Twin Removal in Genetic Algorithms for Protein Structure Prediction using Low Resolution Model

Md Tamjidul Hoque, Madhu Chetty, Andrew Lewis, and Abdul Sattar

Abstract—This paper presents the impact of twins and the measures for their removal from the population of genetic algorithm (GA) when applied to effective conformational searching. It is conclusively shown that a twin removal strategy for a GA provides considerably enhanced performance when investigating solutions to complex *ab initio protein structure prediction* (PSP) problems in low resolution model. Without twin removal, GA crossover and mutation operations can become ineffectual as generations lose their ability to produce significant differences which can lead to the solution stalling. The paper relaxes the definition of chromosomal twins in the removal strategy to not only encompass identical, but also highly-correlated chromosomes within the GA population, with empirical results consistently exhibiting significant improvements solving PSP problems.

Index Terms— Genetic algorithms, twin removal, protein structure prediction, search algorithms, chromosome.

1 INTRODUCTION

THE *ab initio* protein structure prediction (PSP) problem [1] essentially involves investigating the folding relationship of a linear chain of amino acids into a three dimensional (3D) structure based on the properties and appearance pattern of amino acids. The folding relationship can be extremely complex [2], [3], [4], [5], [6], [7], [8], [9], [10], and to unravel this convoluted relation, lattice models such as the *hydrophobic-hydrophilic* (HP) model [11] have been widely applied. In the HP model, the copolymer chains of H (hydrophobic) and P (polar or hydrophilic) monomers are configured as *self-avoiding-walks* (SAW) favoring H-H interaction in a 2D square or a 3D cube lattice or a *face-centred-cube* (FCC) arrangement [12]. These simple or low-resolution models allow the development, testing and comparison of various search algorithms which we have studied in this paper. The low-resolution model is applied within a hierarchical approach [13], [14], [15], [16], [17] to locate potential (backbone) conformation of the folded protein quickly and reliably [18] within the complex and convoluted search landscape before being explored further by a more realistic model. The approach is especially suitable for the computationally expensive *ab initio* prediction. The rationale behind the low-resolution model [3], [11], [19], [20], [21], [22], [23], [24], [25] and its current applications are widely available elsewhere [16], [26], [27], [28], [29], [30], [31].

For a structural search algorithm, the Genetic Algorithm (GA) is promising, because of GA's crossover and mutation operation. Crossover being the heart of GA, has also been adapted in many other nondeterministic search approaches [32], [33], [34]. Though effective [32], [33], [34], [35], [36], [37], [38], GA search can stall (i.e., stop searching) [39], especially for long sequences [37], and like other available algorithms, fails to solve PSP-like hard problems.

We investigate the aforementioned limitation by focusing on the growth of *twins* or *identical chromosomes* within the GA population leading to stalling and we revisit the concept of identical chromosomes. Moreover, we relax the concept by defining a new *chromosome correlation factor* (CCF) to include similar (strongly correlated) chromosomes to maintain an optimum percentage of diversity, which is confirmed empirically using benchmark PSP sequences [40], [41] along with other biological sequences. Randomly selected single-point, instead of multi-point crossover and mutation are used [36], [37], [38] to avoid exacerbated collision or non-self-avoiding-walks [33], [42].

In the remainder of the paper, Section 2 defines the HP model, while Section 3 provides the reasoning for preferring GAs for solving PSP problems. Section 4 describes the appearance of twins in GAs, existing remedies and limitations, and then broadly redefined twins. After Section 5 on experiments and results, Section 6 discusses the impact of twin removal. Section 7 compares the best twin removal approach with other state-of-the-art PSP search approaches. Section 8 highlights the effect of twin removal combined with sophisticated GAs. Section 9 provides conclusions. A list of acronyms used throughout the paper is provided in the Appendix.

- *Md Tamjidul Hoque, Andrew Lewis and Abdul Sattar are with Institute for Integrated and Intelligent Systems (IIIS), Griffith University, Australia. E-mails: Tamjidul.Hoque@gmail.com, {a.lewis, a.sattar} @ Griffith.edu.au.*
- *M. Chetty is with the Gippsland School of Information Technology (GSIT), MONASH University, Australia. E-mail: madhu.chetty@ infotech.monash.edu.au.*

Manuscript received 2 June 2008; revised 24 Feb. 2009; accepted 1 Mar. 2009.

Digital Object Identifier 10.1109/TCBB.2009.34 1545-5963/$25.00 © 2009 IEEE

2 The HP Model

The HP lattice model is based on the premise that the effect of *hydrophobic* (H) interaction dominates protein folding with the energy of a conformation being defined by the number of topological neighboring contacts (TNs) between immediately non-sequential Hs. For an amino-acid sequence, $s = s_1, s_2, s_3, \cdots, s_N$ of length *N*, the PSP formally involves finding the conformation g where, $g^* \in G(s)$ and energy $E^* = E(G) = \min\{E(g) \mid g \in G\}$. $G(s)$ is the set of all valid SAW conformations of s [35]. If the number of TNs in a conformation g is q then the value of $E(g)$ is defined as $E(g) = -\varepsilon q$ and the optimum *fitness* is given by $F = \min(E(g))$, where F is regarded as the fitness of the conformation and the value of ε is usually assigned 1 [43] (a practice we follow in this paper). For chromosome presentation in the 2D HP model [44], three possible moves exist: (anti-clock wise) *Left* (L), (clock-wise) *Right* (R) and *Forward* (F) which are encoded as 0, 1 and 2 respectively (see Fig. 1).

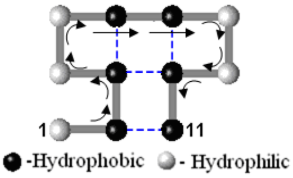

Fig.1. The conformation of the sequence *phhpphhpphh* in a 2D HP model is shown by the solid line. Dotted lines indicate *TN*. Fitness = -(*TN* Count) = -4. The '1' and '11' in the figure indicate the starting and ending residue in the sequence. Three different arrows indicating (anti-clock wise) *Left* (0), (clock wise) *Right* (1) and *Forward* (2) move can be used for chromosome encoding. The given conformation can be encoded 001122110.

3 Search Algorithms for Protein Structure Prediction

While solving PSP problems, using this simplified model on short sequences [45], [46], [47], [48], can however lead to an inordinately large number of conformations having admissible SAW [45]. For instance, for a sequence of N amino acids, the number of feasible SAW conformations is directly proportional to μ^N [48], where the connective constant μ (also referred to as the effective coordinate number) is lattice dependent [46]. The search for the best solution therefore becomes a major challenge, and the use of exhaustive search becomes quite infeasible [45], [46], [47], [48]. Further, the prediction problem has been proven to be NP-complete [49], [50] implying that a polynomial time algorithm is not feasible either. Statistical approaches to the PSP problem include *Contact Interaction* [51] and *Chain Growth* [52]. Both these techniques are characterized by exhibiting lower accuracy as the sequence length increases and also by being non-reversible in their move-steps while searching for optimum conformation. Alternative PSP strategies include *Artificial Neural Networks* (ANN) [53], *Support Vector Machines* (SVM) [54] and *Bayesian Networks* (BN) [55], while *Hidden Markov Models* (HMMs) which are based on *Bayesian* learning, have also been used to convert multiple sequence alignment into *position-specific scoring matrices* (PSSM), which are subsequently applied to predict protein structures [56], [57]. These approaches are often dependent on the training set and thus mostly applicable to the homology modeling and threading-based approaches rather than *ab initio* PSP problems. In particular, if the training sets are unrelated to the test sets, then information relating to a particular motif does not assist in a different motif.

For deterministic approaches to the PSP problem, approximation algorithms [58], [59], [60] provide an insight, though they are not particularly useful in identifying minimum energy conformations [40], and while linear programming (LP) methods have been used for protein threading [61], [62], [63], they have not been applied in *ab initio* applications, with the recent LP focus being confined to approximating the upper bound of the fitness value based on sequence patterns only [64].

Therefore, non-deterministic search techniques have dominated attempts to solve the PSP problem, of which there are a plethora including *Monte Carlo* (MC) simulation, *Evolutionary* MC (EMC) [32], [33], *Simulated Annealing* (SA), *Tabu Search* with *Genetic Algorithms* (GTB) [34], *Ant Colony Optimization* [35], *Immune Algorithm* (IA) based on *Artificial Immune System* (AIS) [27], *Conformational Space Annealing* (CSA) [114], and so on. Due to their simplicity and search effectiveness, *Genetic Algorithms* [36], [37], [38], [42], [43], [65], [66] are very attractive [37], [38] especially for the crossover operation which can build new conformation by exchanging sub-conformations.

While GA performance can be very effective it still does not ensure the final generation contains an optimal solution since it can stall at any point [39], a phenomenon that becomes more prevalent as the sequence length increases [43]. As the GA search proceeds, growing similarity in the population leads to the growth of twins or identical chromosomes [67] which has been considered to contribute to stalling. Therefore, similarity growth is assumed to have particular significance in the PSP problem as GAs need relatively large numbers of iterations to converge compared to other less complex problems. So it is therefore necessary to gain an understanding of this condition to develop a more accurate and efficient method of PSP solution. A first stage is to revisit the idea of twins or identical chromosomes in the population, before relaxing the concept to embrace similar (highly-correlated) chromosomes.

4 Twins in GA Polulation

The working principle of GAs supports the hypothesis that similarity would grow within the population [67], [68], [69], [70], [71], [72], [73], [74]. With an initial random population, earlier generations cover a large search space in the fitness landscape and then the search moves stochastically, converging to a smaller search space. This implies a lower level of variation amongst the population. As a consequence, explicitly defined GA operations such as the crossover operation become implicitly controlled as the similarity in the population increases. This means that although we can set the crossover rate to a desired value, in many cases the operation generates no variation due to the similarity, thus effectively lowering the overall rate. Thus it was observed earlier that GA perform unreliably in finding the optimum solution, specially for hard optimization problems, as the GA search becomes stagnant [75], leading to stalling.

