straint function can be constructed. In most problems, such constraint information is available and the construction of the constraint functions is easier than construction of the cost function. We discussed the problem of redundant information for the reshaped fitness landscape. To address this problem, a more thorough theoretical investigation of GA's and NN's and the relationship between these methods is needed. The preliminary results of job-shop scheduling problems show that the result found by hybrid GA's is better than the result found by Hp NN's with randomly chosen initial points, and the model of an iiGA with NN outperform the model of a "standard" GA with NN. For future work, we intend to apply the technique to more complicated, highly constrained problems and combinatorial problems.

Reference

[1] H. Shirai, etc., "A Solution of Combinatorial Optimization Problem by Uniting Genetic Algorithms with Hopfield's Model," *IEEE World Congress on Comp. Intl.*, 1994, pp. 4704-4709.

[2] D. Tank and J. Hopfield, "Simple Neural Optimization Networks: An A/D Converter, Signal Decision Circuit, and a Linear Programming Circuit", *IEEE Trans. on Circuits and Systems*, Vol. CAS-33, No. 5, 1986, pp. 533-541.

[3] S. Lin, W.F. Punch, and E.D. Goodman, "Coarse-Grain Parallel Genetic Algorithms: Categorization and New Approach," *IEEE SPDP*, 1994, pp. 28-39.

[4] J. M. Zurada, *Introduction to Artificial Neural Systems*, 1992, West Publishing Company.

[5] M. Hagiwara, "Pseudo-Hill Climbing Genetic Algorithm (PHGA) for Function Optimization," *IEEE IJCNN*, 1993, pp. 713-716.

[6] A. E. Eiben, et al., "Repairing, adding constraints and learning as a means of improving GA performance on CSPs," *the 6th Belgian-Dutch Conf. on Machine Learning*, 1994.

[7] A. E. Smith and D. M. Tate, "Genetic Optimization Using a Penalty Function," *Proc. Fifth ICGA*, June 1993, pp. 499-505.

[8] J. T. Richardson and M. R. Palmer, "Some Guidelines for Genetic Algorithms with Penalty Functions," *Proc. Third ICGA*, June 1989, pp. 191-197.

[9] D. E. Van Den Bout and T. K. Miller III, "Graph Partitioning Using Annealed Neural Networks," *IEEE Trans. on Neural Networks*, VOL. 1. No. 2., June 1990, pp. 192-203.

[10] H. Muhlenbein, "Evolution in Time and Space - The Parallel Genetic Algorithm," *Foundations of Genetic Algorithms*, Ed. G. J.E. Rawlins, Morgan Kaufmann Pub. pp. 316-337.

[11] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, 1975.

[12] J. J. Hopfield and D. W. Tank, "Neural Computation of Decisions in Optimization Problems," *Biological Cybernetics*, 52, 1985, pp. 141-152.

[13] K. R. Baker, *Introduction to Sequencing and Scheduling*, 1974, John Wiley & Sons.

[14] E. G. Coffman, *Computer and Job-Shop Scheduling Theory*, 1976, John Wiley & Sons.

[15] J. F. Muth and G. L. Thompson (eds), *Industrial Scheduling*, 1963, Prentice-Hall.

formulated as an integer programming problem [13]. Let x_{ik} and t_{ijk} denote the completion time of job i on machine k and the processing time of operation j of job i on machine k. We define the linear energy function as

$$E = \sum_{i=1}^{n} x_{ik_i} + \sum_{i=1}^{n} \sum_{j=2}^{m} H_1 \times F(x_{ik} - x_{ih} - t_{ijk}) + \sum_{i=1}^{n} H_2 \times F(x_{ik} - t_{i1k})$$
$$+ \sum_{k=1}^{m} \sum_{i,p} H_3 \times (F(x_{pk} - x_{ik} + H(1 - y_{ipk}) - t_{pqk}) + F(x_{ik} - x_{pk} + Hy_{ipk} - t_{ijk}))$$

where H_1, H_2, and H_3 are large positive constants and y_{ipk} is defined as

y_{ipk} = 1 if job i precedes job p on machine k
= 0 otherwise

We apply this hybrid model to the 6/6 job-shop scheduling problem described in [15]. The number of x_{ik} variables is 36. We encode each x_{ik} using 4 bits, so the search space is $2^{4x36} = 2.2301x10^{43}$. The population size is 50 and the number of evaluations is 1000. The crossover rate and mutation rate are 0.6 and 0.001 respectively. Fig 8 shows the best result of the hybrid model, compared to the results from 1000 random initial points. The best results found by the hybrid model and randomly searching are 298.04 and 306.89 respectively. We next apply the hybrid model with an iiGA and a Hp NN to the same problem. We permute the 144-bit chromosome to 12x12. Four nodes with block sizes 1x2, 1x4, 2x1, and 4x1 are in the higher level. One node with block size 1x1 is in the lowest level. The nodes in higher levels inject their best results to the nodes in the lowest level every 20 generations. The best result we obtained was 284.62. Fig. 9 shows the experimental results.

