Research Article

A P-Based Hybrid Evolutionary Algorithm for Vehicle Routing Problem with Time Windows

Yunyun Niu,¹ Juanjuan He,² Zhongyu Wang,² and Jianhua Xiao³

¹ School of Electronic Engineering and Computer Science, Peking University, Beijing 100871, China

² Key Laboratory of Image Processing and Intelligent Control, Huazhong University of Science and Technology, Wuhan 430074, China

³ The Research Center of Logistics, Nankai University, Tianjin 300071, China

Correspondence should be addressed to Jianhua Xiao; jhxiao@nankai.edu.cn

Received 29 April 2014; Accepted 15 June 2014; Published 2 July 2014

Academic Editor: Yang Tang

Copyright © 2014 Yunyun Niu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The ability to solve optimization problems using membrane algorithms is an important application of membrane computing. This work combines membrane systems and genetic operators to build an approximated algorithm for the vehicle routing problem with time windows. The algorithm is based on a tissue-like membrane structure combined with cell separation rules and communication rules; during such processes membranes collect and disperse information. Genetic operators are used as the system's subalgorithms. We also design a special improvement strategy to speed up the search process in subsystems. The experimental results show that the solution quality from the proposed algorithm is competitive with other heuristic or metaheuristic algorithms in the literature.

1. Introduction

The membrane computing is a branch of natural computing: it is inspired by the structure and the function of living cells [1] as well as the organization of cells in tissues, organs, and other higher order structures. The computational devices in membrane computing are called P systems, which consist of three parts: the membrane structure, the objects in the membrane regions, and the evolution rules. Roughly speaking, there are three types of P systems, that is, cell-like P systems [2, 3], tissue-like P systems [4, 5], and neuron-like P systems [6, 7]. They provide distributed parallel and nondeterministic computing devices. Membranes delimit space to different regions, and objects evolve according to evolution rules in each region.

The membrane algorithm, which was introduced by Nishida [8] in 2005, is a membrane computation optimization algorithm. It combines the concepts of P systems with some local approximate algorithms. In the framework of membrane computing, the objects and the membranes evolve step by step according to specific evolution rules such as rewriting, communications, and division rules. This process can describe many of the evolutionary concepts of living

cells. A membrane algorithm is also called a P-based algorithm. In a membrane algorithm, the membranes separate space into several regions, where subalgorithms and tentative solutions are placed as well as transporting mechanism and transforming mechanism of membrane structure. The subalgorithms are always selected from heuristics operators. From a computational point of view, a P system provides a good framework for an approximation algorithm because of its hierarchical structure and intrinsic parallelism. Compared with a traditional hybrid optimization algorithm, the membrane algorithm contains rich changes in algorithm structure and stochastic processes.

Proper architecture design is an important research aspect for membrane algorithms. The membrane system proposed by Nishida (see [8]) includes a nested membrane structure (sometimes the structure can be changed during the computation process), while a local search algorithm supplies the evolution scheme for each region. Tentative solutions were placed in the regions delimited by the membranes. When computing the results for the different regions, exchanges are included by applying transportation and membrane structure transformation mechanisms. The membrane algorithm designed in [8] efficiently deals with the traveling

salesman problem (TSP). In [9], the minimum storage problem was successfully solved using a membrane algorithm. In [10], Zaharie and Ciobanu discussed the similarities between distributed evolutionary algorithms and P systems. They suggested novel variants of distributed evolutionary algorithms to solve numeric optimization problems. Several approaches combining (nested membrane structure) NMS with genetic algorithms have been presented to solve numeric optimization problems [11], traveling salesman problems [12], controller design problems for a marine diesel engine [13], gasoline blending and scheduling problems [14], and optimal control problems for a time-varying unstable plant [15]. A quantum-inspired evolutionary algorithm based on P systems (QEPS) was proposed in [16]. It incorporated a one-level membrane structure (OLMS) and a quantum-inspired evolutionary algorithm. The difference between the performances of a membrane algorithm with OLMS and a membrane algorithm with a nested membrane structure was also analyzed. In [17], a modified QEPS with a local search was discussed by analyzing radar emitter signals. In [18], OLMS was integrated with DE and ant colony optimization methods to solve numeric optimization and traveling salesman problems. These studies clearly illustrate that membrane systems can be used to improve the performance of many metaheuristic search methods and that they sometimes balance exploration and exploitation efficiently [19, 20]. To date, there are very few papers that consider a tissue P system to be a membrane structure of P-based algorithms [21], but our paper is a contribution to this line of research.