For a hard optimization problem such as PSP, the stalling of GA can have a detrimental effect. The search can frequently become trapped in local minima, unable to explore the vast search space [39], [76]. If the landscape is less complex then the negative impact of the presence of twins may remain insignificant [68]. However, if the landscape is relatively more complex, as it is in an *ab initio* based PSP problem [3], [4], [5], [6], [7], [8], [9], [10], [77], it takes more time to converge because of twin-growth which is an inherent property of GAs. On the other hand, increasing randomness within GA to maintain diversity can help GA to surf around a vast and convoluted landscape. However, this is also not a promising approach because it would not allow getting the optimum solution within a feasible time scale [68]. Thus GAs need to be operated somewhere in between the two extremes: (a) excessive twins or similarity within the population leading to stalling and (b) less similarity (i.e., increased randomness) within the population can lead to extremely slow convergence, especially for the hard optimization problem such as PSP. Concerns of search effectiveness pose questions of whether to accept increasing similarity or to take steps to remove similarity within the population. In this paper we attempt to find the optimal similarity within the population to avoid these pitfalls.

4.1 Existing Remedies for the Problem of Twins

The existence of twins and attempts for their removal in GA are not new ideas. Haupt proposed avoiding twin-growth by starting each chromosome with different patterns while initializing the population [71]. However, if the twin-growth is inherent in a GA search, then the effect of initialization using different patterns would quickly decline after initialization. Further, it has been advocated that tests need to be continually applied with the population to ensure identical chromosomes do not breed [69], [78]. If the similarities among chromosomes are reduced, then a GA may not converge. The high dissimilarity within the population would lead to searching too random, whereas if the twins are allowed to grow more then finding a non-similar chromosome to mate with will be difficult because of the inevitable occurrence of many twins, and it will rather be a costly exercise of finding dissimilar chromosomes. A resemblance of this is the creation of two distinct chromosome groupings within the population: a large collection of highly correlated chromosomes and a much smaller set of dissimilar ones, with mating restricted exclusively to members of the respective groups, thus producing different offspring. However, the problem is that these will soon become more similar as they inherit the communal features from each group, the equivalent of two dissimilar parents breeding an entire next generation.

Contrary to these approaches, Deb and Goldberg suggested [75] a totally opposite solution, advocating individuals be allowed to reproduce if they are very closely similar. In [44], we have shown that crossover between phenotypically identical chromosomes is a mutation operation which can indirectly introduce more randomness in the search. Poloni and Pediroda [79] proposed an alternative approach of *local Pareto selection* to maintain diversity. This approach consists of placing the population on a toroidal grid and choosing the members by means of a random walk in the neighborhoods of the given grid points. In fact, randomness in GA can provide a temporary remedy to the stalling condition, but requires infeasibly long time to converge for complex problems like PSP. Alternatively, for preserving diversity in the population, the crowding operator [80] applied earlier by DeJong may be used. In crowding, a newly formed offspring replaces the existing individual most similar to itself. Once again the optimal degree of similarity for PSP problem is not known. Another technique to maintain diversity is to impose a *niched-penalty* [81]. In this case, any group of individuals of sufficient similarity will have a penalty imposed to discourage their participation in the next generation. However, the percentage of similarity to treat as a group in this case is not well specified.

Therefore, we focus on the need for maintaining optimal twin level by twin removal which was originally highlighted in [67] to emphasize that duplicate chromosomes or twins reduce diversity and ultimately lead to poorer performance. The study was, however, confined solely to the detection and removal of identical chromosomes only, which we would like to extend by grading the level of similarity to be able to specify the best level for solving a particular problem.

In solving the PSP problem using GA, to mitigate the limitations caused by stalling many methods such as special operators [40], [51], statistical approaches [34], [51], [52] and special treatment techniques such as cooling [36], [37], [38], constraints and hybridization [32], [42], [65], [66], [82] have been attempted. Instead, a generic approach maintaining optimal similarity or optimal diversity for the PSP problem is needed. We seek a generic enhancement that can be combined with other non-generic approaches for further improvement.

4.2 Redefining Twins

The problem of stalling due to increasing similarity in a population suggests that the degree of similarity between chromosomes should be investigated. We review the notion of twins and then broaden it to include not only identical but also similar (highly correlated) chromosomes in the population. We define a *chromosome correlation factor* (CCF) defines the degree of similarity existing between chromosomes. CCF can take a value from 0 to 1 indicating similarity of chromosome from 0% (CCF = 0) to 100% (CCF = 1). Hence, a value of CCF = 0.8 implies that the similarity is 80% between two chromosomes. For similarity measurement between two individuals in the genotype, we measure similarity by counting the number of bits or characters used for presenting a chromosome that are identical in the two individual chromosomes as fol-

TABLE 1
BENCHMARK PROTEIN SEQUENCES FOR 2D HP MODEL

Length	*Sequences*	*Ref*
50	H2(PH)3PH4PH(P3H)2P4H(P3H)2PH4P(HP)3H2	[41]
60	P2H3PH8P3H10PHP3H12P4H6PH2PHP	[41]
64	H12(PH)2(P2H2)2P2HP2H2PPHP2H2P2(H2P2)2(HP)2H12	[41]
85	4H4P12H6P12H3P12H3P12H3P1H2P2H2P2H2P1H1P1H	[40]
100_a	6P1H1P2H5P3H1P5H1P2H 4P2H2P2H1P5H1P10H1P2H1P7H11P7H2P1H1P3H6P1H1P2H	[40]
100_b	3P2H2P4H2P3H1P2H1P2H1P4H8P6H2P6H9P1H1P2H1P11H2P3H1P2H1P1H2P1H1P3H6P3H	[40]

'H' and 'P' in the sequence indicate hydrophobic and hydrophilic amino acid, respectively.

lowed in [83]. For presenting the chromosome in the 2D HP (the 2D HP model is used in this paper) for the PSP problem [44], a 3-level code (0, 1 and 2) is used for presenting the moves' directions: Left, Right and Forward (see Fig. 1).

5 SIMULATION AND EXPERIMENTAL RESULTS

The protein structure prediction (PSP) problem in the lattice model with long sequences generally has complex energy landscapes [3], [4], [5], [6], [7], [8], [77], and hence those sequences will take longer to converge. Simulations were undertaken using a simple GA on a selection of PSP benchmark sequences [40], [41] shown in Table 1 for the 2D square HP lattice model [11], with particular emphasis towards longer protein sequences, length ranging from 50 to 100. As highlighted earlier, these longer sequences demonstrate a greater propensity towards twin formation, so the impact on convergence for them can be expected to be more severe. We attempt to determine the optimum value of CCF for solving the PSP problem.

The default GA parameters [71], [84] for all experiments were set as population size (Pop_z) of 200, crossover rate (p_c) of 0.8 or 80%, mutation rate (p_m) of 5% and for the elitism the elite rate was set to 5%. We allowed a maximum of 6000 generations per PSP sequences per run.

A group of simulations were carried out with twin removal being omitted from one run and twins being removed in other runs for a range of CCF settings from r = 1.0 (identical chromosomes only) to r = 0.2 (the loosest chromosome similarity tested), i.e., 0.2 ≤ CCF ≤ 1.0, in steps of 0.1.

In implementing twin removal, when comparing any two chromosomes, a knock-out system was adopted based on the superior fitness value in a *correlated twin removal* (CTR) algorithm (Fig. 2), where the chromosome with the lower fitness was removed rather than selecting chromosomes arbitrarily for removal. CTR uses the minimum admissible correlation value r when comparing chromosome pairs for conformational similarity (Line 4), and if a twin is identified, the one with the lower fitness is removed (Lines 5 to 7). Following twin removal from a population, the gap is filled by randomly generated chromosomes, which for simplicity are not crosschecked for further twins.

```
INPUT: Population size= $Pop_z$ , Chromosome ($C$) length = $n$
         Minimum admissible correlation $r$ with $CCF \geq r$

OUTPUT: Population without twins of size $\leq Pop_z$
Assumption: RetSimilarity (i, j) returns percentage of similarity
between $C$(i) and $C$(j), where $i \neq j$.

BEGIN
1.  FOR $i$ = 1 to $(Pop_z - 1)$ DO
2.   { IF $C(i)$.MarkDeleted = False THEN
3.      FOR $j$ = $i$+1 to $Pop_z$ DO
4.        { IF RetSimilarity(i, j)≥ $r$% THEN
5.          { IF $|C(i).Fitness| < |C(j).Fitness|$ THEN
6.                        s{Swap ($C(i)$, $C(j)$) }
7.            $C(j)$.MarkDeleted = True } }}
8.        }}
END.
```

Fig.2. Correlated Twin Removal (CTR) Algorithm.

To clarify the notation used in describing the results, simulation runs that include twins are referred as WT, and are the same as the GA-based approach by Unger *et al*. [37] but without cooling. Those runs with twin removal implemented for all chromosomes with CCF ≥ r are denoted as TR-r, so TR-60 refers to the removal of all chromosome twins having an admissible similarity value of 0.6 (60%) or greater. Thus, TR-100 refers to removal of chromosomal twins that are identical, the only case considered by Ronald in his version of twins removal [67].

Table 2 shows the fitness results for a typical sequence of length 50 from Table 1, that was allowed to run to 6000 generations per iteration, with the number in brackets being the actual generation number when the optimum fitness value of -21 was reached. If during a run the optimal value was not reached, the minimum value achieved

TABLE 2
RESULTS OF 5 ITERATIONS OF SEQUENCE LENGTH 50.

