Information Sciences xxx (2014) xxx–xxx

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Heuristic space diversity control for improved meta-hyper-heuristic performance

Q1 Jacomine Grobler [a,*], Andries P. Engelbrecht [b], Graham Kendall [c,d], V.S.S. Yadavalli [e]

[a] Department of Industrial and Systems Engineering, University of Pretoria, Council for Scientific and Industrial Research, Pretoria, South Africa
[b] Department of Computer Science, University of Pretoria, Pretoria, South Africa
Q2 [c] School of Computer Science, University of Nottingham, UK
[d] University of Nottingham Malaysia Campus, Malaysia
[e] Department of Industrial and Systems Engineering, University of Pretoria, Pretoria, South Africa

ARTICLE INFO

Article history:
Received 11 March 2014
Received in revised form 14 August 2014
Accepted 7 November 2014
Available online xxxx

Keywords:
Hyper-heuristics
Diversity management
Heuristic space diversity

ABSTRACT

This paper expands on the concept of heuristic space diversity and investigates various strategies for the management of heuristic space diversity within the context of a meta-hyper-heuristic algorithm in search of greater performance benefits. Evaluation of various strategies on a diverse set of floating-point benchmark problems shows that heuristic space diversity has a significant impact on hyper-heuristic performance. An exponentially increasing strategy (EIHH) obtained the best results. The value of *a priori* information about constituent algorithm performance on the benchmark set in question was also evaluated. Finally, EIHH demonstrated good performance when compared to a popular population based algorithm portfolio algorithm and the best performing constituent algorithm.

1. Introduction

Over the last decade, research into hyper-heuristics has 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, hyper-heuristics search through a heuristic space of available heuristics or heuristic components [3]. The idea is to either select or construct an "optimal" selection of heuristics to address the specific problem at hand. A meta-hyper-heuristic can be defined as a hyper-heuristic where the constituent or low level algorithms consist of meta-heuristic algorithms [17].

Diversity management of the decision space is another important concept that has received increasing attention recently [12]. 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 exploration near the end of the optimization run, time is wasted which could have been used to further refine promising solutions. Based on the importance of the 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 hyper-heuristic performance.

This paper describes the concept of heuristic space diversity (HSD) and investigates whether influencing the diversity of the set of available constituent algorithms has an effect on hyper-heuristic performance. Six strategies for controlling HSD

* Corresponding author.
Q1 *E-mail address:* jacomine.grobler@gmail.com (J. Grobler).

http://dx.doi.org/10.1016/j.ins.2014.11.012
0020-0255/

throughout the optimization process are introduced. These strategies include linearly increasing and decreasing and exponentially increasing and decreasing HSD strategies. The strategies are evaluated on a set of varied floating-point benchmark problems. The value of *a priori* knowledge of constituent algorithm performance on the benchmark problem set under consideration, is also investigated. The performance of the HSD control strategies are also compared to that of the population based portfolio algorithm [31], which is a well known multi-method algorithm, and the covariance matrix adapting evolutionary strategy algorithm (CMAES) [1], which is the best performing constituent algorithm. The empirical study will show that HSD and mechanisms to control it does have a positive influence on the performance of the hyper-heuristic.

To the best of the authors' knowledge, Grobler et al. [21] were the first authors to explicitly introduce the concept of heuristic space diversity and the control of HSD to influence hyper-heuristic performance. This paper expands on that previous work by extending the analysis to a larger benchmark problem set and removing the need for *a priori* knowledge when using an increasing HSD control strategy. This generalization does, unfortunately, result in reduced performance which is evaluated in detail in this paper.

The rest of the paper is organized as follows: Section 2 provides some background on hyper-heuristics, the heuristic space, and existing diversity management strategies. Section 3 provides a brief overview of the heterogeneous meta-hyper-heuristic (HMHH) algorithm used as the basis for this investigation. Section 4 provides the definition of HSD and describes the HSD control strategies which are evaluated, before the results are documented in Section 5. Finally, the paper is concluded in Section 6.

2. Hyper-heuristics, heuristic space, and diversity

Burke et al. [4] defines a hyper-heuristic as "*a search method or learning mechanism for selecting or generating heuristics to solve computational search problems*". The basic idea is not only to obtain an appropriate solution for a specific problem, but rather to focus on automating the development of the method used to obtain an appropriate solution. To achieve this aim, a search space of low level heuristics is defined, upon which a high level search strategy operates. Typical hyper-heuristics thus consist of three elements namely, one or more entities representing solutions which are evolved over time, a set of low level heuristics, and a high level strategy responsible for guiding the use of the low level heuristics in evolving the entities over time.

This approach is in contrast to the "traditional" meta-heuristic strategy of searching directly through a search space of candidate solutions. From this comparison, it is clear that hyper-heuristic implementations benefit from increased generality, a valuable attribute considering the specialist resources required for the development of advanced AI-based algorithms as well as the problem-dependent nature of most meta-heuristic algorithm implementations. A comprehensive review of hyper-heuristic research conducted over the last two decades can be found in Burke et al. [3].

The relationship between heuristic space and solution space was considered in detail for the first time in Burke et al. [6]. Since then, further efforts to analyze the heuristic space by means of landscape analysis were also conducted [28,27,30]. 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 [32,19,16], 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. The AMALGAM algorithm of Vrugt et al. [43] makes use of a species selection mechanism to maintain solution space diversity. Sabar et al. [35] and Veerapen et al. [42] take both solution space diversity and solution quality into account when allocating entities to algorithms. Segredo et al. [37] converts a single objective problem into a multi-objective optimization problem through the addition of a second diversity-based objective. A hyper-mutation operator is triggered in the evolutionary-based hyper-heuristic of Salcedo-Sanz et al. [36] when the solution space diversity drops to zero.

Solution space diversity management is also an important consideration in a field closely related to hyper-heuristics, namely memetic computing [11]. Črepinšek 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 of Caponio et al. [7]. Fitness diversity-adaptive algorithms are based on the idea of using population diversity to guide the exploration versus exploitation balance of the algorithm. Multiple refinement methods are usually involved, each with a different impact on solution space diversity. A fitness diversity measure is calculated at each iteration and a self-adaptive criterion determines which refinement method is applied. Caponio et al.'s algorithm made use of a Hooke–Jeeves [22] and Nelder–Mead simplex algorithm [26], but a large number of other fitness-diversity based algorithms have also been proposed utilizing different types of diversity measures and different algorithms to increase or decrease population diversity [8,23,40].

More closely related to heuristic space diversity is the issue of selecting the set of low level heuristics. Montazeri et al. [25] ensured that their set of low level heuristics contains both exploiter heuristics, designed for intensification, and explorer heuristics, aimed at diversification. Peng et al. [31] proposed a pairwise metric which can be used to determine the risk associated with an algorithm failing to solve the problem in question. Engelbrecht [13] selected complementary swarm behaviours in a heterogeneous particle swarm optimization (PSO) algorithm by analyzing the exploration–exploitation finger prints of the different PSO updates.

Ren et al. [34] identified the issue of low level heuristic parameters that could influence performance. They addressed this issue through the development of a hyper-heuristic with low level parameter optimization consisting of a low level heuristic management module and a low level parameter management model. The additional parameter optimization variables did, however, have a significant influence on the size of the search space. This issue was addressed by means of a heuristic space reduction mechanism. The low level heuristics were subdivided into explorers and exploiters and the algorithm continually alternated between the two types of heuristics in an attempt to manage the exploration–exploitation trade-off.

Recently, researchers have started to dynamically update the set of low level heuristics during the optimization run. Sim et al. [38] 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 Mısır et al. [24] 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. A number of decision mechanisms for activating or de-activating these sub-mechanisms were also employed. The algorithm won the first international domain heuristic search challenge where problems from six different domains were considered [2].

