An Efficient Genetic Algorithm for Solving the Generalized Traveling Salesman Problem

Oliviu Matei
Dept. of Electrical Engineering
North University of Baia Mare
Baia Mare, Romania
Email: oliviu.matei@holisun.com

Petrică Pop
Dept. of Mathematics and Computer Science
North University of Baia Mare
Baia Mare, Romania
Email: petrica.pop@ubm.ro

***Abstract*—The generalized traveling salesman problem (GTSP) is a generalization of the classical traveling salesman problem. The GTSP is known to be an NP-hard problem and has many interesting applications. In this paper we present a local-global approach for the generalized traveling salesman problem and as well an efficient algorithm for solving the problem based on genetic algorithms. Computational results are reported for Euclidean TSPlib instances and compared with the existing ones. The obtained results point out that our GA is an appropriate method to explore the search space of this complex problem and leads to good solutions in a short amount of time.**

***Keywords*-genetic algorithms, generalized traveling salesman problem**

I. Introduction

Classical combinatorial optimization problems can be generalized in a natural way by considering a related problem relative to a given partition of the nodes of the graph into node sets. In the literature one finds generalized problems such as the generalized minimum spanning tree problem, the generalized traveling salesman problem, the generalized Steiner tree problem, the generalized (subset) assignment problem, etc. These generalized problems typically belong to the class of NP-complete problems, are harder than the classical ones and nowadays are intensively studied due to the interesting properties and applications in the real world.

The problem of finding a minimum cost Hamiltonian circuit or cycle of a given graph can be generalized in a natural way by considering instead of nodes node sets (clusters) and asking for finding a minimum cost Hamiltonian circuit or cycle which includes exactly one node from each cluster. This problem is called the generalized traveling salesman problem (GTSP) and it was introduced independently by Henry-Labordere [7], Srivastava et al. [21] and Saskena [19].

A variant of the GTSP is the problem of finding a minimum cost Hamiltonian tour including *at least* one vertex from each cluster. This version of the GTSP was introduced by Laporte and Nobert [8] and by Noon and Bean [11].

Because of the complexity of the problem, several heuristic and meta-heuristic algorithms have been proposed for solving the GTSP: an efficient composite heuristic [18], reinforcing ant colony system [12], a random key genetic algorithm [20], a memetic algorithm [6], etc.

The *Genetic Algorithms* (GA) were introduced by Holland in the early 1970s, and were inspired by Darwin's theory. The idea behind GA is to model the natural evolution by using genetic inheritance together with Darwin's theory. In GA, the *population* consists of a set of *solutions* or *individuals* instead of chromosomes. A *crossover* operator plays the role of reproduction and a *mutation* operator is assigned to make random changes in the solutions. A selection procedure, simulating the natural selection, selects a certain number of *parent* solutions, which the crossover uses to generate new solutions, also called *offspring*. At the end of each iteration the offspring together with the solutions from the previous generation form a new generation, after undergoing a selection process to keep a constant population size. The solutions are evaluated in terms of their *fitness values* identical to the fitness of individuals. GA have seen a widespread use amongst modern meta-heuristics, and several applications to combinatorial optimization problems have been reported, see [2], [20].

The aim of this paper is to describe a new approach to the GTSP based on distinguishing between local and global connections and to develop a new GA in order to solve the problem, which is competitive with the meta-heuristic approaches that have been already proposed, in terms of computing time and quality solution. Computational results for benchmarks problems from TSP library are reported.

II. Definition of the Generalized Traveling Salesman Problem

Let $G = (V, E)$ be an n-node undirected graph whose edges are associated with non-negative costs. We will assume without loss of generality that the graph G is a complete graph (if there is no edge between two nodes, we can add it with an infinite cost).

Let $V_1, ..., V_m$ be a partition of V into m subsets called *clusters* (i.e. $V = V_1 \cup V_2 \cup ... \cup V_m$ and $V_l \cap V_k = \emptyset$ for all $l, k \in \{1, ..., m\}$). We denote the cost of an edge

978-1-4244-8230-6/10/$26.00 ©2010 IEEE

$e = \{i,j\} \in E$ by c_{ij}. Let $e = \{i,j\}$ be an edge with $i \in V_l$ and $j \in V_k$. The *generalized traveling salesman problem* asks for finding a minimum-cost tour H spanning a subset of nodes such that H contains exactly one node from each cluster V_i, $i \in \{1,...,m\}$. Therefore the GTSP involves the following two related decisions:

- choosing a node subset $S \subseteq V$, such that $|S \cap V_k| = 1$, for all $k = 1,...,m$.
- finding a minimum cost Hamiltonian cycle in the subgraph of G induced by S.

