On Solving Hierarchical Problems with Top Down Control

Susan Khor
Concordia University, Canada H3G 2W1
slc_khor@cse.concordia.ca

ABSTRACT

We review recent work on the Hierarchical-If-And-Only-If problem and present a new hierarchical problem, HIFF-M that does not fit with previous explanations for evolutionary difficulty on hierarchical problems decomposed by levels for RMHC2. RMHC2 is a hill climbing algorithm augmented with a multi-level selection scheme. When used with the "ideal" sieve for a problem, as is done in this paper, RMHC2 exerts top-down control on the evolutionary dynamics, in the sense that adaptation of higher levels are given priority over adaptation of lower levels, and creates stabilizing selection pressure with potential to increase evolvability. Through HIFF-M, we discovered that the summary statistic, Fitness Distance Correlation by level, is not a reliable indicator of when a hierarchical problem is solvable by RMHC2, and that the two properties proposed to explain search easiness for RMHC2 are inadequate. Our investigation of this anomaly led us to propose an additional property for hierarchical evolution difficulty under RMHC2: inter-level conflict. We also discuss how hierarchical control can be subverted through the information transfer capacity of the transposition operation.

Categories and Subject Descriptors

I.2.8 [Artificial Intelligence]: Problem Solving, Control Methods and Search

General Terms: Algorithms, Design

Keywords: hierarchical control, level decomposition, transposition, hierarchical test problems

1. BACKGROUND

A problem has hierarchical structure when its variables are organized in some way to form levels and there is a total ordering on the set of levels. Every variable belongs or contributes to one or more levels. Variables of a level are further partitioned into non-overlapping modules. Every variable of level λ belongs to exactly one module of level λ. Modules may be nested or non-nested. Module A is nested within module B if all variables contributing to module A also contribute to module B.

For all problems discussed in this paper:

(i) The binary alphabet is used. This creates, due to constraints of the problems, two global optima, the all ones (1…1) and all zeroes (0…0) genotypes.

(ii) The problem size, N is 2^n where n is a positive integer. n defines the number of levels for a problem. Level n is the highest level and level 1 is the lowest level.

(iii) The optimal (aggregate) fitness is N-1 and the optimal phenotype has the following general form: $\langle 2^0, 2^1, \ldots, 2^{n-1}\rangle$. A phenotype is a sequence of level fitness values in descending level order. Level n is leftmost, level 1 is rightmost.

(iv) There are $N/2^\lambda$ modules at level λ.

(v) The dependency matrix, M, defines the structure and strength of interdependencies between variables. M is a N square, symmetric matrix with zero valued diagonal entries. Entries closer to the diagonal of M belong to lower levels.

The *Hierarchical-If-And-Only-If* (HIFF) problem was introduced in [11] and is explained further in [10]. Briefly, variables in a HIFF problem are organized into a hierarchy of nested modules (Fig. 1). In the discrete version of HIFF which we refer to as *HIFF-D*, fitness points are awarded to a module only when it is optimal. A HIFF module is optimal when all its variables have the same values or match. In the continuous version of HIFF which we refer to as *HIFF-C*, fitness points are awarded for each pair of related variables that have matching values. To simplify calculation, fitness of a HIFF-C module is given by $(p \times q) + (1 - p) \times (1 - q)$ where p and q are the proportion of ones in the first and second halves of the module respectively. Further, a relation at a lower level is worth more than a relation at a higher level. This is reflected in the dependency matrix, M, for HIFF (Fig. 2). Entries closer to the diagonal of M have larger values.

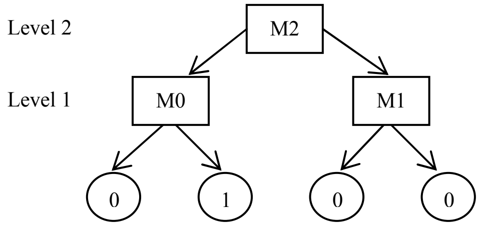

Figure 1. A perfect binary tree showing the decomposition of a genotype of length four.

	0	1	2	3
0	0	1/2	1/8	1/8
1	1/2	0	1/8	1/8
2	1/8	1/8	0	1/2
3	1/8	1/8	1/2	0

Figure 2. The dependency matrix M for HIFF, N=4. The 1/2 entries belong to level 1 and the 1/8 entries belong to level 2.