Caraffni et al. [9] investigated the advantages of employing diverse local search components during the development of their parallel memetic structure (PMS). When compared to its individual components, the difference in performance was not that obvious for low dimensional problems (30 dimensions), but PMS outperformed its components for problems of 1000 dimensions. PMS has also been used in a highly successful computational prototype for automatic design of optimization algorithms: the separability prototype for automatic memes [10].

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 and its predecessor [21] are the first to attempt to define and measure the concept of heuristic space diversity and to manage the diversity of low level heuristic-to-entity allocation to improve hyper-heuristic performance.

3. The heterogeneous meta-hyper-heuristic algorithm

Due to its excellent performance against other popular multi-method algorithms, the tabu-search based HMHH algorithm of Grobler et al. [17], illustrated in Fig. 1, was used as a basis for investigating the management of heuristic space diversity. 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 kth 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

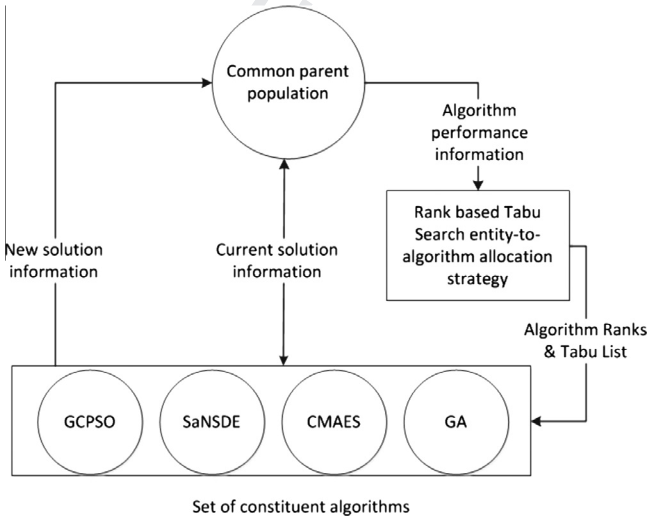

Fig. 1. The heterogeneous meta-hyper-heuristic (HMHH). The HMHH algorithm makes use of four low level meta-heuristics, namely the guaranteed convergence particle swarm optimization algorithm (GCPSO) [41], the self-adaptive differential evolution algorithm with neighborhood search (SaNSDE) [44], a genetic algorithm (GA) with a floating-point representation, tournament selection, blend crossover [14,29], and self-adaptive Gaussian mutation [18], and the covariance matrix adapting evolutionary strategy algorithm (CMAES) [1].

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,

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

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 Burke et al. [5].

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 [14,29], and self-adaptive Gaussian mutation [18].
- The guaranteed convergence particle swarm optimization algorithm (GCPSO) [41].
- The self-adaptive differential evolution algorithm with neighborhood search (SaNSDE) [44].
- The covariance matrix adapting evolutionary strategy algorithm (CMAES) [1].

Two aspects were considered in the selection of the set of constituent algorithms. Firstly, the above algorithms have been shown to exhibit different exploration–exploitation trade-offs and secondly, the algorithms were shown to perform well as single-method algorithms [20].

This set of constituent algorithms is updated throughout the optimization run according to the heuristic space diversity strategies introduced in the next section where the growing and shrinking mechanisms of the set of constituent algorithms is discussed.

Algorithm 1. The heterogeneous meta-hyper-heuristic

```
Initialize the parent population X
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 M(t) according to the selected heuristic space diversity strategy
t = 0
for All entities i ∈ X(0) do
    Randomly select an initial algorithm A_i(0) from M(t) to apply to entity i
end
while A stopping condition is not met do
    for All entities i ∈ X(t) do
        Apply constituent algorithm A_i(t) to entity i for k iterations
    end
    t = t + k
    Calculate Q_δm(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 ∈ X(t) do
        Use Q_δm(t) as input to select constituent algorithm A_i(t) according to
        the rank based tabu search mechanism described in Burke et al. (2003)
    end
    Update M(t) according to the selected heuristic space diversity strategy.
end
```

4. Heuristic space diversity management

The concept of heuristic space diversity is best illustrated by means of an example. In Fig. 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

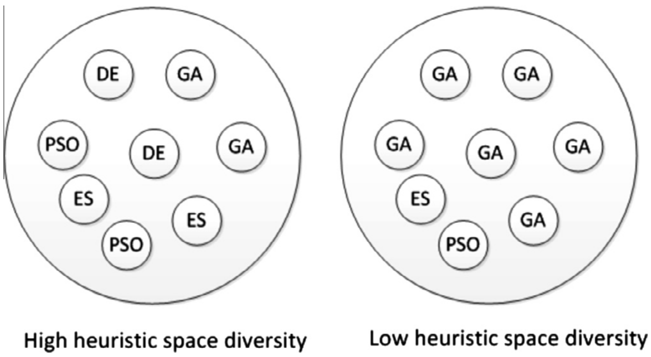

Fig. 2. An example of a population with a high HSD and a population with a low HSD.

of the right were allocated to the GA with only one entity each allocated to PSO and ES. This population can be described as having a low HSD.

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

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

with

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

where n_a is the number of algorithms available for selection by the hyper-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, $UB_{D_h(t)}$ was set to 100 so that $D_h(t) \in [0, 100]$.

The idea of the measure is to calculate a target number of entities per algorithm, T. The absolute deviation between this target and the actual allocation of entities to algorithms is then used to determine the HSD associated with the entity-to-algorithm allocation. Maximum diversity would be achieved when all entities are assigned equally between algorithms and this would translate into a $D_h(t)$ of 100.

Five strategies for controlling HSD throughout the optimization process, inspired by Ratnaweera et al.'s PSO parameter control strategies [33], is investigated in this paper. These strategies are described below:

- **The baseline HMHH algorithm** – This algorithm is the standard HMHH algorithm implemented as described in Section 3. No effort is made to manipulate the HSD in this case.
- **Linearly decreasing HSD hyper-heuristic (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. Removing the worst performing algorithm allows more function evaluations for the better performing algorithms, leading to better algorithm performance. 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 hyper-heuristic 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 hyper-heuristic (EDHH)** – This algorithm is characterized by an exponentially decreasing HSD. All constituent algorithms are available for allocation to entities at the start of the optimization run and algorithms are removed according to their performance at predetermined time intervals. This time, however, the algorithms are removed at exponentially increasing time intervals at 10,400, 23,800, 42,700 and 75,000 function evaluations. The result is a faster changeover from exploration to exploitation of the heuristic space.
- **Linearly increasing HSD hyper-heuristic (LIHH)** – At the start of the optimization run only one constituent algorithm is made available to the hyper-heuristic. As the optimization process progresses, additional algorithms are made available at predetermined linear constant time intervals. Here the hyper-heuristic is forced to move from exploitation to exploration. The idea is to obtain maximum performance gains from the first algorithm. As the performance gains decrease, the rest of the constituent algorithms become available to diversify the heuristic space and improve the overall algorithm performance. Two versions of the LIHH algorithm were investigated. LIHH1 assumes *a priori* 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, with additional algorithms being made available according to their ranking. This has the effect of exploiting the best performing

constituent algorithm at the start, before other algorithms are considered. LIHH2 requires no *a priori* knowledge. Constituent algorithms are sequenced randomly and made available to the HH one by one at linear time intervals without any consideration of previous algorithm performance.

- **Exponentially increasing HSD hyper-heuristic (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. Exponential time intervals increase the rate of change of HSD, leading to a faster changeover from exploitation to exploration. Similar to the LIHH algorithm, two versions, namely EIHH1 (with *a priori* knowledge) and EIHH2 (without *a priori* knowledge), are investigated.

5. Empirical evaluation

The various HSD control strategies were evaluated on the first 17 problems of the 2005 IEEE Congress of Evolutionary Computation benchmark problem set [39] in 10, 30, and 50 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 1 were found to work well for the algorithms under study during previous research by the authors [21]. $m \rightarrow n$ indicates that the associated parameter is decreased linearly from m to n over 95% of the maximum number of iterations, I_{max}.