The vehicle routing problem with time windows (VRPTW) is an NP-hard problem that can be applied to many distribution systems [22–25], such as the rail distribution system, the school bus routing system, and the mail and newspaper delivery system [26]. In the literature, many heuristic or metaheuristic approaches have been proposed for solving the VRPTW [27]. In [28], tabu search was first applied to the VRPTW. Other studies on tabu searches for the VRPTW can be found in references [29–31]. Genetic algorithms were proposed for the VRPTW by [32–37]. Other metaheuristics have been applied to the VRPTW that are based on simulated annealing or the ant colony approach. In recent years, hybrid algorithms have drawn much attention [38–41]; they combine the advantages of different algorithms and always yield better solutions than a single algorithm. However, very few papers emphasize the effect on the experimental results of architecture design throughout the computation process.

In this study, a P-based hybrid evolutionary algorithm (PHEA) is proposed to address the VRPTW. The structure of PHEA is based on a tissue-like membrane structure, and the genetic operators are used as the system's subalgorithms. The subalgorithms associated with membrane systems are inspired by biological phenomena. As we know, DNA sequences are composed of a special material in the nucleus. DNA carries almost all genetic information, and it has its own specific evolutionary concepts including recombination, mutation, and selection. We focus on the influence a membrane structure has on those genetic operations.

The PHEA can be considered as a bioinspired technique. It begins with a tissue membrane structure. After cell separation, genetic operators in the subsystems are used to make better solutions step by step. Some characters of membrane systems, such as cell separation and cell communication, can greatly improve the performance of genetic operators. Moreover, this paper includes a special improvement strategy to speed up the search process. Our algorithm is tested by Solomon's benchmark problems. Compared with the best-known solutions in the literature, our algorithm is competitive in terms of its solution quality.

2. Vehicle Routing Problem with Time Windows

The VRPTW can be described as a problem of designing an optimal set of routes that services all customers. Each customer is visited only once by exactly one vehicle within a given time interval; all routes start and end at the depot, and the total demands of all customers on one particular route must not exceed the capacity of the vehicle. Mathematically, it can be described as follows [33]. There are n customers who are serviced from a depot with vehicles of the same capacity, Q . For each customer i , where $i = 1, \dots, n$, there is a demand (q_i), a service time (s_i), and a service time window ($z_i = [e_i, f_i]$); the service begins at time b_i . The lower bound, e_i , describes the earliest time to start servicing and the upper bound, f_i , describes the latest time to start servicing. The demand q_i of customer i is to be covered by exactly one service within the time window of z_i . In addition, e_0 describes the earliest time a vehicle can depart from depot 0, and f_0 describes the latest time for the vehicle's arrival at the depot. The objective is to determine a feasible route schedule which primarily minimizes the number of vehicles and secondarily the total travel distance.

