Solution space diversity management in a meta-hyperheuristic 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 solution space diversity within the context of a meta-hyperheuristic algorithm. The adaptive local search meta-hyperheuristic (ALSHH), which adaptively applies a local search algorithm when the population diversity strays outside a predetermined solution space diversity profile, is proposed. ALSHH was shown to compare favourably with algorithms making use of local search and diversity maintenance strategies applied at constant intervals throughout the optimization run. Good performance is also demonstrated with respect to two other popular multi-method algorithms.**

I. Introduction

Over the last two decades multi-method algorithms have become a popular alternative for solving real world problems. A multi-method algorithm can be described as consisting of one or more entities representing solutions which are evolved over time, a set of available algorithms or operators, referred to as constituent algorithms, and a high level strategy responsible for allocating the entities to the most suitable algorithms for optimization. Apart from the performance improvements that can be obtained by simultaneously using different solution strategies during the optimization process, one of the other main advantages of this class of algorithms is that the combination of solution strategies combined in a single algorithm takes some of the guesswork out of which algorithm to apply to which problem.

Multi-method algorithms have started to appear in various different domains over the last couple of years. Examples include memetic computation [1], algorithm portfolios [2], [3], algorithm ensembles [4]–[7], adaptive operator selection strategies [8]–[10] and hyperheuristics [11]–[15]. Most of the literature focuses on the use of simple domain specific heuristics or operators which form the set of constituent algorithms. Recently, Grobler *et al.* [16] has investigated the use of a set of metaheuristics as constituent algorithms in a hyperheuristic framework. Although meta-heuristics are well known for their robustness and ability to avoid local optima, room for improvement exists with regard to a meta-heuristic's ability to successfully exploit good solutions further [17]. This is where the use of refinement methods can play a positive role in managing the exploration-exploitation trade off in the hyperheuristic context.

The specific detail with regards to the use of a refinement method in this context is not, however, straightforward. As is the case with the design of a traditional memetic algorithm, a number of design issues need to be addressed. These include, amongst others, the number of individuals which need to be refined, the frequency of application, the type, and intensity of local search to be applied. This paper focuses on investigating the frequency of application of local search in a meta-hyperheuristic framework by attempting to control the solution space diversity of the algorithm.

An adaptive solution space diversity (SSD) management strategy is proposed and 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 Adaptive Local Search Hyperheuristic (ALSHH) algorithm which makes use of an adaptive local search and species selection to control SSD within predefined limits. Good performance was also obtained when the algorithm was compared to a previous local search assisted hyperheuristic and the population based algorithm portfolio algorithm [3], which is a well known multi-method algorithm.

Local search has already been considered to improve hyperheuristic performance [19]. However, the cited investigations focused on using different local search heuristics as high level hyperheuristics and the application of local search at each iteration. To the best of the authors' knowledge, this paper describes the first investigation of the use of an adaptive local search in conjunction with meta-heuristic based low-level heuristics in a hyperheuristic framework.

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 SSD control strategies which were evaluated. The results are documented in Section V before the paper is concluded in Section VI.

II. Adaptive memetic algorithms

There are four main types of algorithms that currently dominate the adaptive memetic algorithm literature, namely hyperheuristic based memetic algorithms [27], self-adaptive and coevolutionary memetic algorithms [28], meta-Lamarckian learning [29], and fitness diversity-adaptive memetic algorithms [33]. The algorithms in this paper are mainly inspired by the work on fitness diversity-adaptive algorithms.

A. *Fitness diversity-adaptive algorithms*

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 diversity measure is calculated at each iteration which determines which refinement method is applied.

The first fitness diversity-adaptive algorithm was introduced by Caponio *et al.* [30]. A self-adaptive criterion based on a fitness diversity measure was used to determine when to use either of two local search algorithms: Hooke-Jeeves or Nelder-Mead simplex. Since then a large number of other fitness-diversity based algorithms have been proposed utilizing different types of diversity measures and different algorithms to increase or decrease population diversity [31], [32].

There are, however, a number of issues that have been identified with regards to the use of fitness-diversity adaptive algorithms. A comparative study of different algorithms [33] has shown that algorithm performance is significantly impacted by the choice of diversity measure used and that the best diversity measure is dependent on both the problem features and characteristics of the algorithm framework. Furthermore, a fitness based diversity measure is normally used instead of calculating the diversity of the actual solutions in order to reduce the computational complexity of the algorithms. Depending on the nature of the fitness landscape this could lead to an incorrect indication of population diversity.