In genotype 0100, genes 2 and 3 match. Under HIFF-C, this match is worth $M_{2\,3} + M_{3\,2} = 0.5 + 0.5 = 1.0$ fitness point.

GECCO'07, July 7–11, 2007, London, England, United Kingdom.
Copyright 2007 ACM 987-1-59593-698-1/07/0007…$5.00.

Because genes 2 and 3 form a module (M1), 1 fitness point is also awarded under HIFF-D. The situation is different for module M2 at level 2. Under HIFF-D, M2 which consists of 0100 is not optimal, so its fitness is 0. Under HIFF-C, p = 0.5 and q = 1.0, giving a fitness value of 0.5 for M2. Fitness of a level is the sum of fitness of all modules at that level. Fitness of a genotype or its aggregate fitness is the sum of fitness of all levels.

The *Hierarchical-If-And-Only-If with Inter-level Interdependency* (*HIFF-II*) problem was introduced in [3] as a hierarchical problem that is more difficult for RMHC2 but less difficult for genetic algorithms. HIFF-II is similar to HIFF-C in decomposition (Fig. 1), its modules are also nested. But its dependency matrix M has non-diagonal zero entries (Fig. 3). Hence every gene is not *directly* related to every other gene. Some constraints or relationships are indirect or implied.

Fitness of a HIFF-II module at level λ is calculated by doing a pair-wise comparison of genes in the first half of a module with genes in the second half of a module. Specifically:
(i) gene i is compared with gene $2^{\lambda-1} + i$ of a module for $i = 0, \ldots, 2^{\lambda-1} -1$,
(ii) one point is awarded if the two genes match, and
(iii) the number of points is divided by half the module size, $2^{\lambda-1}$.

	0	1	2	3
0	0	1/2	1/4	0
1	1/2	0	0	1/4
2	1/4	0	0	1/2
3	0	1/4	1/2	0

Figure 3. The dependency matrix M for HIFF-II, N=4.

RMHC2 [5] augments the random mutation hill climber (RMHC) introduced in [1] with a *phenotype* and a *multi-level selection scheme*. RMHC is a hill climbing algorithm and works on a single genotype. RMHC mutates a number of random bits in each iteration and keeps the mutation if it produces a fitter variant. The decision whether to keep a mutation, that is select a variant over its parent, is made by comparing aggregate fitness values. In contrast, RMHC2 compares phenotypes using a multi-level selection scheme. A phenotype is a sequence of level fitness values arranged from left to right in descending level order. For example, the HIFF-II aggregate fitness of genotype 0100 is 1.5 and its phenotype is ⟨0.5, 1.0⟩ (Table 1).

The multi-level selection scheme for RMHC2 prioritizes the adaptation of levels. Under this prioritization a high priority level may progress at the expense of a low priority level and the progress of a low priority level may be hindered by the regress of a high priority level. Priorities are assigned by the *sieve* component of the multi-level selection scheme. A sieve is a sequence of levels, levels appearing on the left trumps levels on the right. The "ideal" sieve is one with levels arranged in descending order from left to right. The "ideal" sieve for a hierarchical problem with n levels is ⟨n, n-1, …, 1⟩. "Ideal" RMHC2 uses the "ideal" sieve. In this paper, only the "ideal" sieve for problems are used, and RMHC2 refers to "ideal" RMHC2.

Suppose we are using RMHC2 with 1 bit mutation, the problem is HIFF-II and the current genotype is 0100. Figure 4 shows the four reachable genotype variants and their phenotypes. According to the multi-level selection scheme for RMHC2, only genotypes 0000 and 0101 are acceptable variants.

RMHC2 could be viewed as adopting a multi-objectivization [6] approach. Each level is an objective function, a sieve ranks these multiple objectives and the selection scheme allows for both hard and soft constraints.