Consider the following:

$$\begin{aligned} \text{Min } f_1 &= \sum_{k=1}^n y_{0k}, \\ \text{Min } f_2 &= \sum_{i=0}^n \sum_{j=0}^n \sum_{k=1}^n C_{ij} x_{i,j,k}, \end{aligned} \quad (1)$$

subject to

$$\begin{aligned} \sum_{i=1}^n q_i y_{i,k} &\leq Q, \quad k = 1, 2, \dots, K, \\ \sum_{k=1}^n y_{0k} &= K, \\ \sum_{k=1}^n y_{1k} &= 1, \quad i = 1, 2, \dots, n, \end{aligned}$$

FIGURE 1: The flowchart of PHEA.

$$\begin{aligned}
 \sum_{i=1}^n x_{i0k} &= 1, \quad k = 1, 2, \dots, K, \\
 \sum_{i=0}^n x_{ijk} &= y_{jk}, \quad j = 1, 2, \dots, n; \quad k = 1, 2, \dots, K, \\
 \sum_{j=0}^n x_{ijk} &= y_{jk}, \quad i = 1, 2, \dots, n; \quad k = 1, 2, \dots, K, \\
 t_i &\geq e_i, \quad i = 1, 2, \dots, n, \\
 t_i + s_i &\leq f_i, \quad i = 1, 2, \dots, n,
 \end{aligned} \tag{2}$$

where C_{ij} represents the transportation cost from customer i to customer j and K is the number of vehicles.

3. A P-Based Hybrid Genetic Algorithm

In this paper, we propose a P-based hybrid evolutionary algorithm (PHEA). It is designed with the concept of tissue P system and genetic evolution. The flowchart of PHEA is shown in Figure 1.

The representation of a solution we use here is an integer string, first applied by Tan et al. [33]. The length is the number of customers in a route. Each point in the string is the integer node number originally assigned to that customer. The sequence of the points in the string is the order for visiting these customers.

The initial structure is $[]_1 []_2$. We call the membrane with label 2 the operation membrane and call the membrane with label 1 the output membrane. The local best solution in an operation membrane is denoted as l_{best} . The global best solution of the system is denoted as g_{best} . The operation membrane which obtains g_{best} is called the best operation membrane, while the operation membrane which obtains the worst l_{best} is called the worst operation membrane.

The procedure of the PHEA is described as follows.

Step 1. After the parameters have been set, such as the iteration number, the population size, the crossover rate, and the mutation rate, the PHEA randomly generates the initial chromosomes of the problem in membrane 2 using a random number generator.

Step 2. Separation rules $[a]_i \rightarrow [O_1]_i [O_2]_i$ are used to generate m cells labeled with the number 2. The structure variant is shown in Figure 2. The initial chromosomes are separated into m membranes, also with label 2. The number of chromosomes will differ by no more than one.

Step 3. In each operation membrane, a hybrid evolutionary algorithm is used to improve the initial routes.

Step 4. To avoid premature convergence and to add diversity to the new population, communication rules are used to exchange several chromosomes between the best operation membrane and the worst operation membrane for each x_{max} step; see Figure 3(a).

FIGURE 2: Separation process of the membrane with label 2.

FIGURE 3: Exchange solutions between membranes.

Step 5. The operation membrane sends its chromosomes into the output membrane for each $y_{\max}$ step, and then the output membrane reassigns the population to m operation membranes; see Figures 3(b) and 3(c). The number of chromosomes will differ by no more than one.

Step 6. If the iteration number reaches the maximum iteration number $\text{Iter}_{\max}$, the output membrane outputs g_{best} ; see Figure 3(d); otherwise, the chromosomes are sent to the operation membranes, and return to Step 3.

3.1. Hybrid Genetic Algorithm in Subsystems. In each operation membrane, hybrid evolutionary algorithm (HEA) combines several genetic operators with an improvement strategy. The flowchart is shown in Figure 4.

The procedure of HEA is described briefly as follows.

Step 1. Initialize the population.

Step 2. Evaluate individuals and select parents.

Step 3. Use the order crossover operator, the single parent crossover operator, and the mutation operator to generate children from selected parents.

Step 4. Use the improvement strategy to improve the individuals and update the population.

Step 5. If the iteration number of the local search in the operation membrane reaches $y_{\max}$ or the multiples of

FIGURE 4: The flowchart of HEA in the operation membranes.

TABLE 1: Parameters used for the PHEA.