7 Conclusion

In this paper, we discussed the problems in GA's and Hp NN's. The hybrid model introduced possesses many good characteristics compared to other schemes for solving highly constrained problems and combinatorial problems. A GA is a good global search algorithm and a Hp NN is a fast local search method. The combination of these models also eliminates the need for gradient information of the cost function, as long as the con-

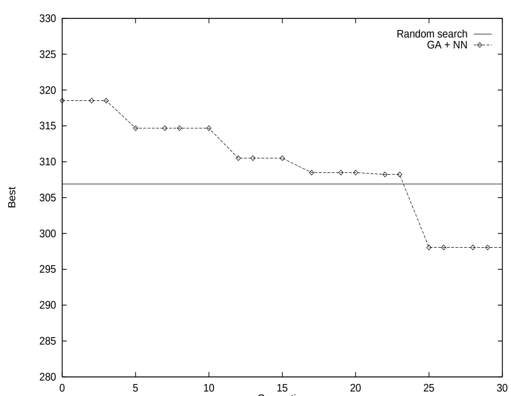


Fig 8. The result of 6/6 job-shop scheduling problem using GA + NN

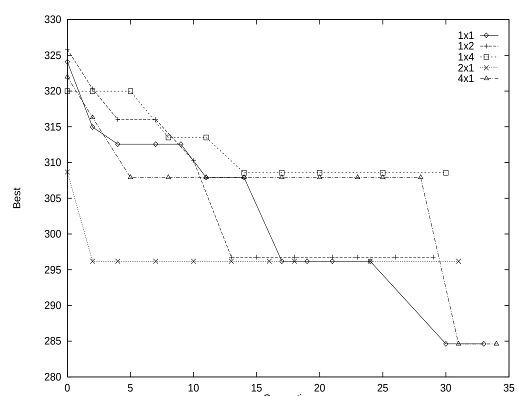


Fig 9. The result of 6/6 job-shop scheduling problem using iiGA + NN

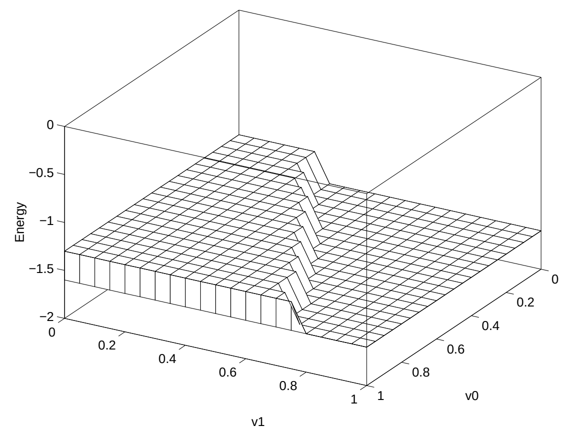


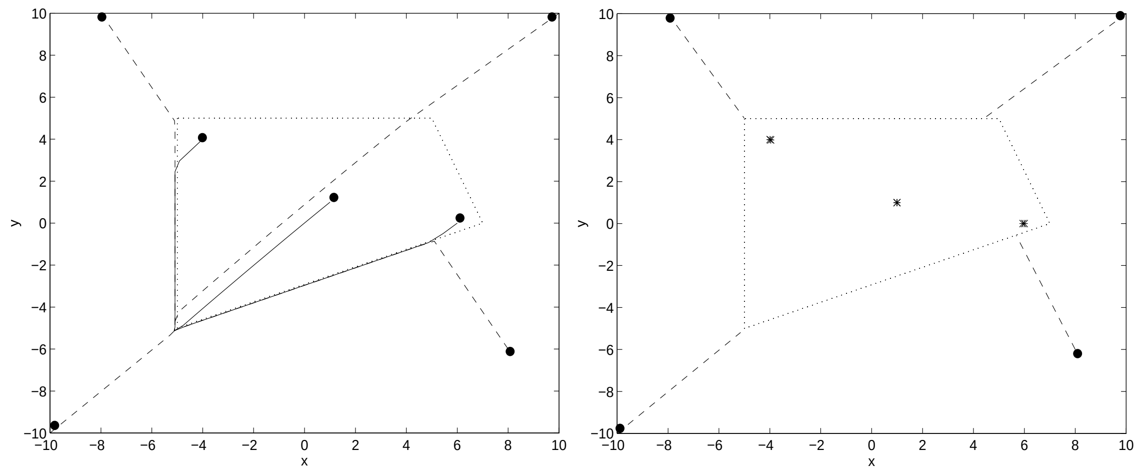
Fig 7. The reshaped fitness landscape of the 2-bit A/D converter with x = 1.8