The results of the first comparison between the various heuristic space diversity management techniques without *a priori* knowledge are presented in Tables 2–4. For all the experiments conducted in this paper, results for each algorithm were recorded over 30 independent simulation runs. 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 5 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, (5-31-15) in row 1 column 2, indicates that the HMHH strategy significantly outperformed LDHH five times over the benchmark problem set. Furthermore, 31 draws and 15 losses were recorded.

From the results it is clear that attempting to manage HSD does lead to a statistically significant difference in hyper-heuristic performance when compared to strategies where no HSD manipulation is used. Table 5 shows that the strategies where the HSD was controlled performed statistically significantly better than the baseline HMHH algorithm for 67 cases. In contrast, only 20 cases of worse performance could be identified out of the 204 cases which were tested. LDHH was the best performing algorithm for the first five unimodal problems, however, comparing the increasing HSD strategies (LIHH2 and EIHH2) to the decreasing HSD strategies (LSHH and EDHH) over the entire benchmark problem set resulted in 53 wins, 101 draws, and 30 losses. These results are mainly due to the good performance of the increasing strategies on the more complex multi-modal problems and indicate that the increasing HSD strategies performed better for the selected benchmark

Table 1
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 ($n_a \leqslant 2$)	0
PSO parameters	
Acceleration constant (c_1)	$2.0 \rightarrow 0.7$
Acceleration constant (c_2)	$0.7 \rightarrow 2.0$
Inertia weight (w)	$0.9 \rightarrow 0.4$
SaNSDE parameters	As specified in Yang et al. [44]
GA parameters	
Probability of crossover (p_c)	$0.6 \rightarrow 0.4$
Probability of mutation (p_m)	0.1
Blend crossover parameter (α)	0.5
GA tournament size (N_t)	13
CMAES parameters	As specified in Auger and Hansen [1]

Table 2
Results of the investigation into alternative heuristic space diversity control mechanisms: Baseline HMHH algorithm (no heuristic space diversity control strategy).

Prob (Dims)	HMHH (baseline)			
	FFV		# FEs	
	μ	σ	μ	σ
1(10)	$1.00E-06$	0	13,310	694.98
1(30)	$1.00E-06$	0	45,040	1783.8
1(50)	$1.00E-06$	0	**74,587**	2362.3
2(10)	$1.00E-06$	0	31,577	2659
2(30)	$1.00E-06$	0	$2.2039e+05$	49,695
2(50)	0.0013341	0.0043412	$4.9119e+05$	17,885
3(10)	238.87	910.04	84,027	11,776
3(30)	$1.3519e+05$	73,374	$3.00E+05$	0
3(50)	$4.1224e+05$	$1.9264e+05$	$5.00E+05$	0
4(10)	$1.00E-06$	0	43,940	9759.9
4(30)	0.94533	2.8539	$3.00E+05$	0
4(50)	340.53	332.25	$5.00E+05$	0
5(10)	$1.00E-06$	0	17,980	1572
5(30)	1360.6	728.2	$3.00E+05$	0
5(50)	4906	1030.8	$5.00E+05$	0
6(10)	0.119	0.41785	65,910	19,543
6(30)	4.2753	22.125	$2.5104e+05$	45,175
6(50)	8.0497	21.09	$4.7265e+05$	45,948
7(10)	0.44533	0.34199	$1.00E+05$	0
7(30)	0.0046667	0.0081931	$1.6146e+05$	$1.153e+05$
7(50)	0.003	0.0059596	$2.3284e+05$	$1.7801e+05$
8(10)	20.061	0.10594	$1.00E+05$	0
8(30)	20.166	0.10969	$3.00E+05$	0
8(50)	**20.262**	0.099158	$5.00E+05$	0
9(10)	0.038	0.17798	38,270	19,385
9(30)	2.245	1.8063	$2.8956e+05$	30,956
9(50)	14.237	4.7209	$5.00E+05$	0
10(10)	16.157	7.0261	$1.00E+05$	0
10(30)	72.135	30.203	$3.00E+05$	0
10(50)	89.756	21.106	$5.00E+05$	0
11(10)	6.8758	1.7015	$1.00E+05$	0
11(30)	27.311	4.4467	$3.00E+05$	0
11(50)	52.332	6.9296	$5.00E+05$	0
12(10)	466.17	620.43	88,740	19,820
12(30)	4489.4	5300.9	$3.00E+05$	0
12(50)	60,290	54,326	$5.00E+05$	0
13(10)	0.50233	0.22999	$1.00E+05$	0
13(30)	**1.8733**	0.441	$3.00E+05$	0
13(50)	**3.8647**	1.1894	$5.00E+05$	0
14(10)	3.653	0.2848	$1.00E+05$	0
14(30)	13.14	0.44261	$3.00E+05$	0
14(50)	22.517	0.55336	$5.00E+05$	0
15(10)	263.91	212.3	**76,663**	33,922
15(30)	347.11	97.654	$3.00E+05$	0
15(50)	302.11	109.61	$5.00E+05$	0
16(10)	137.61	25.198	$1.00E+05$	0
16(30)	195.48	138.18	$3.00E+05$	0
16(50)	159.99	128.26	$5.00E+05$	0
17(10)	134.52	17.935	$1.00E+05$	0
17(30)	**163.15**	110.67	$3.00E+05$	0
17(50)	130.69	108.95	$5.00E+05$	0

problem set. When the rate of change of diversity is considered, the difference in performance is more subtle. For the decreasing strategies, the linearly decreasing HSD algorithm outperformed the exponentially decreasing algorithm. For the increasing strategies the best performing algorithm was the exponentially increasing EIHH2.

Better insight into these results can be obtained by studying HSD over the total number of iterations of the algorithm. Fig. 3 plots the HSD of the median run of each of the HSD control strategies for problems 13–16 from the CEC 2005 problem set in 50 dimensions. Only the graphs of these four problems are provided due to space constraints. These problems are some of the hardest problems investigated in this paper and the graphs associated with the other problems are very similar. The graphs for all problems in all the considered dimensions are, however, available from the corresponding author on request.

As expected, the EDHH converged the quickest to a lower HSD where the allocation of entities-to-algorithms have stabilized. LDHH converged the second quickest, followed by HMHH. Similar to entities converging in a solution space, it is also

Table 3
Results of the investigation into alternative heuristic space diversity control mechanisms: LDHH (linearly decreasing heuristic space diversity) and EDHH (exponentially decreasing heuristic space diversity).