Notation	Value	Parameter
N	40	The total population size
m	5	The number of subsystems
$x_{\max}$	20	Exchange chromosomes between operation membranes for each $x_{\max}$ step
$y_{\max}$	40	Send chromosomes to the output membrane for each $y_{\max}$ step
f	50%	The exchange rate
p_{c1}	10%	The order crossover rate
p_{c2}	10%	The single parent crossover rate
p_m	1	The mutation rate
L	4	The minimized length of the reserving route
$Iter_{\max}$	1000	The maximum iteration number

$y_{\max}$, the operation membrane sends its chromosomes into the output membrane. Otherwise, return to Step 6.

Step 6. If the iteration number of the local search in the operation membrane reaches $x_{\max}$ or the multiples of $x_{\max}$, exchange several chromosomes between the best operation membrane and the worst operation membrane. Otherwise, return to Step 2.

3.1.1. Selection and Evaluation. We implemented the tournament selection method to select parents and then generate offspring. A “tournament” is performed among a few individuals chosen at random from the population and the best parent is selected. The tournament size is an important parameter to adjust selection pressure. If the tournament size is larger, weak individuals have a smaller chance of being selected. However, a smaller size is a better choice to ensure diversity.

FIGURE 5: A route example made using the improvement strategy.

The evaluation of individuals is based on the objective to determine a feasible route schedule that primarily minimizes the number of vehicles and secondarily the total travel distance.

3.1.2. Recombination. Recombination operators are designed to generate children from selected parents. We used two types of reproduction operators in our algorithm. One of the reproduction operators used was a classical order crossover (OX), in which the relative order of the customers in the parent chromosomes is preserved in the child chromosome [43].

Example 1. Consider the following order crossover:

Parent 1:	2	4	1	3	6	5	7	8	9
Parent 2:	1	3	5	9	8	2	4	6	7
Offspring 1:	1	9	8	3	6	5	7	2	4
Offspring 2:	1	3	6	9	8	2	4	5	7

The other reproduction operator used was single parent crossover operator (SPO) [44]. This operator allows for more diversity. In it, each trip (a sequence of customers) in the parent solution has a 50% chance of being duplicated in the offspring solution. All other customers not being duplicated will be inserted into the sequence one by one, randomly. Among all of the possible positions to insert a customer, the customer is added to the position that results in the smallest increase in the total cost.

3.1.3. Mutation. We used three-point mutation in this paper to promote population diversity. The procedure of this operation is listed as follows.

Step 1. Pick three points at random from a sequence.

Step 2. Generate all of the possible permutations of the selected points while keeping the other points in the sequence unchanged. Then, five new sequences can be obtained, and they are regarded as the offspring.

Five chromosomes can be created from a parent's sequence by exchanging three genes, as shown in Example 2. Only the best transformation is used as the offspring produced by the mutation.

Example 2. Consider the following three-point mutation:

Parent:	2	4	1	3	6	5	7	8	9
Offspring 1:	2	4	1	3	8	5	7	6	9
Offspring 2:	2	6	1	3	4	5	7	8	9
Offspring 3:	2	6	1	3	8	5	7	4	9
Offspring 4:	2	8	1	3	6	5	7	4	9
Offspring 5:	2	8	1	3	4	5	7	6	9

3.1.4. Improvement Strategy. We propose a new improvement strategy to combine or even remove shorter routes before update of the population. The strategy is based on the INSERT Algorithm I_1 [19]. We first set a length L . All of the customers in the route whose length is less than L will be reinserted in other routes using the INSERT Algorithm I_1 . Thus, if the

TABLE 2: Comparison of PHEA with the best published results.