B. *Local search and hyperheuristics*

A number of different strategies have already been used in the hyperheuristic literature to exploit the benefits of local search algorithms to improve hyperheuristic algorithm performance. Firstly, perturbative hyperheuristics aim to improve a candidate solution through a process of automatically selecting and applying one of a set of available heuristics to an existing candidate solution [13]. A number of local search strategies have already been used as high-level hyperheuristic strategies. In other words, these hyperheuristics consist of a local search algorithm which manipulates a number of low level algorithms. A detailed review of a large number of perturbative hyperheuristics is provided in [13].

Secondly, local search algorithms can also be incorporated into the set of available low-level heuristics [18]. This option can be considered an intervention in heuristic space diversity, especially when metaheuristics are utilized as low-level heuristics, since a more diverse set of algorithms are made available to the high-level strategy.

Finally, local search can be applied directly to the solution space. A good example of this is Qu and Burke's graph based hyperheuristic framework [19] where a local search algorithm operates directly on the solution space in conjunction with a hyperheuristic strategy which operates in heuristic space.

III. THE HETEROGENEOUS META-HYPERHEURISTIC ALGORITHM

The tabu-search based HMHH algorithm (Figure 1) [24] was used as basis for investigating the management of solution space diversity. The HMHH algorithm [12] 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.

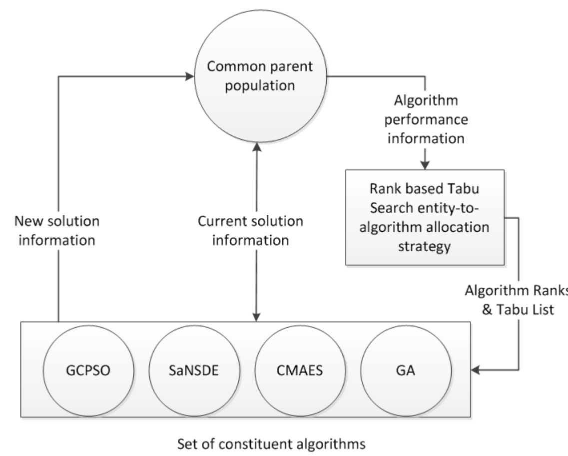

Fig. 1: The heterogeneous meta-hyperheuristic.

The allocation of entities to constituent algorithms is updated on a dynamic basis throughout the optimization run. The idea is that an intelligent algorithm can be evolved which selects the appropriate constituent algorithm at each k^{th} iteration to be applied to each entity within the context of the common parent population, to ensure that the population of entities converge to a high quality solution. The constituent algorithm allocation is maintained for k iterations, while the common parent population is continuously updated with new information and better solutions. Throughout this process, the various constituent algorithms are ranked based on their previous performance as defined by $Q_{\delta m}(t)$ in Algorithm 3 . A tabu list is used to prevent the algorithm from repeatedly using the same poorly performing operators. The highest ranking non-tabu operator is then selected for each entity during re-allocation of entities to algorithms as described in [11] .

This paper 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 [26], [35], and self-adaptive Gaussian mutation [12].
- The guaranteed convergence particle swarm optimization algorithm (GCPSO) [36].
- The self-adaptive (SaNSDE) algorithm of [37].
- The covariance matrix adapting evolutionary strategy algorithm (CMAES) [38].

For all results described in this paper, the Broyden-Fletcher-Goldfarb-Shanno (BFGS) Quasi-Newton method with a cubic line search procedure as implemented in Matlab's optimization toolbox, was utilized.

Algorithm 1: The heterogeneous meta-hyperheuristic.

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

IV. Investigating alternative solutions space diversity management strategies

As is the case with most other optimization algorithms, the ability of a multi-method algorithm to balance exploration and exploitation has a significant impact on its performance. If the algorithm converges too quickly, it is more likely to become stuck in a local optimum. If the algorithm focuses too much on exploring new areas of the search space near the end of the optimization run, time is wasted on exploring the search space which could have been used to further refine promising solutions.