WT	*TR-100*	*TR-90*	*TR-80*	*TR-70*	*TR-60*	*TR-50*	*TR-40*	*TR-30*	*TR-20*
-17	-18	**-21** (287)	**-21** (1244)	**-21** (992)	**-21** (4671)	-20	-20	-17	-17
-19	**-21** (2776)	**-21** (5209)	**-21** (2423)	**-21** (1721)	**-21** (5568)	-20	-19	-17	-16
-18	-20	-20	**-21** (488)	**-21** (611)	**-21** (1668)	-19	-18	-17	-17
-18	-18	**-21** (1711)	**-21** (928)	**-21** (1696)	-20	-20	-17	-18	-16
-19	-20	-20	**-21** (345)	**-21** (295)	-20	-20	-19	-18	-17

Data format: Minimum fitness value (Generation number). Bold entries indicate achieved best values. Maximum generation = 6000 and minimum (optimum) fitness = -21. The number within brackets indicates the actual generation number when the optimum fitness value -21 was reached; otherwise when the numbers within brackets are not shown; it implies the achieved minimum fitness value till 6000 generations irrespective of whenever it is achieved.

is shown. In the 5 separate runs (Table 2), WT never reached the optimum fitness value and stalled mostly before reaching the 250th generation, though the simulation ran for the entire 6000 generations. We consider this WT run as a direct consequence of twins with a higher fitness appearing in the population, thereby slowing the convergence towards an optimum solution over time as the population becomes less diverse.

Fig. 3 shows the *Generation vs. Overall similarity* plot. For the WT run, the overall similarity reached ~ 80% very rapidly (around the 50th generation) from an initial value of ~ 35%, before stabilizing at ~ 90% similarity after the 150th generation. This clearly supports the idea that without any twin removal policy the overall population quickly becomes highly correlated and diversity is lost. In such a satiation, with crossover mostly among similar chromosomes, the offspring remain closely similar to their parents regardless of the crossover position. Thus, the search through the population becomes stalled.

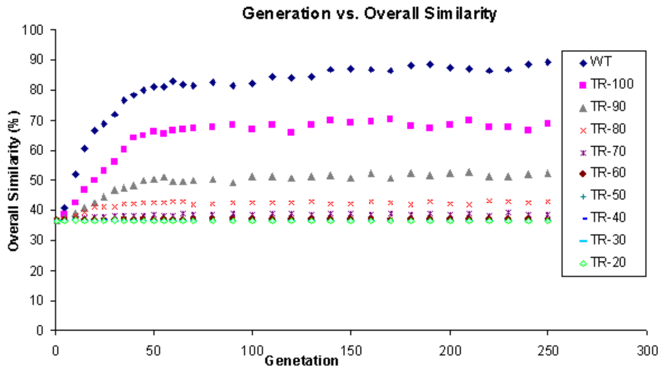

Fig. 3. Generation vs. Overall Similarity (%) plot for seq. of length 50.

Eventually, the overall dissimilarity, or diversity, in the chromosomes settled around 10%, which proved insufficient to maintain a search capability, and so it became trapped. It must be emphasized that with such a large number of generations the effect of mutation is negligible even if elitism (i.e. preserving a small proportion (5% to 10%) of elite chromosomes through the generations) is applied, as also selection fallacies allows less or no contribution. Mutation becomes ineffectual once similarity grows, as demonstrated by the WT runs quickly becoming saturated. Additionally, this mutated chromosome has then to survive (which is unlikely) to subsequently meet with a dissimilar chromosome to produce a fitter chromosome (again, highly improbable), so that the combined probability of these occurrences will tend to be zero.

The TR-100 simulation removes only identical chromosomes so, as Fig. 3 shows, it is closely similar to the WT run. Its average replacement percentage is ~ 17% per generation, meaning similarity has grown significantly within the remaining ~ 83% of the population. Such a high similarity level will inevitably incur an asymptotically long run to ensure convergence to the optimal fitness, which is especially problematic for longer PSP sequences since it implies that the convergence time will become infeasible.

TABLE 3
PERCENTAGE OF AVERAGE CHROMOSOME REMOVAL FOR VARIOUS PSP SEQUENCES, LENGTHS FROM 50 TO 100.

Length	*TR-100*	*TR-90*	*TR-80*	*TR-70*	*TR-60*	*TR-50*	*TR-40*	*TR-30*	*TR-20*
50	17.2	33.6	39.5	45.4	53.2	76.9	95.0	98.3	99.00
60	11.8	25.7	31.1	38.3	45.0	67.9	94.2	98.2	99.01
64	13.4	23.0	28.5	36.7	43.7	64.5	93.1	97.9	99.00
85	6.2	16.7	24.5	32.0	40.4	50.9	91.9	98.0	99.04
100_a	5.4	20.9	23.4	32.8	40.6	49.0	90.5	98.2	99.02
100_b	6.2	19.7	25.4	33.6	41.2	49.8	90.5	98.2	99.04

Table 3 shows the average removal (replacement) percentage of chromosome per generation for various sequence lengths. It can be seen that as the sequence length increases from 50 to 100 the corresponding replacement percentages reduce slowly, so the population of TR-100 for instance becomes less diverse as the sequence lengthens. Conversely, the replacement probabilities for TR-20, TR-30 and TR-40 are very high, which is to be expected given that they are replenishing almost the entire population with random conformations. This leads to a less correlated search, with the corollary that convergence to the optimum failed, as corroborated in Table 2 presented results for the TR-20 to TR-50.

It is also clear that TR-80 and TR-70 display the best removal performance for correlated twins, as the population maintains the most favorable balance between the *overall similarity* (chromosome correlation), keeping the search correlated to aid convergence, and maintaining diversity by supporting the growth of dissimilar but competent chromosomes. Overall, TR-80 provided the best performance by maintaining an optimal level of replacements.

Finally, to investigate the effectiveness of the twin removal approach, five other PSP sequences of different lengths (See Table 1) were also tested, and the 'generation vs. overall similarity graphs' for all of the other sequences were found to be fully consistent with the plots in Fig. 3. In all cases, the results were similar to those presented for sequence length 50, from which we conclude that a twin removal strategy based upon a CCF value ≥ 0.8 (80%) consistently provides the best (optimal) setting. As the landscape of PSP problems has too many local minima, a little more randomness helps, whereas for relatively less complex landscapes less randomness (i.e., higher correlation) would do better.

Additional Biological Sequences: To verify the consistency of the previous result we have selected actual protein sequences of various lengths from the Protein Data Bank (PDB) [85] and then converted the amino acid sequences into HP sequences to make them workable with our algorithm. The results shown in Table 4 are entirely consistent with the results previously shown.

Variations in GA parameter: To justify the preferred settings of the GA parameters, we took a representative set of TR-80, TR-100, WT and varied the GA parameters for

TABLE 4
AVERAGE MINIMUM FITNESS VALUES ACHIEVED FOR PDB SEQUENCES

PDB ID	*Length*	*WT*	*TR-100*	*TR-90*	*TR-80*	*TR-70*	*TR-60*	*TR-50*
1PJF	46	-22.0	-24.2	-24.7	-**25.0**	-24.5	-24.5	-24.0
1AAF	55	-13.2	-14.2	-**15.0**	-14.4	-14.4	-14.3	-14.0
2PTL	78	-20.6	-22.2	-24.6	-**25.0**	-24.8	-24.7	24.4
1GH1	90	-23.2	-25.8	-29.0	-**30.0**	-29.3	-28.5	-28.0
2GG1	102	-28.4	-31.8	-**35.0**	-**35.0**	-34.3	-34.3	-34.0
2CQO	119	-37.6	-41.4	-44.0	-**45.4**	-44.0	-44.0	-40.0

Source: PDB sequences [85]. Bold entries indicate best values achieved. Here, simulation results of 5 iterations of PDB (HP converted) sequences of different lengths. In every case a maximum of 6000 generations was allowed.

population size (Pop_z) 100, 200 and 300, crossover rates (P_c) 0.8, 0.4 and 0.1 and mutation rate (P_m) 0.80, 0.40 and 0.05. For the various parameter settings, twin removal for similarity ≥ 80% (i.e., TR-80) was found to be the most effective (see Table 5) with Pop_z = 200, P_c = 0.8 and P_m = 0.05.

6 DISCUSSION: IMPACT OF TWIN REMOVAL AND PREFERRED CCF

In an earlier section we introduced the concept of removal from the population of twins of different level of similarity. In the experiments reported in the preceding section, we have empirically demonstrated that this concept can provide an effective GA that is less likely to stall. We propose the modified twin removal is an effective approach because without twin removal the inexorable growth of identical and also progressively more highly correlated twins can lead to premature convergence or stalling in the search process [39], [86]. This situation is exacerbated by the crossover creating more twins and the impact of mutation becoming increasingly ineffectual. We will now consider these issues the context of the twin removal.

6.1 Premature Convergence or Stalling

Stalling or premature convergence in the WT runs show that nearly all offspring generated throughout the population are likely to be similar and will go forward to the next generation, with the result that there will be almost no variation in subsequent generations. With ineffectual crossover due to the presence of a large number of highly similar members of population, it can be surmised that the strategies to facilitate efficient removal of both identical and highly correlated twins will improve the GA performance, a premise that is fully corroborated in the experimental results given in previous section for the non-WT runs.