Problems	Best results		PHEA		GIDEON [29]		GenSAT [42]		RE (%)
	NV	TD	NV	TD	NV	TD	NV	TD	
R101	18	1644	19	1650.80	20	1700	18	1644	0.4
R102	17	1434.00	17	1486.12	17	1549	17	1493	3.6
R103	13	1207.00	13	1207.00	13	1319	13	1207	0
R104	10	982.01	10	982.01	10	1090	10	1048	0
R105	14	1377.11	14	1377.11	15	1448	14	1442	0
R106	12	1251.98	12	1251.98	13	1363	12	1350	0
R107	10	1104.66	11	1091.75	11	1187	11	1146	-1.2
R108	9	960.88	9	973.96	10	1048	10	898	1.4
R109	11	1194.73	11	1183.04	12	1345	12	1226	-1.0
R110	11	1080.36	11	1080.36	11	1234	11	1105	0
R111	10	1096.72	10	1096.72	11	1238	10	1151	0
R112	9	982.14	10	953.63	10	1082	10	992	-2.9
C101	10	827.30	11	891.63	10	833	10	829	7.8
C102	10	827.30	10	995.53	10	832	10	829	20.3
C103	10	828.06	9	987.04	10	873	10	835	19.6
C104	10	824.78	9	970.57	10	904	10	835	17.7
C105	10	828.94	10	905.52	10	874	10	829	9.6
C106	10	827.30	10	908.87	10	902	10	829	9.9
C107	10	827.30	10	968.06	10	926	10	829	17.0
C108	10	827.30	10	840.56	10	928	10	829	1.4
C109	10	828.94	10	828.94	10	957	10	829	0
RC101	14	1696.94	15	1650.29	15	1767	14	1669	-2.7
RC102	12	1554.75	12	1554.75	14	1569	13	1557	0
RC103	11	1110	11	1261.67	11	1328	11	1110	13.7
RC104	10	1135.48	10	1135.48	11	1263	10	1226	0
RC105	13	1629.44	14	1602.34	14	1612	14	1602	-1.7
RC106	11	1424.73	12	1384.92	12	1608	13	1420	-2.8
RC107	11	1230.48	11	1230.48	12	1396	11	1264	0
RC108	10	1139.82	10	1139.82	11	1250	10	1281	0
R201	4	1252.37	4	1255.76	4	1478	4	1354	0.3
R202	3	1191.70	4	1169.02	4	1279	4	1176	-1.9
R203	3	939.5	3	939.5	3	1167	3	1126	0
R204	2	825.52	3	858.49	3	909	3	803	4.0
R205	3	994.42	3	994.42	3	1274	3	1128	0
R206	3	838	3	906.14	3	1098	3	838	8.1
R207	2	890.61	2	890.61	3	1015	3	904	0
R208	2	726.82	2	726.82	3	826	2	823	0
R209	3	909.16	3	909.16	3	1159	2	855	0
R210	3	939.37	3	939.37	3	1269	3	1052	0
R211	2	885.71	2	923.80	3	898	3	816	4.3
C201	3	591.56	3	591.56	3	753	3	591	0
C202	3	591.56	3	591.56	3	756	3	707	0
C203	3	591.17	3	591.17	3	855	3	791	0
C204	3	590.60	3	590.60	3	803	3	685	0
C205	3	588.88	3	588.88	3	667	3	589	0
C206	3	588.49	3	588.49	3	694	3	588	0
C207	3	588.29	3	588.29	3	730	3	588	0
C208	3	588.32	3	588.32	3	735	3	588	0
RC201	4	1294	4	1392.38	4	1823	4	1294	7.6
RC202	3	1365.65	4	1365.65	4	1459	4	1291	0

TABLE 2: Continued.

Problems	Best results		PHEA		GIDEON [29]		GenSAT [42]		RE (%)
	NV	TD	NV	TD	NV	TD	NV	TD	
RC203	3	1049.62	3	1206.54	3	1323	3	1203	15.0
RC204	3	798.46	3	798.46	3	1021	3	897	0
RC205	4	1297.65	4	1297.65	4	1594	4	1389	0
RC206	3	1146.32	3	1146.32	3	1530	3	1213	0
RC207	3	1061.14	3	1061.14	3	1501	3	1181	0
RC208	3	828.14	3	848.10	3	1038	3	919	2.4

TABLE 3: Solutions of C103 and C104.