Prob (Dims)	LDHH				EDHH			
	FFV		# FEs		FFV		# FEs	
	μ	σ	μ	σ	μ	σ	μ	σ
1(10)	1.00E−06	0	**12,017**	642.78	1.00E−06	0	12,090	579.15
1(30)	1.00E−06	0	**43,653**	1500.7	1.00E−06	0	46,723	2038.7
1(50)	1.00E−06	0	84,380	9942.9	1.00E−06	0	77,327	4280.2
2(10)	1.00E−06	0	**13,743**	959.41	1.00E−06	0	14,393	902.84
2(30)	1.00E−06	0	**97,973**	14,510	1.00E−06	0	2.0176e+05	95,248
2(50)	0.0030009	0.016431	3.7491e+05	1.6617e+05	0.55267	1.3763	4.11e+05	1.6417e+05
3(10)	1.00E−06	0	20120	2020.8	1.00E−06	0	**19,550**	1444.1
3(30)	55,462	48,441	2.9701e+05	16,395	70,567	75,788	3.00E+05	0
3(50)	2.1491e+05	1.4672e+05	5.00E+05	0	5.266e+05	3.2535e+05	5.00E+05	0
4(10)	1.00E−06	0	**15,550**	1344.9	1.00E−06	0	16,003	1023.7
4(30)	0.0033343	0.018257	1.502e+05	1.0167e+05	1.1467	3.7805	1.6407e+05	1.0009e+05
4(50)	37.163	181.92	2.837e+05	1.7079e+05	295.84	1291.7	**2.6099e+05**	1.4466e+05
5(10)	1.00E−06	0	17,313	1044.1	1.00E−06	0	**17,077**	1054
5(30)	1086.8	849.77	3.00E+05	0	1200.2	662.45	3.00E+05	0
5(50)	5052.4	1274.2	5.00E+05	0	5238.2	1421.9	5.00E+05	0
6(10)	0.26533	1.0098	37,347	18,132	0.016	0.083896	**35,933**	13,900
6(30)	1.1653	1.9058	2.3937e+05	54,813	1.076	1.7304	2.7588e+05	41,820
6(50)	21.662	26.963	4.8019e+05	53,900	35.225	49.587	5.00E+05	0
7(10)	0.15467	0.12678	1.00E+05	0	0.106	0.093055	93,167	17,726
7(30)	0.004	0.010372	1.4953e+05	1.1659e+05	0.0066667	0.01561	1.4694e+05	1.1871e+05
7(50)	0.0016667	0.0064772	**1.8404e+05**	1.4432e+05	0.0056667	0.011351	2.1627e+05	1.7421e+05
8(10)	**20.034**	0.077442	99,290	3888.8	20.114	0.16596	1.00E+05	0
8(30)	**20.15**	0.13011	3.00E+05	0	20.239	0.1972	3.00E+05	0
8(50)	20.622	0.34904	5.00E+05	0	20.829	0.3542	5.00E+05	0
9(10)	0.0046667	0.0050742	**33,440**	15,441	0.52933	0.88824	54,163	35,998
9(30)	9.8607	5.3668	2.9727e+05	14,953	18.916	8.0898	3.00E+05	0
9(50)	29.842	16.353	5.00E+05	0	48.514	19.672	5.00E+05	0
10(10)	19.136	9.4447	1.00E+05	0	15.582	9.6233	98,393	8800.1
10(30)	73.098	31.88	3.00E+05	0	73.336	28.963	3.00E+05	0
10(50)	40.668	16.476	5.00E+05	0	**29.142**	20.344	5.00E+05	0
11(10)	6.9141	1.3129	1.00E+05	0	6.1124	2.0343	98,277	9439.1
11(30)	21.74	7.7398	3.00E+05	0	19.711	5.2357	3.00E+05	0
11(50)	49.067	9.0971	5.00E+05	0	**43.424**	13.12	5.00E+05	0
12(10)	156.24	420.61	69,417	35,661	131.04	401.7	**60,357**	37,188
12(30)	4646.4	4889	3.00E+05	0	10,348	13,200	3.00E+05	0
12(50)	43,832	26,157	5.00E+05	0	56,752	44,038	5.00E+05	0
13(10)	0.54667	0.19359	1.00E+05	0	0.585	0.22664	1.00E+05	0
13(30)	2.7937	0.87927	3.00E+05	0	2.2563	0.63171	3.00E+05	0
13(50)	4.9403	2.2133	5.00E+05	0	4.794	3.0517	5.00E+05	0
14(10)	3.457	0.54519	1.00E+05	0	3.5107	0.42496	1.00E+05	0
14(30)	13.204	0.31822	3.00E+05	0	13.185	0.38084	3.00E+05	0
14(50)	22.283	0.91453	5.00E+05	0	22.524	1.0094	5.00E+05	0
15(10)	222.08	194.97	79,050	33,379	251.95	179.67	91,060	23,362
15(30)	340.26	89.804	3.00E+05	0	330.33	102.54	3.00E+05	0
15(50)	297.23	99.908	5.00E+05	0	307.76	101.03	5.00E+05	0
16(10)	135.35	27.021	94,230	6630.6	134.17	26.114	78,710	20,528
16(30)	161.8	103.94	3.00E+05	0	189.7	154.73	3.00E+05	0
16(50)	122.27	118.27	5.00E+05	0	**102.04**	125.19	4.9486e+05	13,023
17(10)	137.86	18.663	1.00E+05	0	137.21	25.201	1.00E+05	0
17(30)	198.1	160.95	3.00E+05	0	207.01	180.66	3.00E+05	0
17(50)	169.22	159.14	5.00E+05	0	**120.22**	129.83	5.00E+05	0

evident that the slower converging LDHH was much more capable of adjusting and recovering towards a higher HSD when this adjustment was required. The alternative would be converging too quickly to a suboptimal entity-population allocation which could adversely affect solution quality if insufficient diversity remained in the set of available constituent algorithms.

The success of the increasing HSDs can be largely attributed to the hyper-heuristic being able to exploit the performance benefits of a single algorithm before expanding the set of available algorithms. The resulting increased HSD allowed the hyper-heuristic to incorporate new types of operators to continue the optimization process from where the first algorithm may have already stagnated. With regard to the rate of change of HSD, EIHH2 started exploring the heuristic space sooner than LIHH2 and maintained a higher HSD for longer when compared to LIHH2, leading to better exploration of the heuristic space and better overall solution quality.

Table 4
Q7 Results of the investigation into alternative heuristic space diversity control mechanisms: LIHH2 (linear increasing heuristic space diversity without *a priori* knowledge) and EIHH2 (exponentially increasing heuristic space diversity without *a priori* knowledge).