6.2 Ineffective Mutation

The growth of correlated twins inevitably weakens the impact of mutation despite its introduction of random variations. Therefore, different variant component or sub-conformations within a conformation will quickly disappear in the many highly correlated chromosomes in the population. Thus, when the number of highly correlated chromosome count (w_k) becomes very close to the population size (Pop_z), the chromosomes selected for mutation ($C_{mutated}$) are likely to be similar (high CCF value) to the majority of the population. Thus, when the mutational change is insignificant relative to the (previous or parental) conformation, $C_{mutated}$ will remain similar to the majority of chromosomes (C_k); i.e., mutation will have very little impact. However, if the conformational change differs heavily with respect to C_k, then the following two scenarios arise:

i) After mutation, $C_{mutated}$ has a lower fitness ($f_{mutated}$) than average, so it is less likely to be selected, and thus it will not be in the next generation.

ii) After mutation, $C_{mutated}$ has a higher fitness than average, but is not similar to the more populous chromosomes, and so while $f_{mutated} > f_k$, as $w_k \rightarrow Pop_z$ the effect within the proportional selection procedure (such as roulette-wheel, Fig. 4) becomes $f_{mutated} << w_k f_k$. Thus the fitter chromosome can be lost away [69], [87] by relying on the

TABLE 5
VARIATIONS OF THE GA PARAMETERS FOR TWIN REMOVAL

WT (Sequence Length ↓)					*TR-100* (Sequence Length ↓)					*TR-80* (Sequence Length ↓)					*GA Parameters*		
50	60	64	85	100_b	50	60	64	85	100_b	50	60	64	85	100_b	Pop_z	P_c	P_m
18	29.8	31.8	43.6	38	19.8	30.6	34.4	43.2	39.2	20.8	33.4	36.6	45.8	43.8	100	0.8	0.05
17.4	30.2	30.8	41.6	39.4	20	31.8	34.4	45.4	40.6	**21**	33.4	35.6	45.8	44.4	300	0.8	0.05
17.6	29.7	30.1	40.2	38.5	19.8	30.6	35.2	45.4	40	**21**	33.6	36.4	45.6	44.5	200	0.1	0.05
17.8	30.6	28.4	40.8	37.6	19.2	31.2	34.2	45.6	40.6	**21**	**33.8**	36.4	46.4	44.2	200	0.4	0.05
17.4	30	29.8	42.8	38	19.8	30.6	33	44.6	39.2	**21**	**33.8**	36.6	45.4	42.2	200	0.8	0.4
17.7	30.3	30.7	43.5	39.5	19.5	30.9	33.2	44.3	39.7	20.8	33.6	35.8	45.6	43.6	200	0.8	0.8
18.2	29.4	29.4	42.2	38.8	19.4	32.6	34.2	45	39.8	**21**	**33.8**	**37**	**46.8**	**44.8**	200	0.8	0.05

Each entry indicates average |fitness| value achieved from 5 iterations and each of the iterations ran up to 6000 generations. Five benchmark sequences length from 50 to 100 have been used. Bold entries indicate best values achieved.

fallaciously weighted selection procedure. For instances, to notice the 'fallaciously weighted selection', consider Fig. 4, where pie chart is assumed as a chart of a population of seven chromosomes having differing fitness as in Fig. 4(a) 7, 6, 5, 4, 3, 2, 1 and in Fig. 4(b) 7, 5, 5, 5, 5, 2, 1. Fig. 4(a) is the scenario usually assumed in the literature [39], [69], [70], [86], whereas in our findings in Fig. 4(b), it is shown that fitness 5 occupies 68% in total, so the probability of a rolling marble (on the assumed roulette wheel (selection procedure)) randomly selecting a pie slot having fitness 5 can be expressed as $p_{effective} = \sum_{i=2}^{5} p_i = 20/30$. The *effective* selection probabilities for corresponding chromosomes (in the respective order) for Fig. 4(b): C_1, (C_2 or C_3 or C_4 or C_5), C_6 and C_7 are thus can actually be computed 7/30, 20/30, 2/30 and 1/30 respectively. The fallacy in this example is that the selection probability of the chromosome having fitness 5 is actually higher than of the chromosome having fitness 7.

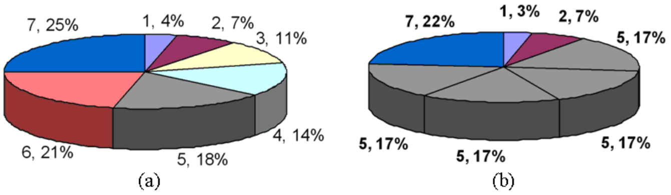

Fig. 4. Pie chart of a population of seven chromosomes having differing fitness factors (a) 7, 6, 5, 4, 3, 2, 1 and (b) 7, 5, 5, 5, 5, 2, 1. Each individual fitness is indicated by a separate slot in the pie. ***Legend***: *Fitness*, *Fitness* % (with respect to the sum of the fitness values)

That is, the chances of $C_{mutated}$ being selected for reproduction in the next generation are lower and it is likely that the fitter $C_{mutated}$ can die away due to selection fallacy, thereby leading to an effective mutation rate of $p_m \approx 0$.

6.3 Proposed CCF value

From the empirical result in the previous section, we have seen that TR-80 (i.e., 80% and higher percentage of similarity removal) outperformed the other runs with different CCF values and the WT runs as well. Therefore, based on the empirical result we prefer TR-80 for solving PSP problems.

6.4 Cost of Twin Removal

For the sake of completeness, we have also computed the cost of twin removal. From our 'Correlated Twin Removal' (CTR) Algorithm (see Fig. 2), it can be seen that the time complexity is proportional to the quadratic of the chromosomal length. However, this overhead may not impact heavily as more time is involved in handling other part of the overall process, such as handling phenotypical rotation, checking for self-avoiding-walk in every move and so on. To provide an indication of the cost, we have determined the actual run-time and included that in Table 6 in Section 7. Thus, for the sequences ranging from 50 to 100, the TR-*x* needed around 5 times more time compared to WT. However, the time for each operation, such as mutation and crossover, is not the same and each of the algorithms mentioned in Section 7 are not using the same set of operators. Therefore, straight time-based comparison may not be appropriate. Further, in this article our major concern is to achieve improved accuracy rather than fast computation as the accuracy becomes a more important issue for this hard optimization problem. As we see, many fast algorithms running for a long time do not produce optimum solutions, as can be seen in Table 6 and Table 7. However, the time for twin removal issues can be reduced easily by stopping the comparison immediately after encountering *x*% of dissimilarities in a TR-*x* approach. Thus, for TR-100 the comparison between two chromosomes can be stopped as soon as a single dissimilarity is encountered or for TR-80 the comparison can be stopped immediately after encountering at least 20% of dissimilarities and so on.

7 Comparision with Other Nondeterministic Approaches

To confirm 'twin removal' as a generally applicable component within GAs, we empirically compare TR-80 with other methods in their core forms (since, twin removal within GA is not domain knowledge dependent). For this, we discuss the fundamental nondeterministic approaches to be considered and then extract the core form of the state-of-art PSP algorithms (as named in Section 2).

The major categories of nondeterministic search approaches in their core forms are mostly covered by: the iterated hill climbing (IHC) approach, simulated annealing (SA) and the genetic algorithm [69]. IHC starts with a random bit string and then obtains a set of neighboring solutions by single bit changing (i.e., mutating) of the current solution (S_{curr}). Among the new solution(s) (S_{new}) (including the current one), the best is retained as the current solution (see Equation (1)), with the process being repeated until the stopping criterion is met. SA uses the same framework (see Equation (1)), but differs in its acceptance criterion when the new solution is not better than the current, SA can still accept the solution based upon some defined criteria (see Equation (2)).

$$S_{curr} \leftarrow S_{new}, \quad if\ f(S_{new}) > f(S_{curr}) \tag{1}$$

$$S_{curr} \leftarrow S_{new}, \quad if\ \text{random}[0,1) < \exp\left(\frac{f(S_{new}) - f(S_{curr})}{T}\right) \tag{2}$$

Here, f is the fitness function and T is a (symbolic temperature) variable, after having an initial value; T is gradually decreased in each iteration which is often regarded as cooling. SA explores more solution space compared to IHC, with the randomness introduced for selection in (1) and (2). The selection approach is regarded as a Monte Carlo (MC) method and so the SA with MC method is often simply referred to as MC in the literature [36], [37], [38], [51]. GA differs by mainly having a population of solutions and GA obtains new solutions by mixing the current solutions using a crossover operation, and then randomly replacing bit(s) by mutation. Next, we analyse the state-of-art approaches to extract their core form.

The GA-based approach proposed by Unger and Moult pioneered the working paradigm of the nondeter-

TABLE 6
COMPARISON OF VARIOUS ALGORITHMS BASED ON BENCHMARK SEQUENCES (SEE TABLE 1)