We will call such a cycle a *generalized Hamiltonian tour*. An example of a Hamiltonian tour for a graph with the nodes partitioned into 6 clusters is presented in figure 1.

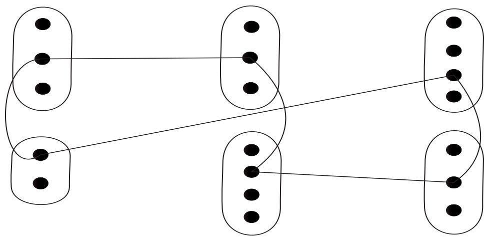

Figure 1. Example showing a generalized Hamiltonian in the graph $G = (V,E)$

Both problems GTSP and the at least version of the GTSP are NP-hard, as they reduce when each cluster consists of exactly one node ($|V_i| = 1, \forall\, i = 1,...,m$) to traveling salesman problem which is known to be an NP-hard problem, see Garey and Johnson [4].

The GTSP has several applications to location problems, telecommunication problems, postal routing, computer file sequencing, order picking in warehouses, railway optimization, etc. More information on these problems and their applications can be found in Fischetti, Salazar and Toth [3], Laporte and Nobert [8], Pop et al. [15], Saskena [19], Snyder and Daskin [20], etc.

III. The local-global approach to the Generalized Traveling Salesman Problem

Let G' be the graph obtained from G after replacing all nodes of a cluster V_i with a super-node representing V_i. We will call the graph G' the global graph. For convenience, we identify V_i with the super-node representing it. Edges of the graph G' are defined between each pair of the graph vertices $V_1, \ldots, V_m$.

The local-global approach was introduced by Pop [13] in the case of the generalized minimum spanning tree problem. Since then this approach was successfully applied for in order to obtain exact algorithms, strong mixed-integer programming formulations, heuristic and meta-heuristic algorithms for several generalized network design problems, see [14], [15], [5].

The local-global approach to the GTSP aims at distinguishing between *global connections* (connections between clusters) and *local connections* (connections between nodes from different clusters). As we will see, given a sequence in which the clusters are visited (global Hamiltonian tour) it is rather easy to find the corresponding best (w.r.t. cost minimization) generalized Hamiltonian tour.

There are several generalized Hamiltonian corresponding to a global Hamiltonian tour. Between these generalized Hamiltonian tours there exist one called the best generalized Hamiltonian tour (w.r.t. cost minimization) that can be determined either by building a layered network having $m+1$ clusters corresponding to the clusters and one additional duplicating the first cluster and showing that there exists a one-to-one correspondence between Hamiltonian tours visiting the clusters according to a given sequence and the paths in the layered network (see Pop *et al.* [15] for further details) or by solving a linear integer program as we will describe in what it follows.

We introduce the following binary variables $x_e \in \{0,1\}$, $e \in E$ and $z_i \in \{0,1\}$, $i \in V$ to indicate whether an edge e respectively a node i is contained in the Hamiltonian tour, and the variables y_{ij}, $(i,j \in \{1,\ldots,m\})$ to describe the global connections. So $y_{ij} = 1$ if cluster V_i is connected to cluster V_j and $y_{ij} = 0$ otherwise. We assume that y represents a Hamiltonian tour. The convex hull of all these y-vectors is generally known as the Hamiltonian tour polytope on the global graph G'.

Following Miller *et al.* [10] this polytope, denoted by P_{TSP}, can be represented by the following polynomial number of constraints:

$$\sum_{\substack{j=1,\\ j\neq i}}^{m} z_{ij} = 1, \quad \forall i \in \{1,...,m\} \tag{1}$$

$$\sum_{\substack{i=1,\\ i\neq j}}^{m} z_{ij} = 1, \quad \forall j \in \{1,...,m\} \tag{2}$$

$$u_i - u_j + (m-1)z_{ij} \leq m-2, \forall i,j \in \{2,...,m\},\ i \neq j \tag{3}$$

$$1 \leq u_i \leq m-1, \qquad \forall i \in \{2,...,m\} \tag{4}$$

where the extra variables u_i represent the sequence in which city i is visited, $i \neq 1$.