One way to influence this balance is by controlling the solution space diversity of the algorithm. The algorithms studied in this paper are inspired by the fitness-diversity adaptive local search algorithms of Caponio et al [30]. Here, the effective balance of exploration and exploitation by applying different refinement methods depending on the population diversity, is taken one step further. This paper is based on the assumption that a desirable solution space diversity profile is linearly decreasing from the initial randomly generated population diversity to a significantly small value over the maximum allowable algorithm iterations, when the algorithm has converged on one or two promising solutions (Figure 2).

A linearly decreasing upper, UB_{div}, and a lower bound, LB_{div}, is defined for solution space diversity as follows:

$$UB_{div}(1) = SSD(1) + \alpha SSD(1) \quad (1)$$
$$UB_{div}(I_{max}) = \alpha SSD(1) \quad (2)$$
$$LB_{div}(1) = SSD(1) - \alpha SSD(1) \quad (3)$$
$$LB_{div}(I_{max}) = 0 \quad (4)$$

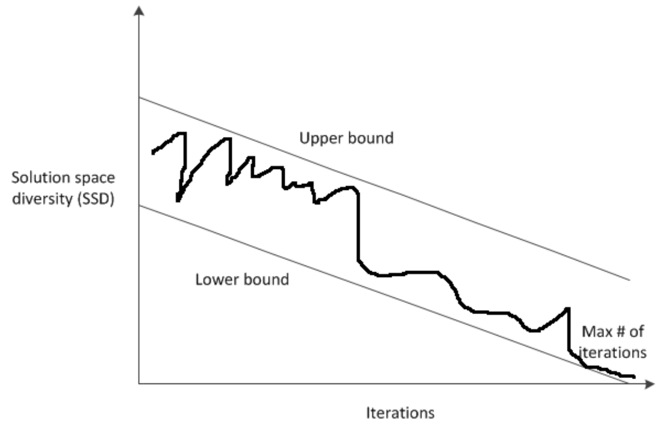

Fig. 2: Upper and lower SSD bounds.

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

$$SSD(t) = \frac{1}{n_s} \sum_{i=1}^{n_s} \sqrt{\sum_{j=1}^{n_x} (x_{ij}(t) - \overline{x}_j(t))^2} \quad (5)$$

where n_s is the number of entities in the population and n_x is the number of dimensions; $x_{ij}(t)$ is the position of the j^{th} dimension of the i^{th} entity at time t.

As soon as the population diversity exceeds the upper bound, a local search method is applied to a randomly selected individual to decrease the overall solution space diversity of the population. The local search is repeatedly applied in consecutive iterations until the population diversity decreases to fall within the acceptable bounds. When the SSD decreases to below the lower diversity bound, then a speciation selection approach [34] is applied to the population at each iteration to ensure that the SSD increases to fall within the defined bounds once more.

Three variations on the above idea are investigated in this paper:

- LSHH-Div applies a local search to a single randomly selected entity from the population at each iteration, and the species selection approach at each n iterations to increase algorithm diversity and prevent the population from collapsing into a relatively small basin of attraction. This constant application of local search and species selection is a good baseline to investigate the impact of adaptive application of the two SSD control mechanisms.
- ALSHH makes use of an adaptive local search algorithm applied every time the upper diversity bound is breached. The diversity enhancing mechanism is applied constantly at each n iterations.
- ALSHH2 makes use of both an adaptive local search and species selection according to the bounds which are exceeded.

V. Empirical evaluation

The various strategies were evaluated on the first 14 problems of the 2005 IEEE Congress of Evolutionary Computation benchmark problem set [40] in both 10 and 30 dimensions. This benchmark problem set enables algorithm performance evaluation on both unimodal and multimodal functions and includes various expanded and hybridized problems, some with noisy fitness functions. The algorithm control parameter values listed in Table I were found to work well for the algorithms under study during previous research by the authors. $m \longrightarrow n$ indicates that the associated parameter is decreased linearly from m to n over 95% of the maximum number of iterations, I_{max}.

TABLE I: HMHH algorithm parameters.

Parameter	**Value used**
Number of entities in common population (n_s)	100
Number of iterations between re-allocation (k)	5
Size of tabu list	3
PSO parameters	
Acceleration constant (c_1)	$2.0 \longrightarrow 0.7$
Acceleration constant (c_2)	$0.7 \longrightarrow 2.0$
Inertia weight (w)	$0.9 \longrightarrow 0.4$
DE parameters	
Probability of reproduction (p_r)	$0.75 \longrightarrow 0.25$
Scaling factor (F)	$0.75 \longrightarrow 0.125$
GA parameters	
Probability of crossover (p_c)	$0.6 \longrightarrow 0.4$
Probability of mutation (p_m)	0.1
Blend crossover parameter (α)	0.5
GA tournament size (N_t)	13
CMAES parameters	As specified in [38].