Length	*Labels*	*SA*	*IHC*	*UGA*	*CSA*	*WT*	*TR-100*	*TR-80*
50	Min. / Gen.	-14 / 3737	-16 / 2713	-19 / 3100	-19 / 512	-19 / 223	**-21**/ 2776	**-21**/ 2423
	Med./ Gen.	-13 / 1967	-15 / 3087	-18 / 53	-16 / 249	-18 / 778	-20/ 3711	**-21**/ 928
	Avg.	-13.4	-15.4	-17	-17.0	-18.2	-19.4	**-21.0**
	Run Time	205.85	213.61	453.51	357.25	306.30	1845.56	1834.45
60	Min./ Gen.	-26 / 124	-30 / 2678	-31 / 2855	-32/ 1708	-32/ 1288	**-34**/ 3659	**-34**/ 4859
	Med./ Gen.	-25 / 3064	-29 / 5169	-30 / 1613	-31/ 386	-30/ 1982	-32/ 3225	**-34**/ 2176
	Avg.	-25.0	-28.8	-29.6	-30.4	-29.4	-32.6	**-33.8**
	Run Time	477.60	442.92	926.16	727.56	557.47	3073.88	2777.88
64	Min./ Gen.	-27 / 5882	-28 / 3962	-33 / 4063	-30/ 586	-31/ 5730	**-38**/ 5825	**-38**/ 5185
	Med./ Gen.	-22 / 2918	-26 / 802	-29 / 5630	-30/ 192	-30/ 5829	-35/ 3973	**-37**/ 4837
	Avg.	-23.2	-26.2	-29.4	-29.0	-29.4	-34.2	**-37.0**
	Run Time	454.23	439.31	910.73	705.25	550.56	3233.85	2944.54
85	Min./ Gen.	-32 / 4833	-39 / 2392	-46 / 841	-46/ 1207	-44 / 455	-46/ 4199	**-49**/ 2930
	Med./ Gen.	-32 / 487	-37 / 2317	-43 / 3898	-43/ 676	-43 / 963	-45/ 5820	**-47**/ 5349
	Avg.	-31.8	-37.0	-42.6	-43.3	-42.2	-45.0	**-46.8**
	Run Time	755.34	727.84	1500.76	1243.26	885.86	4854.49	4334.77
100_a	Min./ Gen.	-28 / 1388	-31 / 2846	-39 / 1173	-40 / 261	-41/ 4566	-42/ 5332	**-45**/ 2083
	Med. /Gen.	-27 / 1593	-31 / 245	-38 / 5879	-38/ 1719	-37/ 624	-40/ 663	**-43**/ 2179
	Avg.	-27.0	-30.6	-36.4	-37.6	-37.8	-39.6	**-43.2**
	Run Time	739.25	667.15	1426.89	1201.42	862.29	5647.41	5039.35
100_b	Min./ Gen.	-26 / 5671	-32 / 1356	-40 / 3071	-39/ 5439	-41/ 2517	-43/ 734	**-46**/ 5486
	Med. /Gen	-26 / 3180	-31 / 477	-40 / 375	-37/ 198	-39/ 3932	-41/ 166	**-45**/ 4838
	Avg.	-25.6	-31.1	-38.4	-37.1	-38.8	-39.8	**-44.8**
	Run Time	678.13	723.74	1431.07	1255.39	874.42	5661.52	4992.44

Here, Min. indicated minimum energy achieved (i.e. maximum fitness), Med. indicates median and Avg. indicates average, where Gen. indicates (GA equivalent) generations when the particular fitness value is achieved. 'Run Time' indicates the total run time (in seconds) needed for 6000 GA or GA equivalent generations. Bold entries indicate best values achieved.

ministic search for the PSP problem using the lattice model. To solve the PSP problem using the HP model, Unger *et al.* [37], [38] enhanced the Simple Genetic Algorithm (SGA) by including the selection criteria given in Equation (1) and Equation (2) within the GA. This selection strategy is referred to as *cooling*. This augmented GA or, Unger's GA (UGA) outperformed the MC variants [36], [37]. We have considered this competitive approach for comparison with TR-80.

Two variations of Monte Carlo (MC), named MC algorithm [32] and the evolutionary Monte Carlo (EMC) [33] algorithm performed close to UGA but incorporated most of the components of UGA and domain knowledge, such as the lattice version of the known secondary structure. However, the incorporation of such secondary structure has a potential risk of easily missing the putative ground energy state [40], especially for longer sequences. If this is avoided the reduced core form ultimately becomes the same as the UGA. Further, Jiang *et al.* [34] applied the GA with *Tabu* search (GTB) for the 2D HP sequences. Tabu search is a local search technique which enhances the performance of a local search method by using memory structures and maintains dissimilar solutions by rejecting duplicates or closely similar chromosomes. However, there tends to be an inordinately large number of possible solutions [45], [46], [47], [48], and so the memory requirements tend to be prohibitive for longer sequences [40]. Similarly, the Elastic Net algorithm with Local Search method (ENLS) [88] also uses memory structures to store intermediate results and so also scales poorly. GTB and EMLS performance deteriorated for sequences greater than moderate length, whereas our proposed TR-80 can maintain optimal diversity without extensive memory requirements. Ant colony optimisation (ACO) has also been applied to the lattice-based PSP problem [35]. However, using the core ACO algorithm [89], the outcome is the same as in UGA, but it failed much earlier [88] for longer sequences.

Conformational space annealing (CSA) was developed with the aim of maintaining diversity by considering conformational distance [90], [91], conceptually close to TR-80. The CSA was applied to solve PSP using HP lattice [90] and recently incorporated in ROSETTA [92], [93], [94], [95], [96]. CSA is basically the combination of the best part of the GA and Monte Carlo selection approaches. To apply the CSA, typically two things are necessary: first, a method for perturbing a seed conformation and second, a distance measure between two conformations to compare their conformational dissimilarities, with a view to maintaining diversity. Additionally, a local optimisation method is generally associated with the CSA approach. For solving PSP using the HP model, four local *moves* had been applied (similar to the EMC approach). Populations are renamed banks in CSA. We used the core form of the CSA by avoiding the domain knowledge-based local moves to compare with TR-80.

In summary, the proposed TR-80 approach was compared with SA, Iterated Hill Climbing (IHC) [69], Unger's GA (UGA) [37], and Conformational Space Annealing (CSA) [90]– all in their core form, and also including WT to represent Simple GA (SGA) and TR-100 to represent the impact of identical twin removal attempted by Ronald [67]. To ensure the equity of the analysis, all the different algorithms were allowed the same effective operations: up to 6000 generations of 200 population-based GA equivalent operations. Both SA and IHC search approaches were run by including 30 neighbors for IHC in each generation, while for SA, in addition, the parameters for the cooling temperature were 2 initially, decreasing by 0.99 every 200000 steps (set according to [37]) until the temperature becomes 0.15. The same is applied for selection strategies in UGA. For each method and each sequence at least 5 iterations were carried out. Table 6 and Table 7 compare the results for SA, IHC, UGA, CSA, WT, TR-100 and TR-80, based upon the benchmark and PDB sequences respectively. These tables provide minimum fitness and the generation when it was achieved, median value of the fitness and when that was achieved and the average value of the fitness obtained for each of the different runs for each individual sequence. The average values and the run time (in seconds) for 6000 GA iterations (or its equivalent) are also shown.

The performance of SA was the poorest. IHC showed slightly improved performance over the SA approach. UGA, CSA and WT performed close to each other, but were better than both SA and IHC. TR-100 outperformed UGA, CSA and WT in almost all cases. However, TR-80 performed best of all in all cases. It should also be noted that with increasing length of the sequences the compara-

TABLE 7
COMPARISON OF VARIOUS ALGORITHMS BASED ON PDB SEQUENCES.

PDB ID / Length	*Labels*	*SA*	*IHC*	*UGA*	*CSA*	*WT*	*TR-100*	*TR-80*
1PJF / 46	Min./ Gen.	-19/ 3384	-21/ 5030	-24/ 746	-22/ 531	-23/ 1732	-25/ 2066	-**26**/ 3671
	Med./ Gen.	-19/ 545	-21/ 615	-22/ 392	-21/ 196	-22/ 332	-24/ 1979	-**25**/ 1381
	Avg.	-18.6	-20.8	-20.4	-21.4	-22.0	-24.2	-**25.0**
1AAF / 55	Min. /Gen.	-10/ 1104	-11/ 1985	-14/ 767	-13/ 176	-14/ 1130	-**15**/ 5597	-**15**/ 3379
	Med./ Gen.	-9/ 2295	-10/ 496	-14/ 376	-13/ 116	-14/ 37	-**14**/ 1187	-**14**/ 153
	Avg.	-9.4	-10.4	-13.4	-12.6	-13.2	-14.2	-**14.4**
2PTL / 78	Min./ Gen.	-14/ 4472	-17/ 1270	-24/ 73	-22/145	-21/ 901	-24/ 2715	-**26**/ 4830
	Med./ Gen.	-13/ 4343	-16/ 1024	-21/ 1426	-17/ 18	-21/ 793	-22/ 804	-**25**/ 1806
	Avg.	-13.2	-16.0	-21.6	-18.0	-20.6	-22.2	-**25.0**
1GHI/ 90	Min./ Gen.	-17/ 618	-20/ 4392	-28/ 3849	-26/ 4973	-26/ 534	-27/ 2695	-**31**/ 2103
	Med./ Gen.	-16/ 2195	-19/ 915	-27/ 4999	-24/ 101	-23/ 1285	-27/ 324	-**30**/ 1735
	Avg.	-16.0	-19.2	-26.6	-23.2	-23.2	-25.8	-**30.0**
2GGI / 102	Min./ Gen.	-23/ 5033	-27/ 2526	-35/ 1345	-31/ 468	-32/ 2556	-32/ 4408	-**36**/ 1954
	Med. / Gen.	-21/ 2552	-26/ 83	-31/ 1908	-30/ 95	-27/ 1698	-32/ 2659	-**35**/ 363
	Avg.	-21.4	-25.8	-31.8	-30.0	-28.4	-31.8	-**35.0**
2CQO / 119	Min./ Gen.	-23/ 1389	-27/ 3797	-38/ 3318	-37/ 547	-40/ 3229	-44/ 4916	-**46**/ 3993
	Med. /Gen	-22/ 51	-27/ 2663	-38/ 2528	-35/ 316	-37/ 2095	-42/ 2470	-**45**/ 5914
	Avg.	-22.0	-27.3	-37.0	-35.0	-37.6	-41.4	-**45.4**

Here, Min. indicated minimum energy achieved (i.e. maximum fitness), Med. indicates median and Avg. indicates average, where Gen. indicates (GA equivalent) generations when the particular fitness value is achieved. Bold entries indicate best values achieved.

tive performance of all the approaches became clearer with greater difference among the achieved fitnesses.