Table 1. Genotypes and their fitness values, N = 4.

gnum	genotype	HIFF-D			HIFF-C			HIFF-II			*hd*
		L2	*L1*	*f*	*L2*	*L1*	*f*	*L2*	*L1*	*f*	
0	0000	**1**	**2**	**3**	**1.0**	**2.0**	**3.0**	**1.0**	**2.0**	**3.0**	0
1	0001	0	1	1	0.5	1.0	1.5	0.5	1.0	1.5	1
2	0010	0	1	1	0.5	1.0	1.5	0.5	1.0	1.5	1
3	0011	0	**2**	2	0.0	**2.0**	2.0	0.0	**2.0**	2.0	2
4	0100	0	1	1	0.5	1.0	1.5	0.5	1.0	1.5	1
5	0101	0	0	0	0.5	0.0	0.5	**1.0**	0.0	1.0	2
6	0110	0	0	0	0.5	0.0	0.5	0.0	0.0	0.0	2
7	0111	0	1	1	0.5	1.0	1.5	0.5	1.0	1.5	1
8	1000	0	1	1	0.5	1.0	1.5	0.5	1.0	1.5	1
9	1001	0	0	0	0.5	0.0	0.5	0.0	0.0	0.0	2
10	1010	0	0	0	0.5	0.0	0.5	**1.0**	0.0	1.0	2
11	1011	0	1	1	0.5	1.0	1.5	0.5	1.0	1.5	1
12	1100	0	**2**	2	0.0	**2.0**	2.0	0.0	**2.0**	2.0	2
13	1101	0	1	1	0.5	1.0	1.5	0.5	1.0	1.5	1
14	1110	0	1	1	0.5	1.0	1.5	0.5	1.0	1.5	1
15	1111	**1**	**2**	**3**	**1.0**	**2.0**	**3.0**	**1.0**	**2.0**	**3.0**	0

gnum is genotype number, the integer equivalent of the binary string; *Lλ* is the fitness for level λ; *f* is aggregate fitness; *hd* is Hamming distance to a nearest global optimum. Optimal fitness values are bolded.

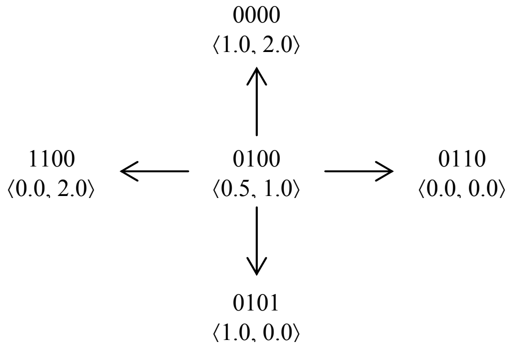

Figure 4.

The number of bits mutated by random bit assignment in each RMHC2 iteration depends on the mutation rate parameter, P_m . We use P_m = 0.0625 and N = 128. This means 1 to 8 (P_m × N) random bits are mutated each time. For all experiments, we do 30 runs with a different random number seed each time, and the maximum evaluations per run is 1 million.

Paper Overview

In this paper we present another hierarchical problem, HIFF-M (M for minimal). Modules are not nested in HIFF-M. The variables in HIFF-M are minimally connected and the degree distribution is highly rightly skewed, there are fewer nodes with more edges and more nodes with fewer edges. The dependency matrix for HIFF-M has even more non-diagonal zero entries than the dependency matrix for HIFF-II, and its non-zero entries all have the same value.

What is interesting about this new problem is firstly the unexpected success of RMHC2 on HIFF-M. This result was unexpected because:
(i) its per-level FDC values has the same pattern as that of HIFF-D which indicates greater search difficulty at higher levels,
(ii) like HIFF-II, HIFF-M has inter-level interdependency by our definition in section 2, and
(iii) RMHC2 was not successful on HIFF-D or HIFF-II.

These three factors led us to expect that RMHC2 should not work on HIFF-M. But the empirical results proved us wrong (section 3). Therefore something is clearly missing from our previous explanation for the performance of RMHC2 on a hierarchical problem. To remedy this situation, we propose that a missing condition is *inter-level conflict* (section 4).

Secondly, our experiments (not reported here) thus far indicate that HIFF-M (N = 128) is not easily solved by RMHC1 even when macro-mutation with $P_m = 0.5$ is used. In a macro-mutation, the mutated bits are consecutive and periodic boundary is used. The steps for RMHC1 are identical to RMHC2 except RMHC1 does not create phenotypes or use the multi-level selection scheme. RMHC1 compares aggregate fitness. We leave the investigation of this finding to the future.