Prob (Dims)	LIHH2				EIHH2			
	FFV		# FEs		FFV		# FEs	
	μ	σ	μ	σ	μ	σ	μ	σ
1(10)	1.00E − 06	0	18,870	8668.3	1.00E − 06	0	16,610	8578.8
1(30)	1.00E − 06	0	60,323	30,956	1.00E − 06	0	49,367	28,510
1(50)	1.00E − 06	0	1.053e + 05	62,707	1.00E − 06	0	89,440	55,856
2(10)	1.00E − 06	0	32,310	21,104	1.00E − 06	0	28,097	12,646
2(30)	1.00E − 06	0	1.7347e + 05	85,128	1.00E − 06	0	1.0933e + 05	72,461
2(50)	0.0090008	0.031985	3.4577e + 05	1.8287e + 05	1.00E − 06	0	**2.9271e + 05**	1.5095e + 05
3(10)	3.7033	17.172	57,867	37,168	1.00E − 06	0	43,277	19,645
3(30)	51,869	86,498	2.1972e + 05	1.0788e + 05	6796	17,006	**1.6836e + 05**	1.0592e + 05
3(50)	2.1453e + 05	2.7217e + 05	**3.975e + 05**	1.5265e + 05	99,777	1.0943e + 05	4.2713e + 05	1.2437e + 05
4(10)	1.00E − 06	0	39,220	23,734	1.00E − 06	0	29,923	13,977
4(30)	1.00E − 06	0	1.4722e + 05	97,022	1.00E − 06	0	**1.3888e + 05**	73,296
4(50)	344.97	930.78	3.9416e + 05	1.6501e + 05	0.578	2.5913	2.9951e + 05	1.712e + 05
5(10)	1.00E − 06	0	46,650	20,429	1.00E − 06	0	31,960	7965
5(30)	1064.4	1734.8	2.8774e + 05	34,956	**670.92**	905.42	2.8844e + 05	45,133
5(50)	3908.7	2977.8	5.00E + 05	0	**3395.5**	2249.4	4.605e + 05	1.0539e + 05
6(10)	0.25133	0.95807	57,533	24,690	0.00066667	0.0025371	41,690	18,418
6(30)	1.8913	5.2325	**2.2451e + 05**	77,133	0.901	2.9717	2.3416e + 05	52,399
6(50)	6.1297	8.7732	4.6432e + 05	76,351	5.24	7.869	**4.4701e + 05**	80,799
7(10)	0.67867	1.1493	72,220	43,161	0.61767	1.0662	**67,773**	40,968
7(30)	0.0043333	0.0089763	**1.2978e + 05**	1.1301e + 05	0.0083333	0.012888	1.5247e + 05	1.1976e + 05
7(50)	0.0023333	0.0067891	2.3375e + 05	1.7057e + 05	0.003	0.0079438	2.0818e + 05	1.9341e + 05
8(10)	20.057	0.10968	1.00E + 05	0	**20.034**	0.079333	1.00E + 05	0
8(30)	20.226	0.27014	3.00E + 05	0	20.235	0.26664	3.00E + 05	0
8(50)	20.463	0.38992	5.00E + 05	0	20.558	0.36105	5.00E + 05	0
9(10)	0.067333	0.24812	46,027	28,073	0.038	0.17798	39,850	20,348
9(30)	3.4097	3.542	**2.6593e + 05**	68,493	2.446	1.9105	2.8476e + 05	37,453
9(50)	**13.851**	8.9144	4.5523e + 05	1.1611e + 05	16.123	14.872	4.8928e + 05	58,698
10(10)	17.456	15.677	97,077	16,012	**14.653**	17.797	1.00E + 05	0
10(30)	**52.991**	38.962	3.00E + 05	0	57.979	48.81	3.00E + 05	0
10(50)	110.71	60.034	5.00E + 05	0	110.28	62.968	5.00E + 05	0
11(10)	3.8435	2.491	91,433	23,930	3.7648	2.8672	**89,520**	27,470
11(30)	**19.651**	9.1953	3.00E + 05	0	24.223	9.253	3.00E + 05	0
11(50)	44.635	14.737	5.00E + 05	0	43.44	18.5	5.00E + 05	0
12(10)	254.83	564.94	73,370	32,129	220.08	561.14	63,337	29,948
12(30)	5033.1	5266.4	3.00E + 05	0	**3296.3**	5065	2.9662e + 05	13,275
12(50)	**25,332**	17,809	5.00E + 05	0	26,310	27,152	5.00E + 05	0
13(10)	**0.43733**	0.15909	1.00E + 05	0	0.50733	0.16607	1.00E + 05	0
13(30)	2.216	0.90388	3.00E + 05	0	2.0943	0.48971	3.00E + 05	0
13(50)	4.7787	2.0317	5.00E + 05	0	6.6917	4.4423	5.00E + 05	0
14(10)	**3.1147**	0.6036	1.00E + 05	0	3.1803	0.50128	1.00E + 05	0
14(30)	**12.096**	1.2419	3.00E + 05	0	12.492	0.99427	3.00E + 05	0
14(50)	**21.387**	1.7448	5.00E + 05	0	21.534	1.2894	5.00E + 05	0
15(10)	225.03	195.33	83,517	27,458	230.52	221.47	78,130	27,953
15(30)	307.57	116.67	3.00E + 05	0	**284.64**	126.64	2.9793e + 05	11,338
15(50)	298.08	97.404	5.00E + 05	0	**283.46**	95.071	5.00E + 05	0
16(10)	125.43	26.513	1.00E + 05	0	**121.64**	24.171	1.00E + 05	0
16(30)	**126.52**	115.06	3.00E + 05	0	181.14	161.42	3.00E + 05	0
16(50)	171.22	173.72	5.00E + 05	0	156.36	107.54	5.00E + 05	0
17(10)	120.71	22.704	1.00E + 05	0	**119.64**	24.8	1.00E + 05	0
17(30)	169.03	163.13	3.00E + 05	0	228.75	178.19	3.00E + 05	0
17(50)	226.73	138.82	5.00E + 05	0	150.06	130.81	5.00E + 05	0

The next experiment focused on investigating the impact of the availability of *a priori* knowledge. The two increasing strategies which make use of *a priori* knowledge, namely LIHH1 and EIHH1, were compared to the increasing strategies which do not make use of any *a priori* knowledge, namely LIHH2 and EIHH2. The results of this comparison are provided in Tables 4 and 6. From the results it is clear that the availability of *a priori* knowledge of constituent algorithm performance on the benchmark problem set under consideration did have significant advantages. A dramatic performance improvement was obtained for most problems by LIHH1 and EIHH1 when compared to LIHH2 and EIHH2. The exception is problems nine and 12, where the no *a priori* strategies performed better. Unfortunately, this knowledge is not always readily available or can be time consuming to obtain.

Table 5
Hypotheses analysis of alternative heuristic space diversity control mechanisms.

	HMHH	LDHH	EDHH	LIHH2	EIHH2	Total
HMHH	NA	5-31-15	8-29-14	4-31-16	3-26-22	20-117-67
LDHH	15-31-5	NA	8-38-5	9-28-14	5-27-19	37-124-43
EDHH	14-29-8	5-38-8	NA	9-24-18	7-22-22	35-113-56
LIHH2	16-31-4	14-28-9	18-24-9	NA	1-42-8	49-125-30
EIHH2	22-26-3	19-27-5	2-22-7	8-42-1	NA	71-117-16

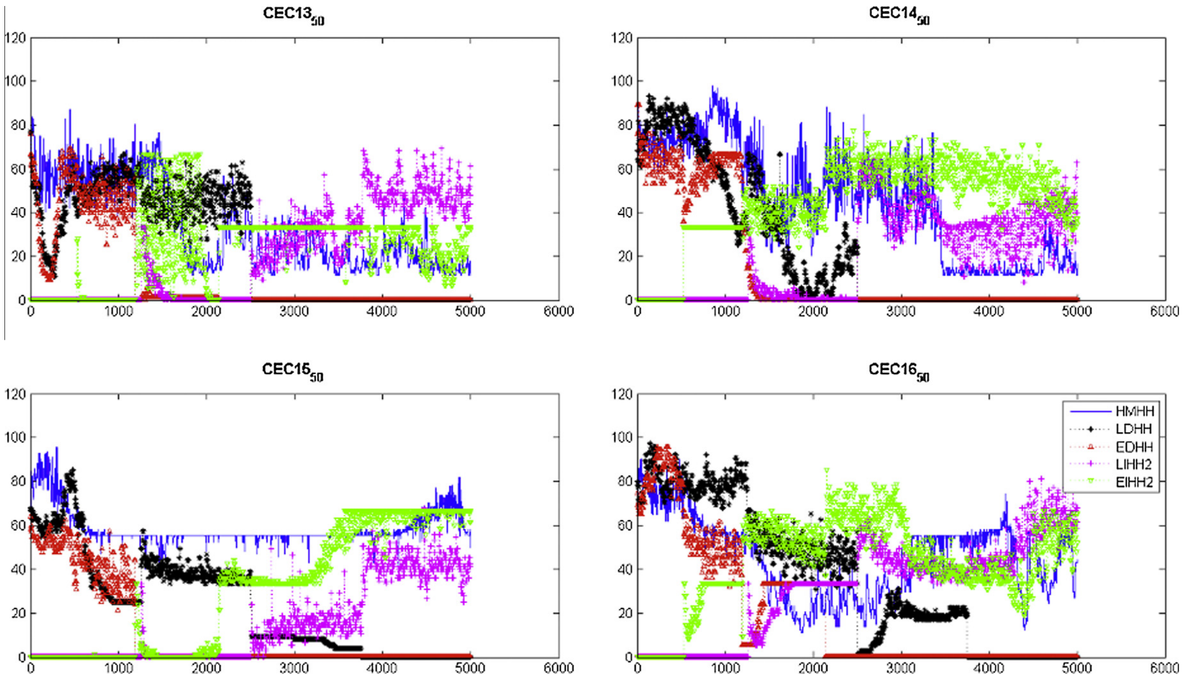

Fig. 3. Graphs of heuristic space diversity (*y*-axes) versus iterations (*x*-axes) on a selected set of problems from the CEC 2005 benchmark problems in 50 dimensions.

In an attempt to further verify the performance of the HSD management strategies, the best performing solution diversity management strategy from Table 5, EIHH2, were also compared under similar conditions to PAP [31]. PAP was found in a previous study [20] to be one of the better performing multi-method algorithms currently available. Finally, the best performing constituent algorithm (CMAES) [1], was also added for comparison purposes. The results are recorded in Table 8. The number of "Mann–Whitney U wins-draws-losses" obtained by EIHH2 when compared to PAP and CMAES is recorded Q3 in Table 9 (see Table 7).

