoa, R. Qu, E. Burke, "Analyzing the landscape of a graph based hyper-heuristic for timetabling problems," *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pp. 341-348, 2009.

[6] 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.

[7] N. Sabar, M. Ayob, G. Kendall, and R, Qu, "Grammatical evolution hyper-heuristic for combinatorial optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 17, pp. 840–861, 2013.

[8] N. Veerapen, Y. Hamadi, and F. Saubion, "Using local search with adaptive operator selection to solve the progressive party problem," *Not yet submitted for publication, Microsoft Research*, 2013.

[9] E. Segredo, C. Segura, and C. León, "Memetic algorithms and hyperheuristics applied to a multiobjectivised two-dimensional packing problem," *Journal of Global Optimization*, pp. 1–26, 2013.

[10] S. Salcedo-Sanz, J. Matías-Román, S. Jiménez-Fernández, A. Portilla-Figueras, and L. Cuadra, "An evolutionary-based hyper-heuristic approach for the jawbreaker puzzle," *Applied Intelligence*, pp. 1–11, 2013.

[11] X. Chen, Y. Ong, M. Lim, and K. Tan, "A multi-facet survey on memetic computation," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 5, pp. 591–607, 2011.

[12] M. Črepinšek, S. Liu, M. Mernik, "Exploration and exploitation in evolutionary algorithms: a survey," *ACM Computing Surveys*, 45, 35, 2013.

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

[14] M. Montazeri, M. Baghshah, and A. Enhesari, "Hyper-Heuristic algorithm for finding efficient features in diagnose of lung cancer disease," *Journal of Basic and Applied Scientific Research*, vol. 3, pp. 134–140, 2013.

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

[16] K, Sim, E. Hart, and B. Paechter, "Learning to solve bin packing problems with an immune inspired hyper-heuristic," *Proceedings of the 12th European Conference on Artificial Life*, pp. 856–863, 2013.

[17] M. Msr, K. Verbeeck, P. De Causmaecker, and G. Van den Berghe, "Design and Analysis of an Evolutionary Selection Hyper-heuristic Framework," *KAHO Sint-Lieven*, 2013.

[18] 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.

[19] 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.

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

TABLE IV: Comparison results of the different HSD management strategies on the 2005 IEEE CEC benchmark problem set.