2. THE STORY SO FAR

Previous work made use of the Fitness Distance Correlation (FDC) [2] summary statistic to explain in a quantifiable way, differences in performance of RMHC2 on HIFF-D [4], HIFF-C [5] and HIFF-II [3] (Table 2). FDC was measured for each level of a problem using level fitness values and Hamming distance to a closer global optimum (Tables 3).

Table 2. Results for RMHC2, N = 128.

Algorithm	HIFF-D	HIFF-C	HIFF-II
Times found	0/30	30/30	0/30
Avg. (std. dev.) evals	-	1,855 (545)	-

The per-level FDC values were used to contrast level search difficulty in the three problems and to explain in part why RMHC2 works on HIFF-C but not on HIFF-D or HIFF-II. RMHC2 works by prioritizing the adaptation of levels: a high priority level may progress at the expense of a low priority level or the progress of a low priority level may be hindered by the regress of a high priority level. The per-level FDC values for HIFF-C indicate that search is easier at higher levels for HIFF-C. Therefore, it follows that RMHC2 is successful on HIFF-C. Search is equally difficult at all levels in HIFF-II. Thus, unlike HIFF-C, there is no easier route for RMHC2 to take on HIFF-II. In HIFF-D, the slightly easier levels are in the bottom half of the hierarchy and are sandwiched between more difficult levels. However lower levels in HIFF-D lack the global view higher levels have which is crucial to finding a globally optimal solution.

Search easiness at higher levels for HIFF-C was attributed to two factors: (i) *inter-level interdependency* and (ii) *aggregate-ability*. Specifically, HIFF-D and HIFF-C have no inter-level interdependency, and HIFF-C and HIFF-II have aggregate-ability. In contrast, HIFF-II has inter-level interdependency and HIFF-D does not have aggregate-ability.

Table 4.

Property	HIFF-D	HIFF-C	HIFF-II
Inter-level interdependency	No	No	Yes
Aggregate-ability	No	Yes	Yes

Inter-level Interdependency

A hierarchical problem has inter-level interdependency when no single level can produce only globally optimal solutions on its own. The evolutionary algorithm must then focus on adapting more than one level concurrently if a globally optimal solution is to be found. A comparison between sets of optimal genotypes by level for HIFF and for HIFF-II illustrates this point (Table 5).

The optimal genotypes for a level satisfy all the constraints for that level and might satisfy or violate some of the constraints of other levels. A constraint is defined by a relationship between two variables. For example, requiring the first two variables to have the same value is a level 1 constraint for HIFF, and requiring the first and third variables to have the same value is a level 2 constraint for HIFF-II.

In Table 5, we see that prioritizing the adaptation of level 2 for HIFF-II may produce genotypes other than a globally optimal genotype. However, only globally optimal genotypes are produced by level 2 in the HIFF problem. This is because in HIFF, level 2 has all the constraints of the problem, while in HIFF-II, the constraints are spread out amongst the levels. We have verified experimentally that it is possible, and just as efficient, to solve HIFF-C with RMHC2 using just the highest level fitness values as feedback.

Table 3. Fitness Distance Correlation by Level

Level	HIFF-D		HIFF-C		HIFF-II	
	N=8	N=16	N=8	N=16	N=8	N=16
4	-	-0.0287	-	-0.6770	-	-0.2394
3	-0.2877	-0.2463	-0.6972	-0.4787	-0.3486	-0.2394
2	-0.5403	-0.3789	-0.4930	-0.3385	-0.3486	-0.2394
1	-0.3486	-0.2394	-0.3486	-0.2394	-0.3486	-0.2394
Aggregate	-0.4690	-0.3281	-0.5712	-0.4199	-0.5325	-0.3894

Table 5. Sets of per-level optimal genotypes for N=4

Level	Optimal level fitness	HIFF		HIFF-II	
		Size	Optimal genotypes	Size	Optimal genotypes
2	1	2	{0000, 1111}	4	{0000, 0101, 1010, 1111}
1	2	4	{0000, 0011, 1100, 1111}	4	{0000, 0011, 1100, 1111}

Table 6.

Step	HIFF-D		HIFF-C		HIFF-II	
	Genotype	Phenotype	Genotype	Phenotype	Genotype	Phenotype
t	0100	⟨0, 1⟩	0100	⟨0.5, 1.0⟩	0100	⟨0.5, 1.0⟩
t+1	1100	⟨0, 2⟩	0000	⟨1.0, 2.0⟩	0101	⟨1.0, 0.0⟩

Aggregate-ability

A hierarchical problem has aggregate-ability or is aggregate-able when fitness points are awarded at the level of variables as opposed to modules. When satisfied, aggregate-ability enables state changes at lower levels to influence fitness at higher levels. Under suitable conditions, higher levels can then use this bottom-up influence to constrain lower levels.