Table 9 shows that both of the HMHH algorithms outperformed PAP in a number of cases. EIHH1 performed better than PAP 33 times out of 51 instances (65% of the time) and EIHH2 performed better 13 times out of 51 instances. EIHH2 did not perform quite as well when compared to CMAES, only performing better 13 times out of 51 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. A more detailed inspection of the results, indicated that the performance of EIHH2 and PAP improved as the complexity of the problems increased. EIHH2 performed especially well on multi-modal problems of smaller dimensions. EIHH1 performed significantly better against CMAES than EIHH2. In this comparison, 13 wins, 32 draws and 6 losses were recorded between EIHH1 and CMAES. This implies that the cost of obtaining *a priori* knowledge should be weighed up against the possible performance benefits when a new optimization problem is addressed. Furthermore, it is promising to note that, for a number of problems, even if the best performing constituent algorithm is selected by chance for solving the problem, the addition of further constituent algorithms will complement the first algorithm, leading to improved performance. In other words, the application of a hyper-heuristic with *a priori* knowledge is valuable even when the best performing constituent algorithm is known.

Table 6
Results of the investigation into alternative heuristic space diversity control mechanisms: LIHH1 (linear increasing heuristic space diversity with *a priori* knowledge) and EIHH1 (exponentially increasing heuristic space diversity with *a priori* knowledge).

Prob (Dims)	LIHH1				EIHH1			
	FFV		# FEs		FFV		# FEs	
	μ	σ	μ	σ	μ	σ	μ	σ
1(10)	1.00E − 06	0	8623.3	244.5	1.00E − 06	0	**8563.3**	277.28
1(30)	1.00E − 06	0	19,253	599.27	1.00E − 06	0	**19,193**	549.57
1(50)	1.00E − 06	0	26,793	739.96	1.00E − 06	0	**26,753**	670.94
2(10)	1.00E − 06	0	9173.3	294.7	1.00E − 06	0	**9106.7**	398.21
2(30)	1.00E − 06	0	26,760	766.36	1.00E − 06	0	**26,410**	806.59
2(50)	1.00E − 06	0	**52,543**	736.57	1.00E − 06	0	52,723	903.89
3(10)	1.00E − 06	0	13,350	483.34	1.00E − 06	0	**13,277**	397.13
3(30)	1.00E − 06	0	**61,567**	1265.3	1.00E − 06	0	61,603	1201.9
3(50)	1.00E − 06	0	**1.5574e + 05**	2671.5	1.00E − 06	0	1.6336e + 05	6549
4(10)	1.00E − 06	0	**9510**	347.75	1.00E − 06	0	9573.3	374.1
4(30)	1.00E − 06	0	29,330	927.79	1.00E − 06	0	**29,267**	986.58
4(50)	1.00E − 06	0	59,993	1287.6	1.00E − 06	0	**59,840**	1043.1
5(10)	1.00E − 06	0	17,520	608.79	1.00E − 06	0	**17,447**	567.35
5(30)	1.00E − 06	0	**2.5721e + 05**	68,420	0.00033427	0.0018254	2.7194e + 05	70,931
5(50)	**0.039999**	0.059073	5.00E + 05	0	190.58	730.17	5.00E + 05	0
6(10)	0.00066667	0.0025371	**18940**	744.91	0.00033333	0.0018257	19,210	716
6(30)	0.133	0.72658	**1.2291e + 05**	46,644	0	0	2.0872e + 05	49,555
6(50)	0.0013333	0.0057135	**3.7927e + 05**	93,024	3.8803	12.716	4.741e + 05	37,361
7(10)	0.001	0.0030513	**7533.3**	339.71	0.00066667	0.0025371	7640	439.91
7(30)	0.00033333	0.0018257	**16,603**	673.38	0	0	16,657	622.94
7(50)	0	0	**24,980**	703.39	0	0	25,013	676.57
8(10)	20.07	0.11418	1.00E + 05	0	20.054	0.096654	1.00E + 05	0
8(30)	20.209	0.15768	3.00E + 05	0	**20.122**	0.10621	3.00E + 05	0
8(50)	21.102	0.14223	5.00E + 05	0	21.123	0.0312	5.00E + 05	0
9(10)	0.495	0.76553	79,563	30,822	0.233	0.41917	58,827	27,886
9(30)	7.8523	4.5941	3.00E + 05	0	3.4547	3.7103	2.7578e + 05	42,827
9(50)	24.662	7.7736	5.00E + 05	0	21.058	6.896	5.00E + 05	0
10(10)	1.6147	1.0913	**88,090**	30,887	1.8793	1.2046	88,200	30,602
10(30)	12.719	5.4544	3.00E + 05	0	**10.553**	4.5767	3.00E + 05	0
10(50)	28.78	10.823	5.00E + 05	0	**26.417**	10.946	5.00E + 05	0
11(10)	1.223	1.2786	74,537	39,566	1.0214	1.149	**66,540**	41,709
11(30)	**8.9018**	3.2821	3.00E + 05	0	10.785	6.4151	3.00E + 05	0
11(50)	**19.799**	4.1018	5.00E + 05	0	21.232	7.9897	5.00E + 05	0
12(10)	174.03	416.19	72,990	41,976	317.66	627.11	63,770	45,137
12(30)	5850.1	3788.7	2.9121e + 05	48,127	4822.2	4782.6	3.00E + 05	0
12(50)	26,343	16,432	5.00E + 05	0	35,970	22,535	5.00E + 05	0
13(10)	0.643	0.21991	1.00E + 05	0	0.44167	0.14283	1.00E + 05	0
13(30)	2.4147	0.67678	3.00E + 05	0	**1.8363**	0.55461	3.00E + 05	0
13(50)	4.597	0.67692	5.00E + 05	0	**4.141**	0.82065	5.00E + 05	0
14(10)	2.8263	0.6434	1.00E + 05	0	**2.6953**	0.37906	1.00E + 05	0
14(30)	**10.059**	0.82305	3.00E + 05	0	10.288	0.96526	3.00E + 05	0
14(50)	**19.482**	0.64356	5.00E + 05	0	19.92	0.74443	5.00E + 05	0
15(10)	340.36	102.85	1.00E + 05	0	366.66	88.409	1.00E + 05	0
15(30)	**276.66**	89.763	3.00E + 05	0	**276.66**	104	3.00E + 05	0
15(50)	**253.35**	93.695	5.00E + 05	0	256.66	77.386	5.00E + 05	0
16(10)	**100.73**	12.229	94,410	17,096	100.8	13.241	1.00E + 05	0
16(30)	179.42	167.61	3.00E + 05	0	**120.7**	142.49	3.00E + 05	0
16(50)	**129.03**	143.34	5.00E + 05	0	148.7	163.59	5.00E + 05	0
17(10)	97.72	8.1602	1.00E + 05	0	**96.525**	7.7402	1.00E + 05	0
17(30)	183.73	205.24	3.00E + 05	0	190.59	177.28	3.00E + 05	0
17(50)	143.73	142.08	5.00E + 05	0	**78.116**	78.053	5.00E + 05	0

Table 7
Hypothesis analysis of the benefit of *a priori* information.

	LIHH2	EIHH2	Total
LIHH1	30-16-5	33-12-6	63-28-11
EIHH1	30-19-2	31-16-4	62-35-6

Table 8
Benchmarking the best heuristic space diversity control strategy: PAP and CMAES.