The first two constraints guarantee that each of the nodes (cities) is visited exactly once. The constraints denoted by (3) ensure that the solution contains no sub-tour on a set of nodes S with $|S| \leq m-1$ and hence, no sub-tour involving less than m nodes and finally the constraints denoted by (4) ensure that the u_i variables are uniquely defined for any feasible tour.

If the vector y describes a Hamiltonian tour on the global graph G', which we shall refer as the global Hamiltonian tour, then the corresponding best (w.r.t. minimization of the costs) generalized Hamiltonian tour can be obtained by solving the following 0-1 programming problem:

$$\begin{array}{lll} \min & \sum_{e\in E} c_e x_e & \\ s.t. & z(V_k) = 1, & \forall\ k \in K = \{1, ..., m\} \\ & x(V_l, V_r) = y_{lr}, & \forall\ l, r \in K = \{1, ..., m\}, l \neq r \\ & x(i, V_r) \leq z_i, & \forall\ r \in K, \forall i \in V \setminus V_r \\ & x_e, z_i \in \{0, 1\}, & \forall\ e = (i, j) \in E, \forall i \in V, \end{array}$$

where

$$x(V_l, V_r) = \sum_{i\in V_l, j\in V_r} x_{ij} \text{ and } x(i, V_r) = \sum_{j\in V_r} x_{ij}.$$

For given y, we denote the feasible set of the linear programming relaxation of this program by $P_{local}(y)$.

Pop et al. [14] proved that if y is the 0-1 incidence vector of a spanning tree of the contracted graph then the polyhedron $P_{local}(y)$ is integral. But as we are going to show in the next example a similar result does not hold when z is the incidence vector of the Hamiltonian tour, namely if z is the 0-1 incidence vector of a Hamiltonian tour on the contracted graph then $P_{local}(y)$ may not be integral.

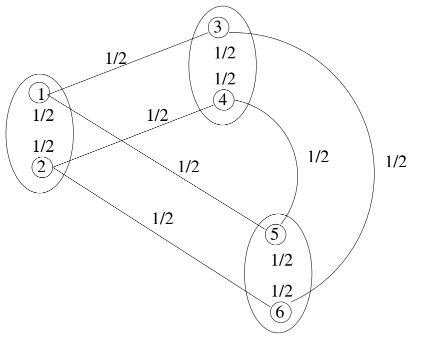

Figure 2. Example showing that $P_{local}(z)$ may have fractional extreme points

If the lines drawn in the figure 2 (i.e., $\{1, 3\}$, $\{2, 4\}$, etc.) have cost 1 and all the other lines (i.e., $\{1, 4\}$, $\{2, 3\}$, etc.) have cost M 1, then $y \equiv \frac{1}{2}$ and $x \equiv \frac{1}{2}$ on the drawn lines is an optimal solution of $P_{local}(z)$, showing that the polyhedron $P_{local}(z)$ is not integral.

The observations presented so far lead to a new model of problem, called the local-global formulation of the GTSP as an 0-1 mixed integer programming problem:

$$\begin{array}{ll} \min & \sum_{e\in E} c_e x_e \\ s.t. & z \in P_{TSP} \\ & (x, z) \in P_{local}(y) \\ & y_{lr} \in \{0, 1\}, \quad 1 \leq l, r \leq m. \end{array}$$

This new formulation of the GTSP was obtained by incorporating the constraints characterizing P_{TSP}, with $y \in \{0, 1\}$, into $P_{local}(y)$.

IV. The genetic algorithm for solving the GTSP

We present in this section a genetic algorithm for solving the GTSP.

A. Genetic Representation

An individual is represented as a pair (C, N), where $C = (V_1, V_2, ..., V_m)$ represents the sequence of clusters, described as Hamiltonian cycles. $N = (N_1, N_2, ..., N_m)$ represents the set of nodes selected from each cluster, meaning that the node N_k is the node selected from the cluster V_k. The individual $I = (C, N)$, with $C = (1, 5, 3, 4, 2)$ and $N = (10, 23, 31, 44, 52)$ represents the tour which passes the clusters in the following order:

$$1 - 5 - 3 - 4 - 2 - 1$$

and the nodes as follows:

$$10 - 52 - 31 - 44 - 23 - 10$$

B. Initial population

The construction of the initial population is of great importance to the performance of GA, since it contains most of the material the final best solution is made of.