From the structure shown in Fig 5, what this hybrid model does is to reshape the fitness landscape of the GA, since it assign the fitness values of converged configurations to the original configurations. Fig 7 shows the reshaped fitness landscape of the 2-bit A/D converter with $x = 1.8$. Compared to Fig 3, there is no slope in the reshaped fitness landscape, since the gradient information is hidden by the Hp NN. From the GA's point of view, such fitness landscapes, consisting of many large plateaus, give much redundant information. For example, in a 6/6 job-shop scheduling problem, there are $(6!)^6 = 1.3931 \times 10^{17}$ semiactive schedules. Using the linear programming method, the number of variables in a Hp NN is 36. If we encode each variable to 4 bits, then the search space of the GA is $2^{4 \times 36} = 2.2301 \times 10^{43}$. The expected number of plateaus is the number of semiactive schedules and the expected size of each plateau will be 1.6008×10^{26}. If we decrease the number of bits per variable to 2, then the search space of the GA becomes 4.7224×10^{21}. This drastically decreases the search space of the GA and the expected size of each plateau becomes 3.3898×10^{4}. Since we don't know how many bits for one variable is enough, such a decision may involve trial-and-error. One promising solution for the redundant information is to use the Injection Island GA (iiGA) architecture[3]. The basic idea of the iiGA is to divide the search space into hierarchical levels and search at all levels simultaneously. The resolutions of solutions in different levels are different. When there is an enormous amount of redundant information, it is necessary to reduce the resolution of the solution. Although the behavior of GA's in such a landscape is not well known, the preliminary results in the next section are encouraging.

Another problem in such a hybrid model is that Hp NN's are useless in unconstrained problems without the knowledge of the derivatives of the cost function. In such problems, non-gradient-like local search procedures, such as hill-climbing and the simplex method, are good choices.

6 Job-Shop Scheduling Problem

Job-shop scheduling [13],[14] is a resource allocation problem. The resources are machines and the tasks are jobs. Each job consists of several operations which are under some precedence restrictions. The determination of an optimal job-shop schedule can be



(a) (b)

Fig 6. (a) Hp NN search with cost function
(b) Hp NN search without cost function

$$-x \le 5$$

$$\frac{5}{12}x - y \le \frac{35}{12}$$

$$\frac{5}{2}x + y \le \frac{35}{2}$$

It is easy to verify that the optimal solution to this problem is $(x, y) = (-5, -5)$. If the derivative of the cost function is available, then we can include the cost function as well as constraint functions in the energy function of the Hp NN. Fig. 6a shows some local searches of a Hp NN. The dotted lines represent the constraints and the valid solution region. There are three initial points in the valid region and four initial points outside the valid region. The solid lines show that the three valid initial points converged to the optimal solution. The dashed lines show that the invalid initial points also converge to the optimal solution. For this situation, Hp NN's do a good job in local search. If the derivative of the cost function is unknown, then only the constraint function is in energy function. Using the same initial points, Fig 6b shows the trajectories of convergence. The three valid initial points don't move to other places since there is no information about the gradient. The four invalid initial points keep the same trajectories as in Fig 6a until they hit the boundary of the valid region. As we said, most optimal solutions lie on the boundary of the valid region. A Hp NN can do the search on the boundary even though the derivative of the cost function is unknown. If the optimal solution is on the boundary, the invalid initial points around the optimal solution will converge to a valid neighborhood of the optimal solution, and the GA can take advantage of such information to search out the optimal solution. If the optimum is inside the region, Hp NN's still give information about valid solutions and help GA's do the global search. For this simple problem, GA's can find the optimal solution easily with Hp NN's without using the derivatives of the cost function.