Prob (Dims)	PAP				CMAES			
	FFV		# FEs		FFV		# FEs	
	μ	σ	μ	σ	μ	σ	μ	σ
1(10)	1.00E − 06	0	14,043	558.74	1.00E − 06	0	8526.7	302.78
1(30)	1.00E − 06	0	33,533	1916.8	1.00E − 06	0	19,110	447.48
1(50)	1.00E − 06	0	53,867	5146.1	1.00E − 06	0	26,930	726.42
2(10)	1.00E − 06	0	18,380	925.65	1.00E − 06	0	9156.7	286.1
2(30)	1.00E − 06	0	88,357	2011.3	1.00E − 06	0	26,783	739.1
2(50)	1.00E − 06	0	2.07E + 05	4396.4	1.00E − 06	0	52,903	869.2
3(10)	1.00E − 06	0	44,810	1673.8	1.00E − 06	0	13,320	379.11
3(30)	1.00E − 06	0	2.84E + 05	5046.4	1.00E − 06	0	61,173	1387.4
3(50)	821.66	829.47	5.00E + 05	0	1.00E − 06	0	1.566e + 05	2244.2
4(10)	1.00E − 06	0	19,440	967.26	1.00E − 06	0	9590	283.27
4(30)	217.08	298.5	2.70E + 05	58,036	1.00E − 06	0	29,357	570.35
4(50)	8594.9	3414.3	5.00E + 05	0	1.00E − 06	0	59,607	998.25
5(10)	1.00E − 06	0	36,600	1586.1	1.00E − 06	0	17,433	546.04
5(30)	299.97	628.18	3.00E + 05	0	1.00E − 06	0	1.1465e + 05	3960.1
5(50)	3097.6	1700.8	5.00E + 05	0	1.00E − 06	0	3.4146e + 05	29,588
6(10)	0.0046, 667	0.02556	54057	12,956	0.00066667	0.0025371	18,950	744.52
6(30)	8.3353	14.475	3.00E + 05	0	0.13267	0.72665	1.2018e + 05	37,870
6(50)	29.91	23.177	5.00E + 05	0	0.13267	0.72665	2.8902e + 05	51,830
7(10)	0.0013333	0.0043417	33,847	37,130	1267	4.6252e − 13	1.00E + 05	0
7(30)	0	0	44,627	48,305	4696.3	2.7751e − 12	3.00E + 05	0
7(50)	0	0	99,407	1.36E + 05	6195.3	0	5.00E + 05	0
8(10)	20.065	0.087247	1.00E + 05	0	20.312	0.11271	1.00E + 05	0
8(30)	20.262	0.11242	3.00E + 05	0	20.892	0.17459	3.00E + 05	0
8(50)	20.287	0.10118	5.00E + 05	0	21.127	0.030189	5.00E + 05	0
9(10)	0.00066667	0.0025371	41,123	2768.9	1.9457	1.5105	88,203	30,601
9(30)	0	0	1.25E + 05	38,786	39.564	6.4543	3.00E + 05	0
9(50)	0.429	1.3714	2.22E + 05	1.27E + 05	62.344	7.3313	5.00E + 05	0
10(10)	6.1153	2.0792	1.00E + 05	0	1.647	1.1767	90,947	27,625
10(30)	37.077	13.279	3.00E + 05	0	9.391	3.2817	3.00E + 05	0
10(50)	84.73	34.295	5.00E + 05	0	24.551	7.5998	5.00E + 05	0
11(10)	4.365	1.3033	1.00E + 05	0	1.2989	1.3647	30,310	10,496
11(30)	20.486	3.1698	3.00E + 05	0	9.0255	3.0546	3.00E + 05	0
11(50)	38.929	5.2649	5.00E + 05	0	19.875	5.2923	5.00E + 05	0
12(10)	55.209	283.64	60,013	30,913	1546.1	2735.5	70,053	43,090
12(30)	6125.1	4267.1	3.00E + 05	0	20,324	19,261	3.00E + 05	0
12(50)	15,581	9725.8	5.00E + 05	0	65,826	69,500	5.00E + 05	0
13(10)	0.29867	0.059116	1.00E + 05	0	0.897	0.25323	1.00E + 05	0
13(30)	1.2317	0.13391	3.00E + 05	0	3.179	0.56064	3.00E + 05	0
13(50)	2.299	0.28842	5.00E + 05	0	5.3587	0.83373	5.00E + 05	0
14(10)	3.17	0.2156	1.00E + 05	0	2.5847	0.53024	1.00E + 05	0
14(30)	12.699	0.25204	3.00E + 05	0	10.394	0.8103	3.00E + 05	0
14(50)	22.3	0.32155	5.00E + 05	0	19.45	1.0963	5.00E + 05	0
15(10)	69.4	112.76	82,833	22,674	343.32	97.143	1.00E + 05	0
15(30)	240.01	72.387	3.00E + 05	0	266.27	64.911	3.00E + 05	0
15(50)	237.29	76.203	5.00E + 05	0	245	63.66	5.00E + 05	0
16(10)	104.81	10.258	1.00E + 05	0	96.626	9.3047	1.00E + 05	0
16(30)	75.746	28.662	3.00E + 05	0	130.49	150.98	3.00E + 05	0
16(50)	83.341	47.487	5.00E + 05	0	98.632	126.2	5.00E + 05	0
17(10)	115.09	10.887	1.00E + 05	0	99.584	10.511	1.00E + 05	0
17(30)	105.97	35.358	3.00E + 05	0	175.66	179.47	3.00E + 05	0
17(50)	135.9	26.259	5.00E + 05	0	182.28	166.26	5.00E + 05	0

Table 9
Further hypotheses analysis of the best performing HSD control mechanisms.

	PAP	CMAES
EIHH1	33-9-9	13-32-6
EIHH2	13-22-16	13-8-30

Q1 Please cite this article in press as: J. Grobler et al., Heuristic space diversity control for improved meta-hyper-heuristic performance, Inform. Sci. (2014), http://dx.doi.org/10.1016/j.ins.2014.11.012

6. 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. The exponentially increasing HSD strategy was shown to outperform the decreasing, linearly increasing, and uncontrolled HSD strategies. An analysis into the performance benefits of *a priori* knowledge was also performed through the comparison of two sets of increasing HSD strategies – one set with and one set without *a priori* knowledge. 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 investigating the resulting HSD profiles of popular existing multi-method algorithms such as AMALGAM, PAP and other bandit-based approaches [15] and the subsequent impact these profiles have on algorithm performance. Instead of adjusting HSD at predetermined time intervals, an adaptive strategy could also be employed where HSD is modified based on certain conditions in the environment. Finally, more research into the definition of HSD and the characterization of HSD measures could be useful.

Acknowledgments

The authors would like to thank Dr. F Peng and Dr. X Yao for their generous provision of the population based algorithm portfolio source code.