HIFF-C and HIFF-II are aggregate-able. HIFF-D is not aggregate-able. A HIFF-D module has zero fitness until its two constituent modules are optimal and form an optimal higher level module. Only when a HIFF-D module is optimal, does its fitness value change from zero to one. As Watson puts it, it is "all or nothing" with HIFF-D [10, p. 121]. This "all or nothing"-ness leads to the possibility of dimensional reduction (reduced coupling between modules) when the evolutionary algorithm manipulates modules properly.

Why RMHC2 is able to solve HIFF-C?

When a hierarchical problem is aggregate-able and does not have inter-level interdependency, such as HIFF-C, the fitness of a module completely and instantaneously reflects the state of its constituent lower level modules. In such a system, and when lower level modules are not misleading in the sense that their optimization lead the system further away from its global goal as is the case with H-Trap [9], the system is poised for evolution under an evolutionary algorithm such as RMHC2 with a suitable mutation rate. RMHC2 uses the fitness information of the highest level to exert stabilizing selection pressure.

Table 6 gives an example of stabilizing selection. For the HIFF-C genotype, there is only one possible move under RMHC2 with 1 bit mutation, to a global optimum (0000). However, there are two possible moves for the HIFF-D and HIFF-II genotypes, one to a global optimum and the other to a local optimum. The local optimum for HIFF-D (1100) is optimal at level 1 while the local optimum for HIFF-II (0101) is optimal at level 2. The behavior of RMHC2 on HIFF-C could be viewed as another form of dimensional reduction, that is reduction of levels pertinent to the evolution of a global solution.

3. HIFF-M

Modules in HIFF-M are not nested and have two variables each (Fig. 5). Modules at level λ are formed from variable pairs ($m \times 2^{\lambda}$, $m \times 2^{\lambda} + 2^{\lambda-1}$) for $m = 0, \ldots, N/2^{\lambda}-1$. If the two variables of a module match, 1 fitness point is awarded. The dependency matrix M for HIFF-M is even sparser (more zero entries) than the M for HIFF-C. Further, all non-zero entries have the same value, 1/2 for any N (Fig. 6).

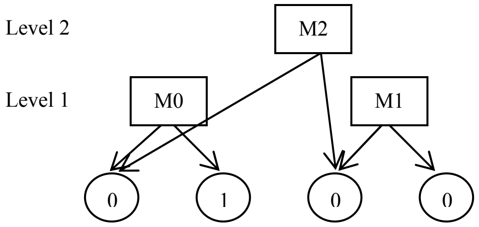

Figure 5. Modules in HIFF-M, N = 4.

	0	1	2	3
0	0	1/2	1/2	0
1	1/2	0	0	0
2	1/2	0	0	1/2
3	0	0	1/2	0

Figure 6. Dependency matrix M for HIFF-M, N = 4.

Table 7. Genotypes and their fitness values, N = 4.

gnum	genotype	HIFF-M			*hd*
		L2	*L1*	*f*	
0	0000	**1**	**2**	**3**	0
1	0001	**1**	1	2	1
2	0010	0	1	1	1
3	0011	0	**2**	2	2
4	0100	**1**	1	2	1
5	0101	**1**	0	1	2
6	0110	0	0	0	2
7	0111	0	1	1	1
8	1000	0	1	1	1
9	1001	0	0	0	2
10	1010	**1**	0	1	2
11	1011	**1**	1	2	1
12	1100	0	**2**	2	2
13	1101	0	1	1	1
14	1110	**1**	1	2	1
15	1111	**1**	**2**	**3**	0

The per-level FDC for HIFF-M approaches 0 higher up the hierarchy (Table 8) giving indication that search is more difficult at higher levels. There is inter-level interdependency in HIFF-M because no single level in HIFF-M produces only globally optimal genotypes (Table 9). Because all modules have only two variables, there is really no distinction between awarding points at the level of variables or at the level of modules for HIFF-M. These three factors, taken together with our previous analysis in section 2, led us to expect that RMHC2 would not succeed on HIFF-M. However, our experiments (Table 11) showed the contrary. RMHC2 found a global solution in every run and did so easily.