5 Problems with the Hybrid Model

Nothing is perfect in the real world. The hybrid model also suffers some problems.

4 The Hybrid Model of GA's and NN's

The problems with GA's and Hp NN's can be solved by combining the two methods. Consider the hybrid model in Fig 5. The Hp NN's are incorporated into the evaluation stage. This model allows invalid configurations in a population, but evaluates new converged valid configurations derived from them and assigns the fitness values to the original configurations without replacing them. Since a Hp NN is a deterministic method, the converged configurations can be easily reproduced. This feature makes it possible to preserve the original configurations after the Hp NN's local search, so the diversity of the population need not decrease due to this local search. Also, no information is lost due to replacement. The converged configurations are all valid if the weighting of the constraint function is harsh enough. Although the cost function may not be considered during the Hp NN's local search, the global search in the GA can make up for that weakness. Actually, one can still fine tune the Hp NN to get a better solution and help the GA do a better search. However, the parameter setting in the Hp NN is not so important, as long as validity is preserved in the converged configurations. The basic GA structure and the operators are preserved, so this model still follows the schema theorem. There is no need to construct a penalty function or a repair process in the GA since the energy function in the Hp NN already contains the constraint mechanism to ensure the validity of converged configurations.

This hybrid model can be applied in highly constrained problems and combinatorial problems as long as the constraint function can be constructed in the Hp NN. As we mentioned before, the attractive feature of hill-climbing and simplex methods as local search procedures is that they don't require the derivative of the cost function. Here we declare that

"A Hp NN is still a good local search procedure for a GA in highly constrained problems and combinatorial problems, even though the derivatives of the objective function are unknown, so long as the constraint functions can be constructed."

We give the following simple example as explanation. This is a linear programming problem described in [2]. The problem is to minimize

$$z = x + y$$

subject to

$$y \leq 5$$

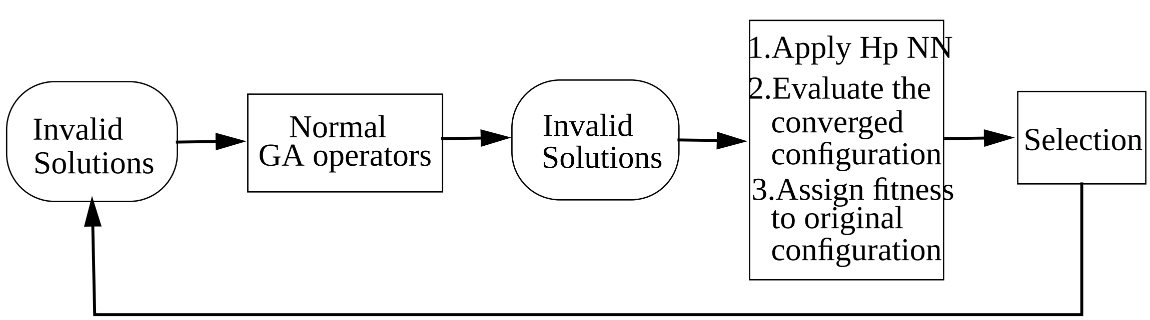


Fig 5. The hybrid model of GA's and Hp NN's

Each stable minimum is an attractor in the state space. The set of initial states which initiates the evolution terminating in one attractor is called the basin of attraction. The attractor and the basin of each attractor are determined by the energy function and the internal parameters of the NN. A Hp NN is a deterministic local search procedure. Once the initial state is selected, it will converge to the minimum in whose basin the initial state is located. Fig 3 shows the energy function of a 2-bit A/D converter [4].

$$\boldsymbol{E} = \frac{1}{2}\left(x - \sum_{i=0}^{1} V_i 2^i \right)^2 - \frac{1}{2}\sum_{i=0}^{1} 2^{2i} V_i (V_i - 1)$$

The first term is the error function of the input x and the binary output (V_0, V_1). The second term is the constraint function which force the output to be 0 or 1. This energy function is mapped to the energy function of a Hp NN as follows:

$$\boldsymbol{E} = -\frac{1}{2}\sum_{j=0}^{1} \sum_{i \neq j = 0}^{1} \left(-2^{i+j}\right) V_i V_j - \sum_{i=0}^{1}\left(-2^{(2i-1)} + 2^i x\right) V_i$$