References

Q4 [1] A. Auger, N. Hansen, A restart cma evolution strategy with increasing population size, in: Proceedings of the 2005 Congress on Evolutionary Computation vol. 2, 2005, pp. 1769–1776.
[2] E.K. Burke, M. Gendreau, M. Hyde, G. Kendall, B. McCollum, G. Ochoa, A.J. Parkes, S. Petrovic, The cross-domain heuristic search challenge–an international research competition, in: Learning and Intelligent Optimization, Springer, 2011, pp. 631–634.
[3] E.K. Burke, M. Hyde, G. Kendall, G. Ochoa, E. Ozcan, R. Qu, Hyper-heuristics: a survey of the state of the art, J. Oper. Res. Soc. 64 (2013) 1695–1724.
[4] E.K. Burke, M. Hyde, G. Kendall, G. Ochoa, E. Özcan, J.R. Woodward, A classification of hyper-heuristic approaches, in: Handbook of Metaheuristics, Springer, 2010, pp. 449–468.
[5] E.K. Burke, G. Kendall, E. Soubeiga, A tabu-search hyperheuristic for timetabling and rostering, J. Heuristics 9 (6) (2003) 451–470.
[6] E.K. Burke, B. McCollum, A. Meisels, S. Petrovic, R. Qu, A graph-based hyper-heuristic for educational timetabling problems, Eur. J. Oper. Res. 176 (1) (2007) 177–192.
[7] A. Caponio, G.L. Cascella, F. Neri, N. Salvatore, M. Sumner, A fast adaptive memetic algorithm for online and offline control design of pmsm drives, IEEE Trans. Syst. Man Cybern. Part B: Cybern. 37 (1) (2007) 28–41.
[8] A. Caponio, F. Neri, V. Tirronen, Super-fit control adaptation in memetic differential evolution frameworks, Soft. Comput. 13 (8-9) (2009) 811–831.
[9] F. Caraffini, F. Neri, G. Iacca, A. Mol, Parallel memetic structures, Inf. Sci. 227 (2013) 60–82.
[10] F. Caraffini, F. Neri, L. Picinali, An analysis on separability for memetic computing automatic design, Inf. Sci. 265 (2014) 1–22.
[11] X. Chen, Y.-S. Ong, M.-H. Lim, K.C. Tan, A multi-facet survey on memetic computation, IEEE Trans. Evol. Comput. 15 (5) (2011) 591–607.
[12] M. Črepinšek, S.-H. Liu, M. Mernik, Exploration and exploitation in evolutionary algorithms: a survey, ACM Comput. Sur. (CSUR) 45 (3) (2013) 35.
[13] A. Engelbrecht, Scalability of a heterogeneous particle swarm optimizer, in: Proceedings of the 2011 IEEE Symposium on Swarm Intelligence (SIS), 2011, pp. 1–8.
[14] L.J. Eshelman, Real-coded genetic algorithms and interval-schemata, Foundations Genet. Algorithms 2 (1993) 187–202.
[15] Á. Fialho, M. Schoenauer, 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, in: Proceedings of the GECCO 2010 Workshop on Black-Box Optimization Benchmarking, 2010, pp. 1535–1542.
[16] J. Grobler, A.P. Engelbrecht, Solution space diversity management in a meta-hyperheuristic framework, in: Proceedings of the 2013 BRICS Congress on Computational Intelligence, 2013, pp. 1–6.
[17] J. Grobler, A.P. Engelbrecht, G. Kendall, V. Yadavalli, Alternative hyper-heuristic strategies for multi-method global optimization, in: Proceedings of the 2010 Congress on Evolutionary Computation, 2010, pp. 1–8.
[18] J. Grobler, A.P. Engelbrecht, G. Kendall, V. Yadavalli, Investigating the impact of alternative evolutionary selection strategies on multi-method global optimization, in: Proceedings of the 2011 Congress on Evolutionary Computation, 2011, pp. 2337–2344.
[19] J. Grobler, A.P. Engelbrecht, G. Kendall, V. Yadavalli, Investigating the use of local search for improving meta-hyper-heuristic performance, in: Proceedings of the 2012 Congress on Evolutionary Computation, 2012, pp. 1–8.
[20] J. Grobler, A.P. Engelbrecht, G. Kendall, V. Yadavalli, Multi-method algorithms: Investigating the entity-to-algorithm allocation problem, in: Proceedings of the 2013 Congress on Evolutionary Computation, 2013, pp. 570–577.
Q5 [21] J. Grobler, A.P. Engelbrecht, G. Kendall, V. Yadavalli, Heuristic space diversity management in a meta-hyperheuristic framework, in: Proceedings of the 2014 Congress on Evolutionary Computation, 2014 (submitted for publication).
[22] R. Hooke, T.A. Jeeves, "direct search" solution of numerical and statistical problems, J. ACM (JACM) 8 (2) (1961) 212–229.
[23] N. Krasnogor, J. Smith, A memetic algorithm with self-adaptive local search: Tsp as a case study, in: Proceedings of the Conference on Genetic and Evolutionary Computation, 2000, pp. 987–994.
[24] M. Mısır, K. Verbeeck, P. De Causmaecker, G. Vanden Berghe, Design and Analysis of an Evolutionary Selection Hyper-Heuristic Framework, Tech. rep., KAHO Sint-Lieven, 2013.
[25] M. Montazeri, M.S. Baghshah, A. Enhesari, Hyper-heuristic algorithm for finding efficient features in diagnose of lung cancer disease, J. Basic Appl. Sci. Res. 3 (10) (2013) 134–140.
[26] J.A. Nelder, R. Mead, A simplex method for function minimization, Comput. J. 7 (4) (1965) 308–313.
[27] G. Ochoa, R. Qu, E.K. Burke, Analyzing the landscape of a graph based hyper-heuristic for timetabling problems, in: Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, 2009, pp. 341–348.
[28] G. Ochoa, J.A. Vázquez-Rodríguez, S. Petrovic, E. Burke, Dispatching rules for production scheduling: a hyper-heuristic landscape analysis, in: Proceedings of the 2009 Congress on Evolutionary Computation, 2009, pp. 1873–1880.
[29] O. Olorunda, A.P. Engelbrecht, An analysis of heterogeneous cooperative algorithms, in: Proceedings of the 2009 Congress on Evolutionary Computation, 2009, pp. 1562–1569.
[30] J.C. Ortiz-Bayliss, H. Terashima-Marın, E. Ozcan, A.J. Parkes, S.E. Conant-Pablos, Exploring heuristic interactions in constraint satisfaction problems: a closer look at the hyper-heuristic space, in: Proceedings of the 2013 Congress on Evolutionary Computation, 2013 (in press).

[31] F. Peng, K. Tang, G. Chen, X. Yao, Population-based algorithm portfolios for numerical optimization, IEEE Trans. Evol. Comput. 14 (5) (2010) 782–800.
[32] R. Qu, E.K. Burke, Hybridizations within a graph-based hyper-heuristic framework for university timetabling problems, J. Oper. Res. Soc. 60 (9) (2008) 1273–1285.
[33] A. Ratnaweera, H. Watson, S.K. Halgamuge, Particle swarm optimizer with time varying acceleration coefficients, in: Proceedings of the International Conference on Soft Computing and Intelligent Systems, 2002.
[34] Z. Ren, H. Jiang, J. Xuan, Z. Luo, Hyper-heuristics with low level parameter adaptation, Evolut. Comput. 20 (2) (2012) 189–227.
[35] N.R. Sabar, M. Ayob, G. Kendall, R. Qu, Grammatical evolution hyper-heuristic for combinatorial optimization problems, IEEE Trans. Evol. Comput. 17 (6) (2013) 840–861.
[36] S. Salcedo-Sanz, J. Matías-Román, S. Jiménez-Fernández, A. Portilla-Figueras, L. Cuadra, An evolutionary-based hyper-heuristic approach for the jawbreaker puzzle, Appl. Intell. (2013) 1–11.
[37] E. Segredo, C. Segura, C. León, Memetic algorithms and hyperheuristics applied to a multiobjectivised two-dimensional packing problem, J. Global Optim. (2013) 1–26.
[38] K. Sim, E. Hart, B. Paechter, Learning to solve bin packing problems with an immune inspired hyper-heuristic, in: Proceedings of the 12th European Conference on Artificial Life, 2013, pp. 856–863.
[39] P.N. Suganthan, N. Hansen, J.J. Liang, K. Deb, Y. Chen, A. Auger, S. Tiwari, Problem Definitions and Evaluation Criteria for the cec 2005 Special Session on Real-Parameter Optimization, Tech. Rep., Nanyang Technological University and Kanpur Genetic Algorithms Laboratory, 2005.
[40] V. Tirronen, F. Neri, T. Kärkkäinen, K. Majava, T. Rossi, An enhanced memetic differential evolution in filter design for defect detection in paper production, Evolut. Comput. 16 (4) (2008) 529–555.
[41] F. Van den Bergh, A. Engelbrecht, A new locally convergent particle swarm optimiser, in: Proceedings of the 2011 International Conference on Systems, Man and Cybernetics, vol. 3, 2002, pp. 6–12.
[42] N. Veerapen, Y. Hamadi, F. Saubion, Using Local Search with Adaptive Operator Selection to Solve the Progressive Party Problem, Tech. Rep., Microsoft Research, 2013.
[43] J.A. Vrugt, B.A. Robinson, J.M. Hyman, Self-adaptive multimethod search for global optimization in real-parameter spaces, IEEE Trans. Evol. Comput. 13 (2) (2009) 243–259.
[44] Z. Yang, K. Tang, X. Yao, Self-adaptive differential evolution with neighborhood search, in: Proceedings of the 2008 Congress on Evolutionary Computation, 2008, pp. 1110–1116.