8 INCORPORATING TWIN REMOVAL WITHIN OTHER SOPHICTICATED GENETIC ALGORITHMS

To check the effectiveness of our twin removal concept, we tested the performance of our TR-80 approach by combining it with (a) more sophisticated GA versions such as UGA and (b) hybrid genetic algorithm (HGA). The HGA version used the 3D face-centred-cube (FCC) lattice model which we developed previously [97]. Table 8 shows the results comparing the performance of "UGA + TR-80" over UGA alone. Table 9 shows the comparison for the HGA approach. The results are found promising: "UGA + TR-80" showed consistently better solutions than UGA alone, and HGA with twin removal consistently yielded results with fitness equal to, or better than, HGA without twin removal.

From these results it may be concluded that twin removal is generally an applicable approach yielding improvements, especially for longer sequences. For the shorter sequences in Table 9, the difference in performances appear very small, as can be true when applying any well-known method. However, for relatively longer sequences the superior performance of HGA combined with TR-80 over HGA alone can be distinguished. However, on rare occasions, performance can be closer based on the simple HP pattern in the sequences [42], [98].

Among possible limitations of the twin removal approach, associated overhead and the corresponding remedy for twin removal have been discussed in Section 6.4. Another seemingly minor issue could be with the application of move-sets or similar operators that can alter the twins and thus affect the diversity. The HGA approach applied here involves such a move-set [97]. Application of the move-set could be complementary to the twin removal approach, since comparatively less twin removal from the population would be required.

TABLE 8
PERFORMANCE COMPARISONS: UGA VERSUS 'UGA + TR-80'

Method ↓	***Length*** →	**50**	**60**	**64**	**85**	**100_a**	**100_b**
UGA	Avg. Fitness	-17.0	-29.6	-29.4	-42.6	-36.4	-38.4
UGA + TR-80	Avg. Fitness	-21.0	-33.2	-37.4	-46.4	-43.8	-44.4
Method ↓	***Length*** →	**46**	**55**	**78**	**90**	**102**	**119**
UGA	Avg. Fitness	-20.4	-13.4	-21.6	-26.6	-31.8	-37.0
UGA + TR-80	Avg. Fitness	-25	-14.2	-25.2	-29.4	-34.8	-44.8

Here, we compare the UGA versus UGA combined with TR-80 to see whether TR-80 can improve more sophisticated version of GA other than simple GA or not. Average (Avg.) fitness from 5 iterations running up to 6000 generations has been compared.

TABLE 9
PERFORMANCE COMPARISONS: HGA VERSUS 'HGA + TR-80'

Length / Sequence [42]	***HGA***	*HGA + **TR-80***
25 / PPHPPHHPPPPHHPPPPHHPPPPHH	-25	-25
36 / P3(H2P2)2P3H7P2H2P4H2P2HP2	-50	-51
48 / P2(HP2H)2HP510HP5(H2P2)2HP2H5	-65	-69
50 / H(HP)4H4PHP3HP3HP3HP4HP3HP3HPH4(PH)4H	-59	-59
60 / P2H3PH8P3H10PHP3H12P4H6PH(HP)2	-114	-117
64 / H12(PH)2((P2H2)2P2H)3PHPH12	-98	-103

Here, we compare the predictability of the developed HGA for solving PSP with and without TR-80. Achieved maximum |fitness| from 15 iterations is shown for every type of runs. All these runs are simulated for the same amount of time using same capacity machine and to have a fair comparison of their performance, runs for same sequence were terminated once any of runs becomes non-progressive.

9 CONCLUSIONS

The ease of genetic algorithm (GA) implementations has made GA popular for solving many optimization problems such as protein structure prediction (PSP) in the form of a conformational search algorithm. This neglects, however, the crucial impact the growth of similarity and chromosome twins has upon the population, which can lead to premature convergence, a condition that is especially evident when the application has a complex landscape, as in longer PSP sequences. Twins cause a population to lose diversity, resulting in both the crossover and mutation operations being ineffectual. This paper has highlighted the need for a chromosome twin removal strategy to maintain consistent performance. The definition of twins has been relaxed, not only to embrace duplicate chromosomes, but also to include high-correlated chromosomes within the twin removal strategy. Simulation results have been presented confirming the performance improvement achieved in the *ab initio* approach for

PSP applications by adopting this generalized twin removal strategy. Our empirical results show that a chromosome correlation factor (CCF) setting ≥ 0.8 affords the best performance for twin removal in terms of balancing optimal convergence with an effective search capability. The choice of TR-80 with GA showed outstanding performance compared to other algorithms in their core form when solving the PSP problem. Further, the proposed twin removal strategy was demonstrated to improve performance of a number of other GA-based algorithms.

Appendix

List of Acronyms

Abbr.	*Elaboration*	*Abbr.*	*Elaboration*
ACO	Ant Colony Optimization	HP	Hydrophobic-Hydrophilic
AIS	Artificial Immune System	IA	Immune Algorithm
CCF	Chromosome Correlation Factor	IHC	Iterated Hill Climbing
CSA	Conformational Space Annealing	MC	Monte Carlo
CTR	Correlated Twin Removal	PDB	Protein Data Bank
EMC	Evolutionary Monte Carlo	PSP	Protein Structure Prediction
ENLS	Elastic Net algorithm with Local Search method	SA	Simulated Annealing
		SAW	Self-Avoiding-Walks
FCC	Face-Centred-Cube	TNs	Topological Neighbouring contacts
GA	Genetic Algorithm	TR-x	Twin Removal for similarity ≥ x %
GTB	Tabu Search with Genetic Algorithm	UGA	Unger's Genetic Algorithm
		WT	Without Twin removal

Abbr. indicates addreviation.

Acknowledgment

The authors gratefully acknowledge support from Australian Research Council (grant no DP0557303). Feedbacks and comments from the anonymous reviewers and useful discussions with Professor Laurence S Dooley are gratefully acknowledged.

References

[1] J. T. Pedersen and J. Moult, "*Ab initio* protein folding simulations with genetic algorithms: simulations on the complete sequence of small proteins," *Proteins.* , vol. 29, pp. 179–184, 1997.

[2] G. B. Lamont and L. D. Merkie, "Toward effective polypeptide chain prediction with parallel fast messy genetic algorithms," in *Evolutionary Computation in Bioinformatics*, G. Fogel and D. Corne, Eds., 2004 pp. 137–161.

[3] K. A. Dill, S. Bromberg, K. Yue, K. M. Fiebig, D. P. Yee, P. D. Thomas, and H. S. Chan, "Principles of protein folding – A perspective from simple exact models," *Protein Science,* vol. 4, pp. 561–602, 1995.

[4] S. D. Flores and J. Smith, "Study of Fitness Landscapes for the HP model of Protein Structure Prediction," in *IEEE CEC*, 2003, pp. 2338–2345.

[5] N. Mousseau and G. T. Barkema, "Exploring High-Dimensional Energy Landscape," *Computing in Science & Engineering, IEEE,* vol. 1, pp. 74–80, 82, 1999.

[6] U. H. E. Hansmann, "Protein Folding in Silico: An Overview," *IEEE CS and the AIP,* pp. 64–69, 2003.

[7] Y. Cui, W. H. Wong, E. Bornberg-Bauer, and H. S. Chan, "Recombinatoric exploration of novel folded structures: A heteropolymer-based model of protein evolutionary landscapes," *PNAS,* vol. 99, pp. 809–814, 2002.

[8] K. Schreiner, "Distributed Project Tackle Protein Mystery," *Computing in Science & Engineering, IEEE,* vol. 3, pp. 13–16, 2001.

[9] K. A. Dill, J. B. Rosen, and A. T. Phillips, "Protein Structure and Energy Landscape Dependence on Sequence Using a Continuous Energy Function," *Journal of Computational Biology,* vol. 4, pp. 227–239, 1997.

[10] O. Schueler-Furman, C. Wang, P. Bradley, K. Misura, and D. Baker, "Progress in Modeling of Protein Structures and Interactions," *Science,* vol. 310, pp. 638–642, 2005.

[11] K. A. Dill, "Theory for the Folding and Stability of Globular Proteins," *Biochemistry,* vol. 24, pp. 1501–1509, 1985.

[12] R. Backofen and S. Will, "A Constraint-Based Approach to Fast and Exact Structure Prediction in Three-Dimensional Protein Models," *Constraints Journal,* vol. 11, pp. 5–30, 2006.

[13] Y. Xia, E. S. Huang, M. Levitt, and R. Samudrala, "Ab Initio Construction of Protein Tertiary Structures using a Hierarchical Approach," *J. Mol. Biol.* , vol. 300, pp. 171–185, 2000.

[14] C. A. Rohl, C. E. M. Strauss, K. M. S. Misura, and D. Baker, "Protein Structure Prediction Using Rosetta," *Methods in Enzymology,* vol. 383, pp. 66–93, 2004.

[15] Y. Zhang, A. K. Arakaki, and J. Skolnick, "TASSER: An Automated Method for the Prediction of Protein Tertiary Structures in CASP6," *PROTEINS: Structure, Function, and Bioinformatics* vol. 7 pp. 91–98, 2005.

[16] T. Hoque, M. Chetty, and A. Sattar, "Extended HP Model for Protein Structure Prediction," *Journal of Computational Biology,* vol. 16, pp. 85–103, 2009.

[17] R. Samudrala, Y. Xia, and M. Levitt, "A Combined Approach for *ab initio* Construction of Low Resolution Protein Tertiary Structures from Sequence " *Pacific Symposium on Biocomputing (PSB),* vol. 4, pp. 505–516, 1999.

[18] D. Chivian, T. Robertson, R. Bonneau, and D. Baker, "AB INITIO METHODS," in *Structural Bioinformatics*, P. E. Bourne and H. Weissig, Eds.: Wiley-Liss, Inc., 2003.

[19] K. M. Flebig and K. A. Dill, "Protein Core Assembly Processes," *J. Chem. Phys.,* vol. 98, pp. 3475–3487, 1993.