In our algorithm the initial population is generated separately for the set of clusters, respectively for the set of nodes selected from the clusters. Each cluster V_k is generated randomly from the set of the m clusters. The set of nodes N selected from the clusters is generated based on the Monte Carlo method. The first element N_1 is chosen randomly from the set of the nodes belonging to the cluster V_1. Each further node N_k, with $k \in \{2, ..., m\}$, is selected randomly inverse proportionally with the distance (cost) from its predecessor node: $d_k = c(N_{k-1}, N_k)$. The probability for each node to be selected is $p_i = \frac{d(N_i, N_k)}{D}$, where $D = \sum_{k=1}^{|V_i|} d_k$. It is obvious that the probability for a node to be chosen increased as the distance to the node N_k decreases.

We define a summing probability:

$$q_i = \sum_{i=1}^{|V_i|} p_i$$

A random number $r \in [0, 1]$ is chosen, which gives the node to be selected as being i which holds $r \in [q_i, q_{i+1})$.

Based on computational experiments we observed that this method assures an initial population with a fitness with 20% better than a population generated entirely randomly.

C. The fitness value

Every solution has a fitness value assigned to it, which measures its quality. In our case the, the fitness value of a feasible solution (generalized Hamiltonian tour) of the GTSP is given by the total cost of the edges selected in the Hamiltonian tour i.e. the objective function of the integer programming model presented in the previous section. The aim is to find the generalized Hamiltonian tour with minimum cost.

D. Genetic operators

1) Crossover: Two parents are selected from the population by the binary tournament method, i.e. the individuals are chosen from the population at random.

Offspring are produced from two parent solutions using the following crossover procedure described by Matei in [9]: it creates offspring which preserve the order and position of symbols in a subsequence of one parent while preserving the relative order of the remaining symbols from the other parent. It is implemented by selecting a random cut point. The crossover operator for the set of nodes N is straightforward. The recombination for the route C require some further explanations. First, the symbols before the cut points copied from the first parent into the offspring. Then, starting just after the cut-point, the symbols are copied from the second parent into the offspring, omitting any symbols that were copied from the first parent. The second offspring is produced by swapping round the parents and then using the same procedure.

Next we present the application of the proposed crossover for the cluster route C. We assume two well-structured parents chosen randomly, with the cutting point between 2 and 3:

$$
\begin{array}{lccccccc}
C_1 = & 1 & 4 & | & 2 & 3 & 5 \\
C_2 = & 2 & 1 & | & 5 & 4 & 3
\end{array}
$$

The sequences before the cutting-point is copied into the two offspring:

$$
\begin{array}{lccccccc}
O_1 = & 1 & 4 & | & \text{x} & \text{x} & \text{x} \\
O_2 = & 2 & 1 & | & \text{x} & \text{x} & \text{x}
\end{array}
$$

The nodes of the parent C_1 are copied into the offspring O_2 if O_2 does not contain clusters in the as the nodes of C_1. Note the the cluster 2 is already represented in O_2. And the same holds for O_1 and C_2:

$$
\begin{array}{lccccccc}
O_1 = & 1 & 4 & | & 5 & \text{x} & 3 \\
O_2 = & 2 & 1 & | & \text{x} & 3 & 5
\end{array}
$$

The already existing clusters are replaced by their correspondents from the other parent:

$$
\begin{array}{lccccccc}
O_1 = & 1 & 4 & | & 5 & 2 & 3 \\
O_2 = & 2 & 1 & | & 4 & 3 & 5
\end{array}
$$

2) Mutation: We use in our GA two random mutation operators: the first one (intra-route mutation) selects randomly a cluster to be modified and replaces its current node by another one randomly selected from the same cluster and the second one (inter-route mutation) is a swap operator, it picks two random locations in the solution vector and swaps their values. Similar mutation operators are reported by Pop et al. in [17].

The developed GA uses the steady-state approach, in which eligible offspring enter the population as soon as they are produced, with inferior individuals being removed at the same time, so that the size of the population remains constant.