The results of the solution space diversity management technique comparison are presented in Table IV, where the 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.

Mann-Whitney U tests were used to evaluate the various strategies according to the number of iterations required to obtain the final fitness function value, as well as the quality of the actual fitness function value. Each strategy was compared to each one of the other strategies, and the number of times the first strategy significantly outperformed the second strategy, performed similarly, or is outperformed by the second strategy, was recorded. The results in Table II are subsequently provided in the form: "Number of wins - number of draws - number of losses". To illustrate, (2-14-12) in row 1 column 3, indicates that the LSHH-Div strategy outperformed ALSHH2 twice over the benchmark problem set. Fourteen draws and 12 losses were recorded.

From the results it is clear that the two adaptive ALSHH algorithms outperformed the LSHH-Div which uses the local search and species selection at constant predefined intervals.

TABLE II: Hypotheses analysis of alternative solution space diversity control mechanisms.

	LSHH-Div	**ALSHH**	**ALSHH2**
LSHH-Div	NA	$0-15-13$	$2-14-12$
ALSHH	$13-15-0$	NA	$2-26-0$
ALSHH2	$12-14-2$	$0-26-2$	NA
TOTAL			
LSHH-Div	$2-29-25$		
ALSHH	$15-41-0$		
ALSHH2	$12-40-4$		

This is indicative of the fact that the use of diversity information from the population to drive the need for exploration or exploitation and subsequent application of local search or species selection, makes a positive contribution to the performance of the algorithms. ALSHH also outperforms ALSHH2. Further investigation into the actual population diversity profile obtained as well as the problems for which each of the algorithms perform well, should shed more light on this.

In an attempt to further verify the performance of the strategies, the best performing solution diversity management strategy from the previous analysis, ALSHH, was also compared under similar conditions to LS2HH [16], PAP [3], and the best performing constituent algorithm (CMAES) [38]. The results are recorded in Table IV. In Table III Mann-Whitney U tests were used to compare the performance of each constituent algorithm to the HMHH algorithm. The same "number of wins-draws-losses" format of Table II was used.

TABLE III: Further hypotheses analysis of the ALSHH algorithm.

Algorithm	**ALSHH**
LS2HH	$10-14-4$
PAP	$13-8-7$
CMAES	$2-2-24$
TOTAL	$25-24-35$

From the results in Table III it can be seen that ALSHH outperforms or matches the results obtained by LS2HH and PAP. The algorithm did not however outperform CMAES. This can be attributed to the fact that a larger number of function evaluations is required to solve the harder optimization problem of determining which entities to allocate to which algorithm, as well as solving the actual optimization problem.

VI. Conclusion

This paper investigated the impact of different solution space diversity management strategies on multi-method optimization algorithm performance. Experimental results indicated that the ALSHH algorithm, which adaptively applies the local search while constantly adding diversity to the population, outperformed the other investigated strategies. It was also shown that the ALSHH algorithm compared favourably to two other multi-method algorithms from literature.

References

[1] 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.

TABLE IV: Results of the evaluation of alternative adaptive local search strategies on the 2005 IEEE CEC benchmark problem set.