For $x = 1.8$, there are two minima. One is (1, 0) which has energy -1.3, the other is (0, 1), which has energy -1.6. Based on the gradient of each point, we can determine the basins of these two attractors. Fig 4 shows the dynamic behavior of the A/D converter. The dotted lines represent the trajectories of different initial points. Three initial points converge to the local minimum (1, 0). For a complicated problem, it is hard to find the global optimum in a Hp NN. Simulated annealing has been applied to Hp NN's in order to avoid local minima [9]. Since the simulated annealing procedure is inherently sequential, it requires large computing time to reach a good solution. Another problem is that it is difficult to find a good balance between the cost and constraint terms in the energy function. This is similar to the problem of penalty functions in GA's.

The advantage of Hp NN's is the fast convergence to a stable minimum. If one adds constraint conditions with a large weighting to the energy function, the initial point will converge to the nearest valid point without influence from the cost function. This is different from GA's, since GA's can't adjust the configuration to a valid solution even if the penalty is harsh. This property makes Hp NN's a good candidate for a local search procedure for use with GA's to solve combinatorial problems and constrained problems.

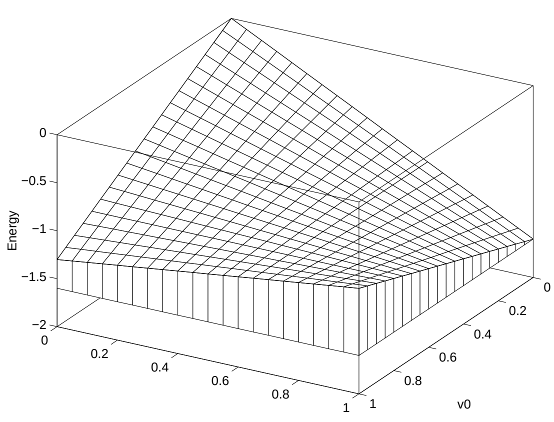


Fig 3. Energy function of 2-bit A/D converter with x = 1.8

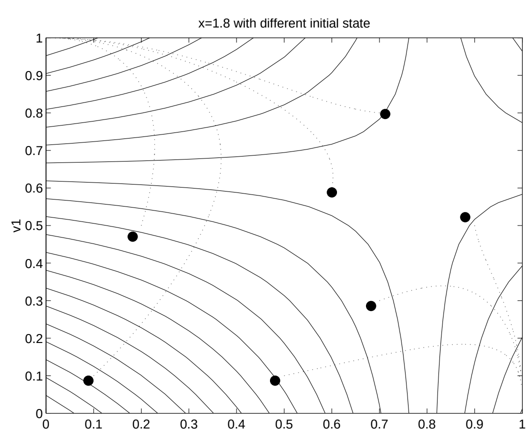


Fig 4. Dynamic behavior of 2-bit A/D converter with x = 1.8

violate the assumptions of the schema theorem and the behavior of these modified GA's becomes more unpredictable if one cannot guarantee exponentially increasing trials for the best building blocks. Also, these modified operators are problem dependent and cannot be applied to other problems. Further, in highly constrained problems, the optimal solutions tend to lie on the boundaries of the valid region. Restricting the search to the valid region makes it much more difficult to find the optima on the boundary, since the boundary isn't "spanned" in the population. The second approach (Fig 2b) allows invalid encodings in the population, but penalizes invalid solutions. In this approach, it is not easy to construct the penalty functions. The penalty functions should be harsh enough so the GA cannot converge to invalid solutions. However, if the penalty function is too severe, the information provided by invalid solutions will be lost. Care must be taken to find a balance between the validity of solution and preservation of information. One problem of this approach is that if the space of valid solutions is very small compared to the whole representation space, the probability of finding one valid solution is very small. GA's will waste all of their time on invalid solutions, and will fail to find a valid solution. The third approach (Fig 2c) utilizes "repair" to transform the invalid solutions to valid solutions [6]. One problem is that the repair process can result in the loss of important genetic material and reduce the efficiency of the search. If the repair process is deterministic and if it is easy to go back and forth between the original configuration and the new configuration, then we don't need to replace the original solution by the repaired configuration, so the information need not be lost and diversity can be preserved. (This is the same issue mentioned in the last section.) Comparing Fig 1 and Fig 2, is there a way to solve the local search problem and the validity problem at the same time without suffering the problems we mentioned? Before we answer this question, we first discuss Hp NN's.