3) Selection: The selection process is deterministic. The first selection is $(\mu + \lambda)$, where μ parents produce λ offspring. The new population of $(\mu + \lambda)$ is reduced again to μ individuals by a selection based of the "survival of the fittest" principle. In other words, parents survive until they are suppressed by better offspring. It might be possible for very well adapted individuals to survive forever. This feature yields some deficiencies of the method [1]:

1) In problems with optimum moving over time, a $(\mu+\lambda)$ selection may get stuck at an outdated good location if the internal parameter setting becomes unsuitable to jump to the new field of possible improvements.
2) The same happens if the measurement of the fitness or the adjustment of the object variables are subject to noise, e.g. in experimental settings.

In order to avoid effects, Schwefel investigated the properties of (μ, λ), selection, where μ parent produce λ $(\lambda > \mu)$ and only the offspring undergo selection. In other words, the lifetime of every individual is limited to only one generation. The limited life span allows to forget the inappropriate internal parameter settings. This may lead to short periods of recession, but it avoids long stagnation phases due to unadapted strategy parameters [22]. The $(\mu+\lambda)$ and (μ, λ) selection fit into the same formal framework with the only difference being the limited life time of individuals in (μ, λ) method.

E. Genetic parameters

The genetic parameters are very important for the success of a GA, equally important as the other aspects, such as the representation of the individuals, the initial population and the genetic operators. The most important parameters are:

- the population size μ has been set to 5 times the number of clusters. This turned out to be the best number of individuals in a generation.
- the intermediate population size λ was chosen twice the size of the population: $\lambda = 2 \cdot \mu$.
- mutation probability was set at 5%.

V. Computational results

The performance of the proposed GA for *GTSP* was tested on nine benchmark problems drawn from *TSPLIB* library test problems. These problems contain between 198 and 442 nodes, which are partitioned into a given number of clusters.

Originally the set of nodes in these problems are not divided into clusters. The CLUSTERING procedure proposed by Fischetti *et al.* [3] divide data into node-sets. This procedure sets the number of clusters $s = [\frac{n}{5}]$, identifies the s farthest nodes from each other and assigns each remaining node to its nearest center. The solution proposed in this paper is able to handle any cluster structure.

The testing machine was an Intel Dual-Core 1,6 GHz and 1 GB RAM. The operating system was Windows XP Professional. The algorithm was developed in Java, JDK 1.6.

The parameters of the GA are critical as in all other metaheuristic algorithms. Currently there is no mathematical analysis developed to give the optimal parameter in each situation. Based on computational experiments in our genetic algorithm for *GTSP*, the values of the parameters were chosen as follows: population size $5 \cdot m$, the number of offspring minimum $2 \cdot \mu$, the number of generations between 400 and 1000, the intra-route mutation rate 5% and the inter-route mutation rate 5%.

In the next tables are shown the computational results obtained for solving the *GTSP* using our proposed **GA** algorithm comparing with the random key genetic algorithm [20], a memetic algorithm [6].

Table 1. Best Values - *MA*, *RK-GA* and *GA* algorithms for *GTSP*

Problem	*m*	*n*	*MA*	*RK-GA*	*GA*
40d198	40	198	10557	10557	10557
40kroa200	40	200	13406	13406	13406
40krob200	40	200	13111	13111	13111
46gr229	46	229	71641	71641	71832
53gil262	53	262	1013	1013	1014
60pr299	60	299	22615	22615	22618
80rd400	80	400	6361	6361	6389
84fl417	84	417	9651	9651	9651
89pcb442	89	442	21657	21657	21665

The table 1 contains the name of the test problem, the number of clusters, the number of nodes and the best costs obtained with the random key genetic algorithm [20], a memetic algorithm [6] and our proposed genetic algorithm.

Analyzing the computational results, it turns out that overall the proposed GA is competitive with the best metaheuristics proposed in the literature developed by Snyder and Daskin [20] and Gutin and Karapetyan [6] for the GTSP in terms of solution quality.

VI. Conclusion

The aim of this paper is to present a new approach based on distinguishing between global connections (connections between clusters) and local connections (connections between nodes from different clusters), an integer programming formulation of the problem and an efficient genetic algorithm for solving the generalized traveling salesman problem.

The GA that has been described here performs well in terms of solution quality in comparison with two of the most efficient metaheuristics developed for the GTSP by Snyder and Daskin [20] and Gutin and Karapetyan [6]. Therefore, it has been demonstrated that our GA is an effective approach for solving the generalized traveling salesman problem.