Prob (Dims)	LSHHD				ALSHH				ALSHH2			
	FFV		# FEs		FFV		# FEs		FFV		# FEs	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
1(10)	1,00E − 06	0	9908.6	552.26	1,00E − 06	0	8756.7	443.87	1,00E − 06	0	8676.7	422.38
1(30)	1,00E − 06	0	40479	2237.2	1,00E − 06	0	31093	2032	1,00E − 06	0	31130	1827.6
2(10)	1,00E − 06	0	14123	686.53	1,00E − 06	0	12793	783.42	1,00E − 06	0	12557	594.04
2(30)	1,00E − 06	0	1.2538e + 05	10043	1,00E − 06	0	93623	7716.7	1,00E − 06	0	94477	9230.4
3(10)	1,00E − 06	0	24538	2900.5	1,00E − 06	0	22027	2143.9	1,00E − 06	0	23280	3246.3
3(30)	17292	16013	3.0012e + 05	0	14447	11998	3,00E + 05	0	11042	15056	2.9581e + 05	22931
4(10)	1,00E − 06	0	15448	1536.1	1,00E − 06	0	13810	1608.5	1,00E − 06	0	13890	1437.3
4(30)	1,00E − 06	0	1.8171e + 05	28722	1,00E − 06	0	1.3473e + 05	22899	1,00E − 06	0	1.4242e + 05	29858
5(10)	1,00E − 06	0	13512	1550.4	1,00E − 06	0	11927	1383.1	1,00E − 06	0	12307	1182
5(30)	970.44	812.38	3.0012e + 05	0	427.57	547.43	3,00E + 05	0	569.81	604.18	3,00E + 05	0
6(10)	0.001	0.0030513	36319	16812	0.0013333	0.0034575	30350	12847	0.001	0.0030513	30077	12782
6(30)	3.8163	13.356	2.6951e + 05	46944	0.664	1.5083	2.3077e + 05	57055	0.79633	1.619	2.3764e + 05	53170
7(10)	0.13967	0.1166	1.0001e + 05	0	0.11633	0.09227	97030	16599	0.145	0.11362	1.0006e + 05	31.784
7(30)	0.013333	0.017682	1.8943e + 05	1.2043e + 05	0.0063333	0.0096431	1.484e + 05	1.2602e + 05	0.015667	0.017555	2.247e + 05	1.1714e + 05
8(10)	20.047	0.076831	1.0001e + 05	9.1539	20.06	0.11424	1.0004e + 05	36.705	20.076	0.13003	1.0003e + 05	39.456
8(30)	20.22	0.15112	3.0012e + 05	0	20.166	0.1233	3.0004e + 05	32.455	20.22	0.19297	3.0005e + 05	33.501
9(10)	0.007	0.0046609	39671	16133	0.007	0.0046609	36603	17726	0.039667	0.17766	39398	18430
9(30)	5.8243	3.3646	3.0012e + 05	0	2.015	1.941	2.9485e + 05	12992	2.7427	2.3152	2.9367e + 05	18807
10(10)	13.235	6.4388	1.0001e + 05	0	13.918	9.1563	1.0006e + 05	27.029	14.17	7.9725	1.0004e + 05	27.468
10(30)	86.726	29.911	3.0012e + 05	0	76.146	22.797	3.0004e + 05	44.716	79.9	29.88	3.0005e + 05	34.709
11(10)	5.8044	2.228	1.0001e + 05	1.8257	5.5514	2.054	1.0004e + 05	37.655	5.9594	1.6531	1.0004e + 05	36.647
11(30)	30.05	3.4757	3.001e + 05	42.238	28.762	3.169	3.0004e + 05	29.998	29.988	3.8819	3.0005e + 05	34.072
12(10)	216.18	549.06	57179	38889	151.99	453.71	46923	41889	398.4	638.91	71757	37250
12(30)	5286.6	8003.9	3.0012e + 05	0	335.25	1.8839	3,00E + 05	0	4313.9	4989.6	2.955e + 05	24667
13(10)	0.42967	0.10217	1.0001e + 05	0	166.52	0.40512	1,00E + 05	10.933	0.44833	0.12887	1.0004e + 05	38.231
13(30)	5.4573	1.7066	3.0012e + 05	0	156.78	0.34684	3,00E + 05	7.303	5.1837	1.6167	3,00E + 05	5.4772
14(10)	3.4127	0.38807	1.0001e + 05	0	3.4127	0.38807	1.0001e + 05	0	3.7497	0.2448	1.0001e + 05	27.207
14(30)	13.238	0.27724	3.0012e + 05	0	13.238	0.27724	3.0012e + 05	0	13.194	0.46734	3,00E + 05	0