3 Hopfield Neural Networks

A Hp NN is characterized as a highly interconnected network of simple analog processors. The network energy decreases continuously in time and the network converges to one of the stable minima in the state space. To solve problems using NN's, the network energy function is mapped to a certain objective function that needs to be minimized. A quadratic energy function is defined in [12] to solve TSP-like problems.

$$\boldsymbol{E}(\boldsymbol{V}) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} T_{ij}V_iV_j - \sum_{i=1}^{N} V_iI_i$$

where $\boldsymbol{V} = (V_1, \ldots, V_N)$ represents the state of the NN. V_i is the output of neuron $\boldsymbol{i}$. T_{ij} is the strength of the synaptic connection between neuron V_i and V_j with $T_{ii} = 0$. $\boldsymbol{I}_i$ represent the input bias current. The NN's require symmetric connections to ensure the convergence to equilibria only. A linear energy function is defined in [2] to solve linear programming problems.

$$\boldsymbol{E}(\boldsymbol{V}) = \boldsymbol{A}^t\boldsymbol{V} + \sum_{j=1}^{m} \boldsymbol{F}\left(\boldsymbol{W}_j^{\,t}\boldsymbol{V} - b_j\right)$$

where $\boldsymbol{A}^t\boldsymbol{V}$ is the cost function subject to the constraint condition $\boldsymbol{W}_j^{\,t}\boldsymbol{V} \geq b_j$. $\boldsymbol{F}$ is a nonlinear function whose derivative $\boldsymbol{f}$ is defined as

$$f(x) = \begin{cases} 0 & x \geq 0 \\ x & x < 0 \end{cases}$$

configuration. This will restrict them largely to unconstrained optimization problems and non-combinatorial optimization problems. The third problem is the diversity problem mentioned before, since they replace the original configurations with the new configurations. These problem will be revisited when we describe the hybrid model.

2.2 Validity of the Solution

The beauty of the GA is in its simple and clean structure. Regarding simplicity, a GA may consist of only selection, evaluation, crossover, and mutation. Regarding cleanliness, the operators in a GA may be totally unrelated to the problem. The most important element is that there is a fundamental theorem, the schema theorem, which guarantees that GA's give at least an exponentially increasing number of trials to the observed best building blocks. Such a simple and clean algorithm seems a near optimal procedure for searching in a solution space [11]. However, when researchers have applied GA's to highly constrained optimization problems and combinatorial optimization problems, they have often found that GA's are inefficient in finding valid and feasible solutions. Usually it is easy to generate an initial population of valid solutions, but generating only valid solutions through GA operators — i.e., crossover and mutation, may not be easy. Three approaches have been used to adapt GA's to such problems. Fig 2 shows the basic structures of these three approaches. The first approach (Fig 2a) modifies the GA operators so only valid solutions are generated. This is often applied to combinatorial optimization problems such as TSP. One problem for this approach is that the modified operators may