Number of route	Route
Name	C103
1	67-78-76-71-70-73-77-79-80-81-82-83
2	24-25-27-29-30-28-26-23-22-20-21-7
3	43-42-41-40-44-46-45-32-19-16-14-12
4	34-33-31-35-37-38-39-36-52-50-51-48
5	90-87-86-94-92-93-97-100-99-98-96-95
6	57-55-54-53-56-58-60-59-49-47
7	13-17-18-15-11-9-6-4-3-1-2
8	65-63-62-74-61-72-64-68-66-69
9	8-10-84-85-88-89-91-75-5
Name	C104
1	24-27-25-52-44-59-45-48-51-50-49-47
2	21-26-10-8-11-9-6-4-2-1-3-5
3	90-91-89-88-85-84-86-65-66-69-64-68
4	28-30-34-36-39-38-37-35-31-33-32-29
5	43-55-54-53-56-58-60-57-40-41-42-46
6	67-87-95-97-92-93-94-100-99-96-98-75
7	83-82-76-78-81-77-79-73-71-70-80
8	13-15-17-18-19-16-14-12-7
9	63-62-74-61-72-22-23-20

length on a route is no less than L , the route can be saved in the next step. Otherwise, the customers of the routes whose lengths are less than L need to choose a new suitable insertion place. We try to add those customers to other routes such that their length is no less than L . Equation (3) [19] is used to calculate the best feasible insertion place for customer u . Equation (5) [19] is used to choose the customer u^* who will be inserted in the next step.

Consider the following:

$$c_1(i(u), u, j(u)) = \min [c_1(i_{p-1}, u, i_p)], \quad p = 1, \dots, n, \quad (3)$$

where

$$c_1(i, u, j) = \alpha_1 c_{11}(i, u, j) + \alpha_2 c_{12}(i, u, j), \quad \alpha_1 \geq 0, \quad \alpha_2 \geq 0,$$

$$c_{11}(i, u, j) = d_{iu} + d_{uj} - \mu d_{ij}, \quad \mu \geq 0,$$

$$c_{12}(i, u, j) = b_{ju} - b_j. \quad (4)$$

The triple function (i_{p-1}, u, i_p) represents customer u being inserted between the points i_{p-1} and i_p , n represents the number of customers, d_{ij} represents the distance from customer i to customer j , b_j represents the time when service at customer j begins, and b_{ju} represents the start time for customer j after inserting customer u .

Consider the following:

$$c_2(i(u^*), u^*, j(u^*)) = \text{optimum} [c_2(i(u), u, j(u))], \quad (5)$$

where

$$c_2(i(u), u, j(u)) = \lambda d_{0u} - c_1(i(u), u, j(u)), \quad \lambda \geq 0. \quad (6)$$

When no more customers with feasible insertions can be found, the algorithm begins a new route (unless it has already routed all customers). This process is illustrated in Figure 5. In Figure 5(a), the round dots indicate customers, and the square represents the storage facility. In Figure 5(b), a tour is constructed by three routes. Supposing that $L = 4$, only the longest route in Figure 5(c) remains unchanged. The customers in the other routes, whose lengths are less than 4, are used to reconstruct new routes. Figure 5(d) shows the new routes that were constructed by the improvement strategy.

4. Computational Experiments

4.1. Benchmark Problems. Our experiment investigates the 56 benchmark problems described by Solomon [19]. Six groups of problems (R1, R2, C1, C2, RC1, and RC2) are used to verify the efficiency of the proposed PHEA. Problems in sets R1 and R2 have the customers' locations generated randomly over a square. Problems in sets C1 and C2 have the clustered customers whose time windows generated based on a known solution. Problems in sets RC1 and RC2 have a combination of randomly placed and clustered customers. Each benchmark problem contains 100 customers. Problems in sets R1, C1, and RC1 have narrow time windows and a small capacity of the vehicle, while problems in sets R2, C2, and RC2 have larger time windows and a larger capacity of the vehicle.