Prob (Dims)	LSHH2				PAP				CMAES			
	FFV		# FEs		FFV		# FEs		FFV		# FEs	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
1(10)	1,00E − 06	0	13205	590.59	1,00E − 06	0	13857	630.64	1,00E − 06	0	8526.7	302.78
1(30)	1,00E − 06	0	69832	4304.7	1,00E − 06	0	39190	5945.1	1,00E − 06	0	19110	447.48
2(10)	1,00E − 06	0	15773	1145.2	1,00E − 06	0	18760	1030.4	1,00E − 06	0	9156.7	286.1
2(30)	1,00E − 06	0	1.1821e + 05	19876	1,00E − 06	0	90063	2729.3	1,00E − 06	0	26783	739.1
3(10)	1,00E − 06	0	24383	3428.6	1,00E − 06	0	46067	2040.3	1,00E − 06	0	13320	379.11
3(30)	811.5	2115.4	2.9699e + 05	15519	1,00E − 06	0	2.8784e + 05	5481.5	1,00E − 06	0	61173	1387.4
4(10)	1,00E − 06	0	17808	1383.5	1,00E − 06	0	20483	1424.7	1,00E − 06	0	9590	283.27
4(30)	1,00E − 06	0	2.0331e + 05	34332	4448.8	5562.2	2.8726e + 05	39164	1,00E − 06	0	29357	570.35
5(10)	1,00E − 06	0	19088	1034.1	1,00E − 06	0	25010	2790.5	1,00E − 06	0	17433	546.04
5(30)	299.83	353.17	3.0012e + 05	0	1115.5	1446.5	3,00E + 05	0	1,00E − 06	0	1.1465e + 05	3960.1
6(10)	0.133	0.72658	40452	15190	0.13267	0.72665	50613	10870	0.00066667	0.0025371	18950	744.52
6(30)	3.7687	12.187	2.9229e + 05	15650	0.51867	1.3463	2.7505e + 05	27166	0.13267	0.72665	1.2018e + 05	37870
7(10)	0.131	0.11161	1.0001e + 05	0	0.0026667	0.0058329	62867	35481	1267	4.6252e − 13	1,00E + 05	0
7(30)	0.01	0.013131	2.0306e + 05	1.1305e + 05	0	0	82047	41811	4696.3	2.7751e − 12	3,00E + 05	0
8(10)	20.087	0.13664	1.0001e + 05	0	20.058	0.086279	1,00E + 05	0	20.312	0.11271	1,00E + 05	0
8(30)	20.232	0.12098	3.0012e + 05	0	20.264	0.16296	3,00E + 05	0	20.892	0.17459	3,00E + 05	0
9(10)	0.137	0.42669	46927	25418	0.22033	0.40067	68493	26929	1.9457	1.5105	88203	30601
9(30)	3.2047	1.4938	3.0012e + 05	0	2.6757	1.92	2.8116e + 05	55032	39.564	6.4543	3,00E + 05	0
10(10)	18.475	7.8114	1.0001e + 05	0	12.857	6.2645	1,00E + 05	0	1.647	1.1767	90947	27625
10(30)	61.737	24.805	3.0012e + 05	0	70.555	19.764	3,00E + 05	0	9.391	3.2817	3,00E + 05	0
11(10)	5.484	1.4235	1.0001e + 05	0	4.085	1.2619	1,00E + 05	0	1.2989	1.3647	30310	10496
11(30)	25.349	5.9663	3.0012e + 05	0	20.034	2.6451	3,00E + 05	0	9.0255	3.0546	3,00E + 05	0
12(10)	416.18	645.96	67772	37142	231.53	539.16	71737	28064	1546.1	2735.5	70053	43090
12(30)	4955.8	5179.1	3.0012e + 05	0	3573.5	3310.2	3,00E + 05	0	20324	19261	3,00E + 05	0
13(10)	0.47067	0.15993	1.0001e + 05	0	0.498	0.15873	1,00E + 05	0	0.897	0.25323	1,00E + 05	0
13(30)	3.3303	3.0567	3.0012e + 05	0	1.74	0.42099	3,00E + 05	0	3.179	0.56064	3,00E + 05	0
14(10)	3.623	0.32138	1.0001e + 05	0	3.4067	0.31705	1,00E + 05	0	2.5847	0.53024	1,00E + 05	0
14(30)	13.068	0.43086	3.0012e + 05	0	13.026	0.31914	3,00E + 05	0	10.394	0.8103	3,00E + 05	0

[2] C. P. Gomes and B. Selman, "Algorithm portfolios," *Artificial Intelligence*, vol. 126, no. 1–2, pp. 43–62, 2001.

[3] 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.

[4] T. G. Dietterich, "Ensemble methods in machine learning," *Lecture notes in computer science*, pp. 1–15, 2000.

[5] R. Mallipeddi, P. N. Suganthan, Q. K. Pan, and M. F. Tasgetiren "Differential evolution algorithm with ensemble of parameters and mutation strategies," *Applied Soft Computing*, vol. 11, pp. 1679–1696, 2011.