References

[1] T. Back, F. Hoffmeister and H. Schwefel, A survey of evolution strategies, in Proc. of the 4th International Conference on Genetic Algorithms, San Diego, CA, July, 1991, Morgan Kaufman.

[2] Barrie M. Baker, M.A. Ayechew, A genetic algorithm for the vehicle routing problem, Computers & Operations Research, Vol. 30, pp. 787-800, 2003.

[3] M. Fischetti, J.J. Salazar and P. Toth, A branch-and-cut algorithm for the symmetric generalized traveling salesman problem, Operations Research, Vol. 45, pp. 378-394, 1997.

[4] M.R. Garey and D.S. Johnson, Computers and intractability: A guide to the theory of NP-completeness, W. H. Freeman & Co., New York, NY, 1990.

[5] B. Hu, M. Leitner and G. Raidl, Combining variable neighborhood search with integer linear programming for the generalized minimum spanning tree problem, Journal of Heuristics, Vol. 14(5), pp. 473-499, 2008.

[6] G. Gutin and D. Karapetyan, A memetic algorithm for the generalized traveling salesman problem, Natural Computing 9, pp. 47-60, 2010.

[7] Henry-Labordere, The record balancing problem: A dynamic programming solution of a generalized traveling salesman problem, RAIRO Operations Research, B2, pp. 43-49, 1969.

[8] G. Laporte and Y. Nobert, *Generalized Traveling Salesman through n sets of nodes: an integer programming approach*, INFOR, Vol. 21, pp. 61-75, 1983.

[9] O. Matei, Evolutionary Computation: Principles and Practices, Risoprint, 2008.

[10] C.E. Miller, A.W. Tucker and R.A. Zemlin, Integer programming formulation of travelling salesman problems, J. ACM, Vol. 3, pp. 326-329, 1960.

[11] C. E. Noon, J. C. Bean, A lagrangian based approach for the asymmetric generalized traveling salesman problem, Operations Research, Vol. 39, pp. 623-632, 1991.

[12] C. Pintea, P.C. Pop and C. Chira, Reinforcing Ant Colony System for the Generalized Traveling Salesman Problem, Proc. of International Conference Bio-Inspired Computing-Theory and Applications (BIC-TA), Wuhan, China, Vol. Evolutionary Computing Section, pp. 245-252, 2006.

[13] P.C. Pop, The Generalized Minimum Spanning Tree Problem, PhD thesis, University of Twente, The Netherlands, 2002.

[14] P.C. Pop, W. Kern and G. Still, A New Relaxation Method for the Generalized Minimum Spanning Tree Problem, European Journal of Operational Research, 170, 900-908, 2006.

[15] P.C. Pop, C. Pop Sitar, I. Zelina and I. Tascu, Exact algorithms for generalized combinatorial optimization problems, in Proc. COCOA Conference, Xi'an, China, Lecture Notes in Computer Science, Vol. 4616, pp. 154-162, 2007.

[16] P.C. Pop, A survey of different integer programming formulations of the generalized minimum spanning tree problem, Carpathian Journal of Mathematics, Vol. 25, No. 1, pp. 104-118, 2009.

[17] P.C. Pop, O. Matei, C. Pop Sitar and C. Chira, A Genetic Algorithm for Solving the Generalized Vehicle Routing Problem, to appear in Proc. of HAIS, Lecture Notes in Computer Science, 2010.

[18] J. Renaud and F.F. Boctor, An efficient composite heuristic for the Symmetric Generalized Traveling Salesman Problem, European Journal of Operational Research, Vol. 108(3), pp. 571-584, 1998.

[19] J. P. Saskena, Mathematical model of scheduling clients through welfare agencies, Journal of the Canadian Operational Research Society, Vol. 8, pp. 185-200, 1970.

[20] L.V. Snyder and M.S. Daskin, A random-key genetic algorithm for the generalized traveling salesman problem, European Journal of Operations Research, Vol. 174, pp. 38-53, 2006.

[21] Srivastava, S. S. S. Kumar, R. C. Garg, and P. Sen, Generalized traveling salesman problem through n sets of nodes, CORS Journal, Vol. 7, pp. 97-101, 1969.

[22] H.P. Schwefel, Collective phenomena in evolutionary systems, in Proc. of 31st Annual Meetting of the International Society for General System Research, pp. 1025-1033, 1987.