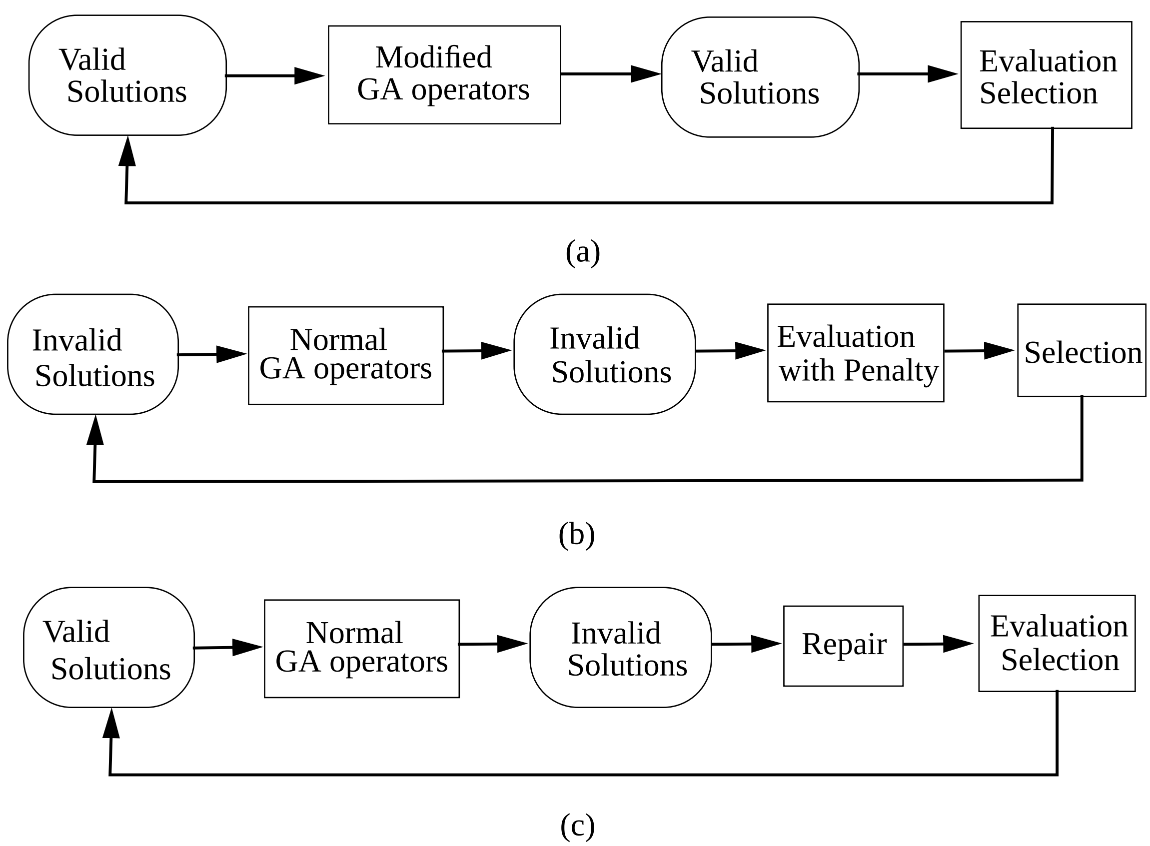


Fig 2. Three approaches for the validity problem

and constrained optimization problems. From our discussion, both problems should be considered at the same time.

2.1 Local Search Problem

GA's do not include a procedure for local search in the vicinity of an individual. The search in GA's is mainly driven by the crossover operation. Generational GA's without crossover can be seen as a sort of hill-climbing. Each new individual competes with other individuals instead of the original one, and the competition occurs only once per generation. If we see the entire population as one entity, such a GA is a population hill-climbing method. Local search is important because it could help GA's do a more rapid and comprehensive search than can crossover and/or mutation. Several methods have been proposed to deal with this problem. Mühlenbein[10] combined two random search methods, multistart hill-climbing and iterated hill-climbing, with two hill-climbing strategies — next ascent hill-climbing and steepest ascent hill-climbing, and incorporated them into a GA. Fig 1 shows the basic relationship between the local search procedure (LSP) and other GA operations. Since the local search procedure is a random search with adaptive step size, one can't reproduce the same new configurations from the original configurations. Also there is no way to go back from the new configurations to the original configurations. Due to these two restrictions, the new configurations replace the original configurations in Mühlenbein's scheme. One problem for this scheme is that if many local searches lead to the same point, the diversity will rapidly decrease in the new configurations, hence in the population. Another method proposed by Hagiwara [5] used the simplified simplex method as the local search procedure. The simplex method estimates the slope of the plane in the search space by comparing three points in the original configurations and generates a new point away from the worst point. The step size is determined by the reflection coefficient. Although this scheme is deterministic, it's not easy to reproduce the new configurations and convert the new configurations to the original because the local search of each individual is not independent. The original simplex method requires n+1 points to estimate the slope in n-dimensional space. Since this original method needs large computing time if n is large, the author uses just 3 points to estimate the slope. This will produce unaccepted levels of error in a high-dimensionality space.

The advantage of both schemes is that they don't require derivatives of an objective function. This feature is attractive for problems in which derivatives of the object function are unknown. If the derivative of the object function can be determined, the advantage vanishes and both schemes become inferior to gradient-type methods, since they require much computing time to evaluate each new configuration during a local search. Another problem with both schemes is that they do not consider the validity of the new