Table 8. Fitness Distance Correlation by Level for HIFF-M.

Level	N = 8	N = 16
4	-	-0.0846
3	-0.1743	-0.1197
2	-0.2465	-0.1693
1	-0.3486	-0.2394
Aggregate	-0.4611	-0.3278

Table 9. Sets of per-level optimal genotypes for N=4

Level (λ)	Optimal level fitness	HIFF-M	
		Size	Optimal genotypes
2	1	8	{0*0*, 1*1*}
1	2	4	{0000, 0011, 1100, 1111}

Table 10.

Property	HIFF-M
Inter-level interdependency	Yes
Aggregate-ability	Yes

Table 11. Results for RMHC2, N=128.

Algorithm	Times found	Avg. (std. dev.) Evaluations
HIFF-M	30/30	2,522 (879)
HXOR-M	30/30	2,281 (585)

4. INTER-LEVEL CONFLICT

A hierarchical problem has inter-level conflict when (i) it has inter-level interdependency and (ii) adaptation at a level locks in (with the help of stabilizing selection) bits that prevent other levels from achieving their optimum. The more constraints a level has, the more bits it has to lock in to increase its fitness, leaving fewer degrees of freedom for other lower priority levels.

We have used RMHC2 with the "ideal" sieve throughout this paper. Therefore a measure of inter-level conflict for a problem can be obtained by examining how much higher levels lock in bits that are in conflict with the interest of lower levels. HIFF-II locks in the maximum N bits while HIFF-M locks in progressively more bits at lower levels (Table 12). We say that HIFF-M has a lower degree of inter-level conflict than HIFF-II. Hence, although HIFF-M has inter-level interdependency as does HIFF-II, the higher levels in HIFF-M do not prevent lower levels from progressing as much as the higher levels of HIFF-II do. Further, the bits that get locked in for HIFF-M provide a scaffolding on which to build an optimal solution because they are less conflicting.

5. DISCUSSION

In the case of HIFF-C, aggregate-ability helped all levels to achieve their optimum because there is no inter-level conflict – what is good for the highest level is also good for other levels. Aggregate-ability impeded global optimality in HIFF-II because there is a high degree of inter-level conflict.

Aggregate-ability facilitates information flow from lower level modules to higher level modules. In the presence of a top-down control hierarchy afforded by RMHC2, this information is then used by higher level modules for their own benefit. When there is little to no conflict between levels, aggregate-ability also increases the adaptability of lower level modules. However, aggregate-ability becomes a liability to lower level modules when inter-level conflict is high.

It would be interesting to explore strategies lower level entities could use to subvert hierarchical control in situations where aggregate-ability is a liability. One such strategy is transposition, which proved successful for discrete HIFF [4] and we have verified, also for HIFF-II.

Transposition moves chunks of randomly selected genes from one location to another location within the same genotype. This lateral movement of genes within a genotype can be viewed as an information flow between previously remote genes, or random rewiring of connections between genes in a genotype.

Table 12. Sets of per-level optimal genotypes for N=8.

Level (λ)	Optimal level fitness	HIFF-M		HIFF-II	
		Size	Optimal genotypes	Size	Optimal genotypes
3	1	128	{0*** 0***, 1*** 1***}	16	{0000 0000, 0001 0001, 0010 0010, 0011 0011, 0100 0100, 0101 0101, 0110 0110, 0111 0111, 1000 1000, 1001 1001, 1010 1010, 1011 1011, 1100 1100, 1101 1101, 1110 1110, 1111 1111}
2	2	64	{0*0* 0*0*, 0*0* 1*1*, 1*1* 0*0*, 1*1* 1*1*}	16	{0000 0000, 0000 0101, 0000 1010, 0000 1111, 0101 0000, 0101 0101, 0101 1010, 0101 1111, 1010 0000, 1010 0101, 1010 1010, 1010 1111, 1111 0000, 1111 0101, 1111 1010, 1111 1111}
1	4	16	{0000 0000, 0000 0011, 0000 1100, 0000 1111, 0011 0000, 0011 0011, 0011 1100, 0011 1111, 1100 0000, 1100 0011, 1100 1100, 1100 1111, 1111 0000, 1111 0011, 1111 1100, 1111 1111}	16	{0000 0000, 0000 0011, 0000 1100, 0000 1111, 0011 0000, 0011 0011, 0011 1100, 0011 1111, 1100 0000, 1100 0011, 1100 1100, 1100 1111, 1111 0000, 1111 0011, 1111 1100, 1111 1111}