Prob	HSD(1)				PAP				EIHH(2)				CMAES			
(Dims)	FFV		# FEs		FFV		# FEs		FFV		# FEs		FFV		# FEs	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
1(10)	1.00E − 06	0	12103	605.99	1.00E − 06	0	13857	630.64	1.00E − 06	0	16610	8578.8	1.00E − 06	0	8526.7	302.78
1(30)	1.00E − 06	0	44767	2035.1	1.00E − 06	0	39190	5945.1	1.00E − 06	0	49367	28510	1.00E − 06	0	19110	447.48
2(10)	1.00E − 06	0	14027	908.55	1.00E − 06	0	18760	1030.4	1.00E − 06	0	28097	12646	1.00E − 06	0	9156.7	286.1
2(30)	1.00E − 06	0	82073	19964	1.00E − 06	0	90063	2729.3	1.00E − 06	0	1.09E + 05	72461	1.00E − 06	0	26783	739.1
3(10)	1.00E − 06	0	21077	1673.7	1.00E − 06	0	46067	2040.3	1.00E − 06	0	43277	19645	1.00E − 06	0	13320	379.11
3(30)	1.00E − 06	0	2.19E + 05	45887	1.00E − 06	0	2.88E + 05	5481.5	6796	17006	1.68E + 05	1.06E + 05	1.00E − 06	0	61173	1387.4
4(10)	1.00E − 06	0	15530	934.46	1.00E − 06	0	20483	1424.7	1.00E − 06	0	29923	13977	1.00E − 06	0	9590	283.27
4(30)	1.00E − 06	0	1.27E + 05	41055	4448.8	5562.2	2.87E + 05	39164	1.00E − 06	0	1.39E + 05	73296	1.00E − 06	0	29357	570.35
5(10)	1.00E − 06	0	16893	910.07	1.00E − 06	0	25010	2790.5	1.00E − 06	0	31960	7965	1.00E − 06	0	17433	546.04
5(30)	18.572	56.955	3.00E + 05	0	1115.5	1446.5	3.00E + 05	0	670.92	905.42	2.88E + 05	45133	1.00E − 06	0	1.15E + 05	3960.1
6(10)	0.00033333	0.0018257	30357	5412	0.13267	0.72665	50613	10870	0.00066667	0.0025371	41690	18418	0.00066667	0.0025371	18950	744.52
6(30)	0.163	0.73937	2.26E + 05	38802	0.51867	1.3463	2.75E + 05	27166	0.901	2.9717	2.34E + 05	52399	0.13267	0.72665	1.20E + 05	37870
7(10)	0.12167	0.094288	1.00E + 05	0	0.0026667	0.0058329	62867	35481	0.61767	1.0662	67773	40968	1267	4.63E − 13	1.00E + 05	0
7(30)	0.0046667	0.0068145	1.71E + 05	1.23E + 05	0	0	82047	41811	0.0083333	0.012888	1.52E + 05	1.20E + 05	4696.3	2.78E − 12	3.00E + 05	0
8(10)	20.075	0.10602	1.00E + 05	0	20.058	0.086279	1.00E + 05	0	20.034	0.079333	1.00E + 05	0	20.312	0.11271	1.00E + 05	0
8(30)	20.227	0.13817	3.00E + 05	0	20.264	0.16296	3.00E + 05	0	20.235	0.26664	3.00E + 05	0	20.892	0.17459	3.00E + 05	0
9(10)	0.006	0.0049827	48507	19386	0.22033	0.40067	68493	26929	0.038	0.17798	39850	20348	1.9457	1.5105	88203	30601
9(30)	2.2783	1.2808	2.95E + 05	21120	2.6757	1.92	2.81E + 05	55032	2.446	1.9105	2.85E + 05	37453	39.564	6.4543	3.00E + 05	0
10(10)	15.256	7.3406	1.00E + 05	0	12.857	6.2645	1.00E + 05	0	14.653	17.797	1.00E + 05	0	1.647	1.1767	90947	27625
10(30)	65.178	23.908	3.00E + 05	0	70.555	19.764	3.00E + 05	0	57.979	48.81	3.00E + 05	0	9.391	3.2817	3.00E + 05	0
11(10)	5.6651	2.0562	1.00E + 05	0	4.085	1.2619	1.00E + 05	0	3.7648	2.8672	89520	27470	1.2989	1.3647	30310	10496
11(30)	28.237	5.5627	3.00E + 05	0	20.034	2.6451	3.00E + 05	0	24.223	9.253	3.00E + 05	0	9.0255	3.0546	3.00E + 05	0
12(10)	500.86	719.27	65240	40969	231.53	539.16	71737	28064	220.08	561.14	63337	29948	1546.1	2735.5	70053	43090
12(30)	1882.6	2790.8	2.88E + 05	44191	3573.5	3310.2	3.00E + 05	0	3296.3	5065	2.97E + 05	13275	20324	19261	3.00E + 05	0
13(10)	0.46433	0.16733	1.00E + 05	0	0.498	0.15873	1.00E + 05	0	0.50733	0.16607	1.00E + 05	0	0.897	0.25323	1.00E + 05	0
13(30)	2.0017	0.47695	3.00E + 05	0	1.74	0.42099	3.00E + 05	0	2.0943	0.48971	3.00E + 05	0	3.179	0.56064	3.00E + 05	0
14(10)	3.5657	0.37704	1.00E + 05	0	3.4067	0.31705	1.00E + 05	0	3.1803	0.50128	1.00E + 05	0	2.5847	0.53024	1.00E + 05	0
14(30)	13.072	0.45905	3.00E + 05	0	13.026	0.31914	3.00E + 05	0	12.492	0.99427	3.00E + 05	0	10.394	0.8103	3.00E + 05	0

TABLE V: Further hypotheses analysis of the best performing HSD control mechanism (HSD(1)) against other popular multi-method algorithms.

	PAP	EIHH2	CMAES	TOTAL
HSD(1)	12-12-4	6-17-5	9-2-17	27-31-26

[21] 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.

[22] 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.

[23] 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.

[24] A. Caponio, G.L. Cascella, F. Neri, N. Salvatore, M. Sumner, "A fast adaptive memetic algorithm for on-line and off-line control design of pmsm drives," *IEEE Transactions on System Man and Cybernetics-Part B, Special Issue on Memetic Algorithms*, 37 (1) (2007) 2841.

[25] V. Tirronen, F. Neri, T. Krkkinen, K. Majava, T. Rossi, "An enhanced memetic differential evolution in filter design for defect detection in paper production," *Evolutionary Computation* 16 (2008) 529555.

[26] A. Caponio, F. Neri, V. Tirronen, "Super-fit control adaptation in memetic differential evolution frameworks," *Soft Computing-A Fusion of Foundations, Methodologies and Applications* 13 (8) (2009) 811831.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

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

[32] 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.

[33] J. Grobler, A.P. Engelbrecht, G. Kendall, and V.S.S. Yadavalli, "The entity-to-algorithm allocation problem: extending the analysis," *Proceedings of the IEEE Symposium on Computational Intelligence and Ensemble Learning*, pp. 1–8, 2014.

[34] 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 IEEE Congress on Evolutionary Computation*, pp. 570–577, 2013.