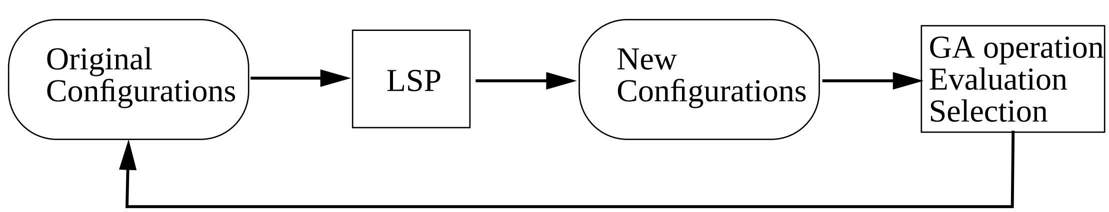


Fig 1. Local search procedure in GA's

A Hybrid Model Utilizing Genetic Algorithms and Hopfield Neural Networks for Function Optimization

ABSTRACT

This paper discusses some problems existing in Genetic Algorithms (GA's) and Hopfield Neural Networks (Hp NN's). In GA's, one problem is the slow speed of local search. Another is the necessity to modify the breeding operators or to apply penalties in the fitness function in order to work efficiently on constrained or combinatorial optimization problems. In Hp NN's, one problem is the lack of global search ability. Another is the difficulty of fine tuning the parameters to ensure a good and valid solution. In this respect, GA's and Hp NN's are somewhat complementary to each other. A hybrid model incorporating a GA and a Hp NN is proposed, and applied to a job-shop scheduling problem. The preliminary results show, first, that the hybrid model outperforms Hp NN's with randomly chosen initial points, and second, that the hybrid model, using an Injection Island GA (iiGA) and Hp NN, can alleviate the redundancy problem and outperform a more classical model of GA's with NN's.

1 Introduction

Genetic algorithms are now recognized as an effective optimization method in many areas, such as image processing, VLSI circuit layout, composite material design, and scheduling problems. The utilization of the linkage among population searches makes the GA a good global search method. However, there still exist some problems in GA's. One is the slow speed of local search. Another problem is that it's necessary to modify the breeding operators or apply penalty in fitness function for the combinatorial optimization problems and constrained optimization problems; otherwise, GA's act poorly in such problems. Neither task is easy. Artificial Neural Networks, particularly the Hopfield model [2], have been applied to a wide variety of problems, such as the traveling salesman problem, job shop scheduling, linear programming, nonlinear programming, and others. By using gradient descent of the energy function, the convergence to one of the stable minima in the state space is ensured. Such gradient-type neural networks don't guarantee convergence to the global minimum because of their lack of global search ability. Another problem in Hp NN's is that many parameters must be properly selected and tuned even to ensure convergence to a valid solution. Furthermore, it is more difficult to fine tune NN's to get a good solution. In some sense, GA's and Hp NN's are complementary to each other. In this paper, we probe both approaches and expand the hybrid model proposed by Shirai et al.[1] to solve more general problems. We also discuss some problems existing in the hybrid model and propose a promising solution. Finally, we apply this hybrid model to a job-shop scheduling problem as a demonstration.

2 Two Problems with GA's

In this section, we discuss two problems with GA's — that is, the slow speed of local rapid search, and and a difficulty encountered with combinatorial optimization problems

A Hybrid Model Utilizing Genetic Algorithms and Hopfield Neural Networks for Function Optimization

Shyh-Chang Lin

Genetic Algorithm Research and Applications Group
Michigan State University
112 Engineering Building
Lansing, MI 48824
Email: linshyh@egr.msu.edu

W.F. Punch III

Genetic Algorithm Research and Applications Group
Department of Computer Science
Michigan State University
Email: punch@cps.msu.edu

E.D. Goodman

Genetic Algorithm Research and Applications Group
Case Center for Computer Aided Engineering and Manufacturing
Michigan State University
Email: goodman@egr.msu.edu

ABSTRACT

This paper discusses some problems existing in Genetic Algorithms (GA's) and Hopfield Neural Networks (Hp NN's). In GA's, one problem is the slow speed of local search. Another is the necessity to modify the breeding operators or to apply penalties in the fitness function in order to work efficiently on constrained or combinatorial optimization problems. In Hp NN's, one problem is the lack of global search ability. Another is the difficulty of fine tuning the parameters to ensure a good and valid solution. In this respect, GA's and Hp NN's are somewhat complementary to each other. A hybrid model incorporating a GA and a Hp NN is proposed, and applied to a job-shop scheduling problem. The preliminary results show, first, that the hybrid model outperforms Hp NN's with randomly chosen initial points, and second, that the hybrid model, using an Injection Island GA (iiGA) and Hp NN, can alleviate the redundancy problem and outperform a more classical model of GA's with NN's.