[6] J. Zhong, M. Shen, J. Zhang, H. Chung, Y. Shi, and Y. Li "A Differential Evolution Algorithm with Dual Populations for Solving Periodic Railway Timetable Scheduling Problem," *IEEE Transactions on Evolutionary Computation*, In press.

[7] J. Tvrdik "Modifications of Differential Evolution with Composite Trial Vector Generation Strategies," *Soft Computing Models in Industrial and Environmental Applications Advances in Intelligent Systems and Computing*, vol. 188, pp. 113-122, 2013.

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

[9] W. Gong, A. Fialho, and Z. Cai, "Adaptive strategy selection in differential evolution," *Proceedings of the 2010 Genetic and Evolutionary Computation Conference*, 2010.

[10] A. Fialho, R. Ros, M. Schoenauer, and M. Sebag, "Comparison-based Adaptive Strategy Selection in Differential Evolution," *Proceedings of the 2011 Genetic and Evolutionary Computation Conference*, 2011.

[11] 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.

[12] 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.

[13] E. K. Burke, M. Hyde, G. Kendall, G. Ochoa, E. Ozcan, and R. Qu, "Hyper-heuristics: A survey of the state of the art," tech. rep., University of Nottingham, 2010.

[14] E. K. Burke, M. Hyde, G. Kendall, G. Ochoa, E. Ozcan, and J. R. Woodward, "A Classification of Hyper-heuristic Approaches," *International Series in Operations Research and Management Science*, In M. Gendreau and J-Y Potvin (Eds.), Springer (in press).

[15] K. A. Dowsland, E. Soubeiga, and E. K. Burke, "A simulated annealing based hyperheuristic for determining shipper sizes for storage and transportation," *European Journal of Operational Research*, vol. 179, pp. 759–774, 2007.

[16] 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.

[17] N. Krasnogor and J. Smith, "A tutorial for competent memetic algorithms: Model, taxonomy and design issues," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 5, pp. 474–488, 2005.

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

[19] 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.

[20] P. Moscato, "On evolution, search, optimization, genetical algorithms and martial arts: Toward memetic algorithms," tech. rep., California Institute of Technology, 1989.

[21] W. E. Hart, N. Krasnogor, and J. E. Smith, "Recent Advances in Memetic Algorithms," *Springer-Verlag*, 2005.

[22] D. Sudholt, "Local search in evolutionary algorithms: The impact of the local search frequency," *Algorithms and Computation (Lecture Notes in Computer Science)*, pp. 359–368, 2006.

[23] D. Whitley, V. S. Gordon, and K. Mathias, "Lamarckian evolution, the baldwin effect and function optimization," *PPSN III Proceedings of the International Conference on Evolutionary Computation*, pp. 6–15, 1994.

[24] 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.

[25] A. K. Qin and P. N. Suganthan,"Self-adaptive differential evolution algorithm for numerical optimization," *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, pp. 1785–1791, 2005.

[26] 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.

[27] A.V. Kononova, D.B. Ingham, M. Pourkashanian, 2008, "Simple scheduled memetic algorithm for inverse problems in higher dimensions: application to chemical kinetics," in: [229], pp. 3906-3913.

[28] N. Krasnogor, J. Smith, "A tutorial for competent memetic algorithms: model, taxonomy, and design issues," *IEEE Transactions on Evolutionary Computation*, vol. 9, pp. 474-488, 2005.

[29] Y.S. Ong, A.J. Keane, "Meta-lamarkian learning in memetic algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 2, pp. 99-110, 2004.

[30] 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*, vol. 37, no. 1, pp. 28-41, 2007.

[31] 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*, vol. 16, pp. 529-555, 2008.

[32] A. Caponio, F. Neri, V. Tirronen, "Super-fit control adaptation in memetic differential evolution frameworks," *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, vol. 13, no. 8, pp. 811-831, 2009.

[33] F. Neri, V. Tirronen, T. Krkkinen, T. Rossi, "Fitness diversity based adaptation in multimeme algorithms: A comparative study," in *'Proceedings of the 2007 IEEE Congress on Evolutionary Computation*, pp. 2374-2381, 2007.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] D. Hadka and P. Reed "Borg: An auto-adaptive Many-Objective Evolutionary Computing Framework," *Evolutionary Computation*, 2012.

[40] 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.