[20] K. Yue and K. A. Dill, "Sequence-structure relationships in proteins and copolymers " *Phys. Rev. E* vol. 48, pp. 2267–2278, 1993.

[21] C. B. Anfinsen, "Studies on the Principles that Govern the Folding of Protein Chains," 1972, http://nobelprize.org/nobel_prizes/chemistry/laureates/1972/anfinsen-lecture.pdf., *last access*, Feb 2009.

[22] K. A. Dill and H. S. Chan, "From Levinthal to pathways to funnels," *Nature Structural Biology,* vol. 4, pp. 10–19, 1997.

[23] H. S. Chan and K. A. Dill, "Protein folding in the landscape perspective: Chevron plots and non-arrhenius kinetics," *PROTEINS: Structure, Function and Genetics,* vol. 30, pp. 2–33, 1998.

[24] D. Baker, "A surprising simplicity to protein folding," *Nature,* vol. 405, pp. 39–42, 2000.

[25] J. Lee, S. Wu, and Y. Zhang, "*Ab Initio* Protein Structure Prediction," in *From Protein Structure to Function with Bioinformatics*, D. J. Rigden, Ed.: Springer Netherlands, 2009, pp. 3–25.

[26] R. Santana, P. Larrañaga, and J. A. Lozano, "Protein Folding in Simplified Models With Estimation of Distribution Algorithms," *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION,* vol. 12, pp. 418–438, 2008.

[27] V. Cutello, G. Nicosia, M. Pavone, and J. Timmis, "An Immune Algorithm for Protein Structure Prediction on Lattice Models," *IEEE Transactions on Evolutionary Computation,* vol. 11, 2007.

[28] C. Thachuk, A. Shmygelska, and H. H. Hoos, "A replica exchange Monte Carlo algorithm for protein folding in the HP

model," *BMC Bioinformatics,* vol. 8, 2007.

[29] Y. Ponty, R. Istrate, E. Porcelli, and P. Clote, "LocalMove: computing on-lattice fits for biopolymers," *Nucleic Acids Research,* vol. 36, pp. 216–222, 2008.

[30] A. D. Pal`u, A. Dovier, and E. Pontelli, "Enhancing the Computation of Approximate Solutions of the Protein Structure Determination Problem Through Global Constraints for Discrete Crystal Lattices," in *BIBMW, IEEE* 2007, pp. 38– 44.

[31] M. Mann, C. Smith, M. Rabbath, M. Edwards, S. Will, and R. Backofen, "CPSP-web-tools : a server for 3D lattice protein studies.," *Bioinformatics,* pp. 1–2, 2009.

[32] U. Bastolla, H. Frauenkron, E. Gerstner, P. Grassberger, and W. Nadler, "Testing a new Monte Carlo Algorithm for Protein Folding," *National Center for Biotechnology Information,* vol. 32, pp. 52–66, 1998.

[33] F. Liang and W. H. Wong, "Evolutionary Monte Carlo for protein folding simulations," *J. Chem. Phys,* vol. 115, 2001.

[34] T. Jiang, Q. Cui, G. Shi, and S. Ma, "Protein folding simulation of the hydrophobic-hydrophilic model by computing tabu search with genetic algorithms," in *ISMB* Brisbane Australia, 2003.

[35] A. Shmygelska and H. H. Hoos, "An ant colony optimization algorithm for the 2D and 3D hydrophobic polar protein folding problem," *BMC Bioinformatics* vol. 6, 2005.

[36] R. Unger and J. Moult, "On the Applicability of Genetic Algorithms to Protein Folding," in *The Twenty-Sixth Hawaii International Conference on System Sciences*, 1993, pp. 715–725.

[37] R. Unger and J. Moult, "Genetic Algorithms for Protein Folding Simulations," *Journal of Molecular Biology,* vol. 231, pp. 75–81, 1993.

[38] R. Unger and J. Moult, "Genetic Algorithm for 3D Protein Folding Simulations," in *5th International Conference on Genetic Algorithms*, 1993, pp. 581–588.

[39] D. B. Fogel, EVOLUTIONARY COMPUTATION Towards a new philosophy of Machine Intelligence: IEEE Press, 2000.

[40] N. Lesh, M. Mitzenmacher, and S. Whitesides, "A Complete and Effective Move Set for Simplified Protein Folding," in *RECOMB*, Berlin, Germany, 2003, pp. 188–195.

[41] W. E. Hart and S. Istrail, "HP Benchmarks," http://www.cs.sandia.gov/tech_reports/compbio/tortilla-hp-benchmarks.html, *last access*: August 2005.

[42] M. T. Hoque, M. Chetty, and L. S. Dooley, "A New Guided Genetic Algorithm for 2D Hydrophobic-Hydrophilic Model to Predict Protein Folding," in *IEEE Congress on Evolutionary Computation (CEC)*, Edinburgh, UK, 2005, pp. 259–266.

[43] D. W. Corne and G. B. Fogel, "An Introduction to Bioinformatics for Computer Scientists," in *Evolutionary Computation in Bioinformatics*, G. B. Fogel and D. W. Corne, Eds., 2004, pp. 3–18.

[44] M. T. Hoque, M. Chetty, and L. S. Dooley, "Non-Isomorphic Coding in Lattice Model and its Impact for Protein Folding Prediction Using Genetic Algorithm," in *IEEE Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)* Toronto, Canada: IEEE, 2006.

[45] M. Chen and K. Y. Lin, "Universal amplitude ratios for three-dimensional self-avoiding walks," *Journal of Physics A: Mathematical and General,* vol. 35, pp. 1501–1508, 2002.

[46] R. Schiemann, M. Bachmann, and W. Janke, "Exact Enumeration of Three – Dimensional Lattice Proteins," *Computer Physics Communications, Elsevier Science.,* vol. 166, pp. 8–16, 2005.

[47] D. MacDonald, S. Joseph, D. L. Hunter, L. L. Moseley, N. Jan, and A. J. Guttmann, "Self-avoiding walks on the simple cubic lattice," *Journal of Physics A: Mathematical and General,* vol. 33, pp. 5973–5983, 2000.

[48] A. J. Guttmann, "Self-avoiding walks in constrained and random geometries," in *Statistics of Linear Polymers in Disordered Media*, B. K. Chakrabarti, Ed.: Elsevier, 2005, pp. 59–101.

[49] P. Crescenzi, D. Goldman, C. Papadimitriou, A. Piccolboni, and M. Yannakakis, "On the complexity of protein folding (extended abstract)," in *the second annual international conference on Computational molecular biology*, 1998, pp. 597–603.

[50] B. Berger and T. Leighton, "Protein Folding in the Hydrophobic-Hydrophilic (HP) Model is NP-Complete," *Journal of Computational Biology,* vol. 5, pp. 27–40, 1998.

[51] L. Toma and S. Toma, "Contact interactions methods: A new Algorithm for Protein Folding Simulations," *Protein Science,* vol. 5, pp. 147–153, 1996

[52] E. Bornberg-Bauer, "Chain Growth Algorithms for HP-Type Lattice Proteins," in *RECOMB* Santa Fe, NM, USA, 1997, pp. 47–55.

[53] J. Vanhala and K. Kaski, "Protein Structure Prediction System based on Artificial Neural Networks," in *ISMB*, Bethesda, MD, USA, 1993, pp. 402–410.

[54] F. Markowetz, L. Edler, and M. Vingron, "Support Vector Machines for Protein Fold Class Prediction," *Biometrical Journal,* vol. 45, pp. 377–389, 2003.

[55] A. Raval, Z. Ghahramani, and D. L. Wild, "A Bayesian network model for protein fold and remote homologue recognition," *Bioinformatics,* vol. 18, pp. 788–801, 2002.

[56] P. Baldi and S. Brunak, *Bioinformatics: The Machine Learning Approach*: The MIT Press, 2001.

[57] B. Rost, "Review: protein secondary structure prediction continues to rise," *Journal of Structural Biology,* vol. 134, pp. 204–218, 2001.

[58] W. E. Hart and S. Istrail, "Fast Protein Folding in the Hydrophobic-hydrophilic Model Within Three-eights of Optimal," in *The twenty-seventh annual ACM symposium on Theory of computing* Las Vegas, Nevada, United States, 1995, pp. 157–168.

[59] R. B. Lyngsø and C. N. S. Pedersen, "Protein Folding in the 2D HP model," BRICS, 2000, http://www.brics.dk/RS/99/16/BRICS-RS-99-16.pdf, *last access*: Feb, 2009.

[60] A. Newman, "A new algorithm for protein folding in the HP model," in *ACM-SIAM symposium on Discrete Algorithms*, San Francisco, California 2002, pp. 876–884.

[61] D. G. Brown, "Bioinformatics Group" School of Computer Science, University of Waterloo Canada, http://monod.uwaterloo.ca/, *last access*: April 2007.

[62] J. Meller and R. Elber, "Linear programming Optimization and a Double Statistical Filter for Protein Threading Protocols," *PROTEINS: Structure, Function, and Genetics,* vol. 45, pp. 241–261, 2001.

[63] M. J. Panik, Linear Programming: Mathematics, Theory and Algorithm, 1996.

[64] R. Carr, W. E. Hart, and A. Newman, "Bounding A Protein's Free Energy In Lattice Models Via Linear Programming," in *RECOMB*, 2004.

[65] O. Takahashi, H. Kita, and S. Kobayashi, "Protein Folding by A Hierarchical Genetic Algorithm," in *4th Int. Symp. AROB*, 1999.

[66] R. König and T. Dandekar, "Refined Genetic Algorithm Simulations to Model Proteins," *Journal of Molecular Modeling,* vol. 5,

pp. 317–324, 1999.

[67] S. Ronald, "Duplicate Genotypes in a Genetic algorithm," *IEEE World Congress on Computational Intelligence,* pp. 793–798, 1998.