In our experiment, transposition helped a one dimensional 149 cell synchronous uniform cellular automata using the majority rule with neighbourhood radius r = 1 and initial configuration ρ= 0.5 to correctly identify the majority on the density classification task [8], 26 times out of 30 runs using a different random number seed each time. It is a known fact that some kind of global coordination or information flow between distant sites, for example through generating "particles" [8] or random Boolean networks [7], is required to solve the density classification task. Perhaps transposons are Nature's way of propagating information and creating transient random connections in the genome.

6. WHAT'S NEXT?

Previous work showed that the HIFF-C problem was easily solved by RMHC2. From this result, we learned that there are two algorithmically distinct ways of decomposing a hierarchical problem: by level and by module. When is a hierarchical problem better decomposed by level and solved with a top down approach?

To answer this question, we have compared features of four hierarchical problems and their behavior under RMHC2. From this study, we identified three factors that contribute to difficulty associated with hierarchical evolution by level under top-down control: (i) aggregate-ability, (ii) inter-level interdependency, and (iii) inter-level conflict.

Hierarchical problems are easily solved through level decomposition and top-down control when (i) there is aggregate-ability and no inter-level interdependency, and (ii) if there is inter-level interdependency, inter-level conflict is low. We are working to make these intuitions more precise, and apply to a wider set of problems.

We also found the per-level FDC not reliable as an indication of hierarchical evolution difficulty. This is not too surprising, given the known problems with FDC. Examining the scatter plots for the four hierarchical problems was not enlightening to us.

In the future, we will also explore less restrictive dependency matrices and different versions of the HIFF-M dependency matrix.

7. ACKNOWLEDGEMENTS

Thanks to Peter Grogono for his comments and support of this work through his NSERC grant.

8. REFERENCES

1 Forrest, S. and Mitchell, M. Relative Building-block Fitness and the Building Block Hypothesis. In D. Whitley (editor) *Foundations of Genetic Algorithms* (FOGA) vol. 2, 1993, Morgan Kaufmann.

2 Jones, T. and Forrest, S. Fitness Distance Correlation as a measure of problem difficulty for Genetic Algorithms. In L. Eshelman (Editor), 6th *International Conference on Genetic Algorithms* (ICGA), 1995, pp. 184 – 192, Morgan Kaufmann.

3 Khor, S. HIFF-II: A Hierarchically Decomposable problem with Inter-level Interdependency. *IEEE Symposium on Artificial Life*, 2007. ISBN: 1-4244-0698-6

4 Khor, S. Hill-climbing on discrete HIFF: Exploring the role of DNA Transposition in long-term artificial evolution. In Proceedings of Genetic and Evolutionary Computation Conference (GECCO) , 2007.

5 Khor, S. Rethinking the adaptive capability of accretive evolution on Hierarchically Consistent problems. *IEEE Symposium on Artificial Life*, 2007. ISBN: 1-4244-0698-6

6 Knowles, J. D., Watson, R. A. and Corne, D. W. Reducing local optima in single-objective problems by multi-objectivization. In Conference on Evolutionary Multi-criterion Optimization (EMO), 2001. Springer-Verlag.

7 Mesot, B. and Teuscher, C. Deducing local rules for solving global tasks with random Boolean networks. *Physica D* 211, 2005, pp. 88-106.

8 Mitchell, M. An Introduction to Genetic Algorithms. MIT Press, 1998.

9 Pelikan, M. and Goldberg, D. E. Escaping Hierarchical Traps with Competent Genetic Algorithms. In Genetic and Evolutionary Computation Conference (GECCO), 2001. Morgan Kaufmann.

10 Watson, R. A. Compositional Evolution – The impact of sex, symbiosis and modularity on the gradualist framework of evolution. MIT Press, 2006.

11 Watson, R. A., Hornby, G. S. and Pollack, J. B. Modeling Building-block Interdependency. In Parallel Problem Solving from Nature (PPSN), 1998 vol. V. Springer, Berlin.