[68] M. T. Hoque, M. Chetty, and L. S. Dooley, "Generalized Schemata Theorem Incorporating Twin Removal for Protein Structure Prediction," in *Pattern Recognition in Bioinformatics*, Singapore, 2007, pp. 84–97.

[69] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution, 1992.

[70] D. Whitley, "An Overview of Evolutionary Algorithms," *Journal of Information and Software Technology,* vol. 43, pp. 817–831, 2001.

[71] R. L. Haupt and S. E. Haupt, *Practical Genetic Algorithms*, 2nd ed., 2004.

[72] M. T. Hoque, M. Chetty, and L. S. Dooley, "Critical Analysis of the Schemata Theorem: The Impact of Twins and the Effect in the Prediction of Protein Folding using Lattice Model," GSIT, MONASH University 2005, TR-2005/8.

[73] L. Altenberg, "The Schema Theorem and Price's Theorem " *Foundations of Genetic Algorithms 3,* 1995.

[74] D. B. Fogel and A. Ghozeil, "Schema processing, proportional selection, and the misallocation of trials in genetic algorithms," *Information Science,* vol. 122, pp. 93–119, 2000.

[75] K. Deb and D. E. Goldberg, "An investigation of niche and species formation in genetic function optimization," in *The Third International Conference on Genetic Algorithms* George Mason University, 1989, pp. 42–50.

[76] W. M. Spears, "Simple Subpopulation Schemes," *Proceedings of the Evolutionary Programming Conference,* pp. 296–307, 1994.

[77] J. Skolnick and A. Kolinski, "Computational Studies of Protein Folding," *IEEE COMPUTING IN SCIENCE & ENGINEERING,* vol. 3, pp. 40–50 2001.

[78] L. J. Eshelman and J. D. Schaffer, "Preventing premature convergence in genetic algorithms by preventing incast," in *The Fourth International Conference on Genetic Algorithms* University of California, San Diego, 1991, pp. 115–122.

[79] C. Poloni and V. Pediroda, "GA coupled with computationally expensive simulations: Tools to improve efficiency," in *In Genetic Algorithms and Evolution Strategies in Engineering and Computer Science: Recent Advances and Industrial Applications*, 1995, pp. 267–288.

[80] M. Mitchell, *An Introduction to Genetic Algorithms*: MIT Press, Cambridge, MA, 1996.

[81] K. Deb and S. Agrawal, "A niched-penalty approach for constraint handling in genetic algorithms," in *Artificial Neural Nets and Genetic Algorithms, Proc. of the Inter. Conf*, Portoroz Slovenia, 1999, pp. 235–243.

[82] R. Backofen and S. Will, "A Constraint-Based Approach to Fast and Exact Structure Prediction in Three-Dimensional Protein Models," *Kluwer Academic Publishers,* vol, 11, pp. 5–30, 2006.

[83] C. A. C. Coello, "An Updated Survey of GA-Based Multiobjective Optimization Techniques, "*ACM Computing Surveys,,* vol. 32, pp. 109–143, 2000.

[84] J. G. Digalakis and K. G. Margaritis, "An experimental Study of Benchmarking Functions for Genetic Algorithms," *Intern. J. Computer Math.,* vol. 79, pp. 403–416, 2002.

[85] PDB, "Protein Data Base," http://www.rcsb.org/pdb/, *last access*: Feb 2009.

[86] D. E. Goldberg, *Genetic Algorithm Search, Optimization, and Machine Learning* Addison-Wesley Publishing Company, 1989.

[87] L. Davis, *Handbook of Genetic Algorithm,* VNR, New York, 1991.

[88] Y. Z. Guo, E.-M. Feng, and Y. Wang, "Exploration of two-dimensional hydrophobic-polar lattice model by combining local search with elastic net algorithm," *Journal of Chemical Physics* vol. 125, pp. 1–6, 2006.

[89] A. Shmygelska, R. Aguirre-Hernández, and H. H. Hoos, "An Ant Colony Optimization Algorithm for the 2D HP Protein Folding Problem," *LNCS,* vol. 2463, pp. 40–52, 2002.

[90] J. Lee, "Conformational space annealing and a lattice model Protein," *Journal of the Korean Physical Society,* vol. 45 pp. 1450–1454, 2004.

[91] J. Lee, H. A. Scheraga, and S. Rackovsky, "New Optimization Method for Conformational energy Calculations on Polypeptides: Conformational Space Annealing," *Journal of Computational Chemistry,* vol. 18 pp. 1222–1232, 1997.

[92] Yiliu, "Rosetta 2.1.0.", 2007-2008 The Rosetta Commons, http://www.rosettacommons.org/tiki/tiki-index.php?page=Change+Log, *last access*: Feb 2009.

[93] R. Bonneau, J. Tsai, I. Ruczinski, D. Chivian, C. Rohl, C. E. M. Strauss, and D. Baker, "Rosetta in CASP4: Progress in Ab Initio Protein Structure Prediction," *PROTEINS: Structure, Function, and Genetics,* vol. 5, pp. 119–116, 2001.

[94] P. Bradley, D. Chivian, J. Meiler, K. M. S. Misura, C. A. Rohl, W. R. Schief, W. J. Wedemeyer, O. Scueler-Furman, P. Murphy, J. Schonbrun, C. E. M. Strauss, and D. Baker, "Rosetta Predictions in CASP5: Success, Failure, and Prospects for Complete Automation," *PROTEINS: Structure, Function, and Genetics,* vol. 53, pp. 457–468, 2003.

[95] K. T. Simons, R. Bonneau, I. Ruczinski, and D. Baker, "Ab Initio Protein Structure Prediction of CASP III Target Using ROSETTA," *PROTEINS: Structure, Function, and Genetics,* vol. 3, pp. 171–176, 1999.

[96] D. Baker, "Prediction and design of macromolecular structures and interactions," *Phil. Trans. R. Soc. B* vol. 361, pp. 459–463, 2006.

[97] M. T. Hoque, M. Chetty, and A. Sattar, "Protein Folding Prediction in 3D FCC HP Lattice Model Using Genetic Algorithm " in *Bioinformatics special session, IEEE Congress on Evolutionary Computation (CEC)*, Singapore, 2007, pp. 4138–4145.

[98] M. T. Hoque, M. Chetty, and L. S. Dooley, "Efficient Computation of Fitness Function by Pruning in Hydrophobic-Hydrophilic Model," in *In the 6th International Symposium on Biological and Medical Data Analysis (ISBMDA)* 2005, pp. 346–354.

Md Tamjidul Hoque received his B. Sc. Engg. and M.Sc. Engg. degrees in Computer Science and Engineering (CSE) from Bangladesh University of Engineering and Technology in 1998 and 2002 respectively and received his PhD degree in IT from Monash University (Australia) in 2008. He was a lecturer at the CSE department, Ahsanullah University of Science and Technology, 1998-99. He was in the technical management being IT incharge and DGM at Bashundhara Group, Dhaka, Bangladesh from December 1999-04. Currently he is a research fellow at Griffith University (Australia) involving two research institutes: IIIS and Eskitis. His current research focus is on '*ab initio* protein structure prediction' and 'high content image analysis and algorithm development'. His other interests include Algorithms, Networking, Communication, Database System, Complier Design, Automata Theory, Distributed Systems and Parallel Computing, Computer Architecture, Petri Net Theory, Security and Operating System.

Dr Madhu Chetty has been with Monash University, Australia since 1995 and is currently the Deputy Head of Gippsland school of Information Technology. His research interests include bioinformatics, optimization, computational intelligence, and modeling complex systems. Dr Chetty has authored over 100 scientific articles which include book chapters and articles in journals and international conferences. He is Senior Member of IEEE and Fellow, Institution of Engineers (India). He is currently serving as Chair of technical committee (TC-20) of International Association for Pattern Recognition (IAPR) on bioinformatics and was General Chair of PRIB'08 (Pattern Recognition in Bioinformatics) conference. He has also served as Vice Chair of the IEEE CIS Technical Committee on Bioinformatics and Bioengineering. He is serving as the Associate Editor of the Elsevier's Neurocomputing journal and is on the editorial board of three other journals in bioinformatics. Prior to his career at Monash, Dr. Chetty worked at VRCE (now VNIT), Nagpur India (1980-1993), and University of Melbourne (1993-1995).

Dr Andrew Lewis is a Senior Research Specialist in Research Computing Services and an Adjunct Senior Lecturer in ICT at Griffith University. Prior to this appointment he worked in industrial applied research with BHP Billiton. His research interests include: parallel optimisation algorithms for large numerical simulations, including gradient descent, direct search methods, evolutionary programming, particle swarm and ant colony systems, multi-objective optimisation techniques for engineering design, and parallel, distributed and grid computing methods. He has numerous publications in the area of optimisation algorithms and applications.

Professor Abdul Sattar is the founding Director of the Institute for Integrated and Intelligent Systems (IIIS) and a Professor of Computer Science and Artificial Intelligence at Griffith University. He is also a Research Leader at National ICT Australia (NICTA) Queensland Research Lab (QRL), where he has held the positions of QRL Education Director (2006-08) and Leader of the Smart Applications for Emergencies (SAFE) project (2005-08), and is currently leading the QRL node of NICTA's largest project, Advanced Technologies for Optimisation and Modelling in Constraints (ATOMIC). He has been an academic staff member at Griffith University since February 1992 as a lecturer (1992-95), senior lecturer (1996-99), and professor (2000-present) within the School of Information and Communication Technology. Prior to his career at Griffith University, he was a lecturer in Physics in Rajasthan, India (1980-82), and a research scholar at Jawaharlal Nehru University, India (1982-85), the University of Waterloo, Canada (1985-87), and the University of Alberta, Canada (1987-1991).