4.2. Parameter Settings. The proposed algorithmic framework was coded in C and executed on an AMD 3500+2.21 GHz with 1 GB of RAM under Windows XP. The parameters used in the proposed algorithm PHEA are shown in Table 1. For each benchmark problem, 10 independent runs were carried out.

TABLE 4: Comparison with the results of PHEA_{-cs,-symp}.

	Problems	BS	MBS	WS	STD	ET
PHEA _{-cs,-symp}	r107	1119.14	1126.37	1130.03	7.16	2207
	r109	1268.87	1280.36	1322.81	23.78	3105
	c108	840.56	893.1	929.7	32.5	8562
	c109	1064.63	1094.26	1131.01	28.98	7340
	rc101	1733.43	1746.11	1777.61	18.25	4985
	rc105	1647.46	1656.5	1674.92	11.21	6279
	R201	1434.71	1440.53	1457.71	10.81	6597
	R210	1203.25	1225.01	1296.43	39.99	4871
	C201	591.56	657.78	686.27	40.52	14673
	C202	643.23	659.67	668.22	10.45	10739
	RC201	1589.95	1603.71	1612.88	12.56	8214
	RC208	1049.6	1063.17	1088.99	14.11	8380
PHEA	r107	1091.75	1101.51	1110.8	7.96	3972
	r109	1183.04	1202.46	1218.88	14.42	9728
	c108	840.56	862.29	894.88	29.75	18370
	c109	828.94	865.16	896.98	37.27	15832
	rc101	1650.29	1654.72	1669	8.03	9856
	rc105	1602.34	1617.85	1629.44	11.6	11725
	R201	1255.76	1284.05	1328.42	38.76	12434
	R210	939.37	956.25	967.5	15.5	16287
	C201	591.56	591.56	591.56	0	26739
	C202	591.56	591.56	591.56	0	28473
	RC201	1392.38	1402.23	1427.08	15.25	16298
	RC208	848.10	867.04	942.78	12.76	18472

The number of runs is 10. BS, MBS, WS, STD, and ET represent the best solution, worst solution, standard deviation, and elapsed time (in seconds), respectively.

4.3. Results of PHEA. The comparison between the best-known solutions from the literature [29, 42, 45] and the solutions obtained by the PHEA is shown in Table 2. The solutions obtained by the PHEA are better than those in the previous literature for R107, R109, R112, RC101, RC105, RC106, and R202. The solution would be improved further if every parameter is set more reasonably. A comparison is also made between the optimal solutions obtained by the PHEA, a hybrid genetic algorithm [29], and an improved genetic algorithm [42]; see Table 2. The results show that our algorithm is competitive with other genetic algorithms.

The solutions for C103 and C104 comprise one vehicle less than the previous best solutions. Two complete solutions that include the sequence of customers are described in Table 3.

4.4. Comparison with the Results of PHEA_{-cs,-symp}. In this subsection, we briefly analyze the algorithm without the processes of cell separation and cell communication. We use the term PHEA_{-cs,-symp} to represent the method, which is based on PHEA but does not include membrane structure transformations. Two instances are selected from each type of benchmark problems. They are R107, R109, C109, C109, RC101, RC105, R201, R210, C201, C202, RC201, and RC208. The performances of the PHEA_{-cs,-symp} are evaluated by using the following criteria: the best solution and the worst solution over 10 runs, the mean best solution over 10 runs,

the standard deviation, and the elapsed time. Experimental results are shown in Table 4. We can see that the solutions are much worse than those solutions generated by PHEA, even though the improvement strategy and the genetic operators continue to be used.

5. Conclusions and Future Work

A membrane system is a good framework for approximate algorithms because of its hierarchical structure and intrinsic parallelism. As objects in membranes evolve according to specific rules, this process is similar to many types of heuristic algorithms. The rules in a membrane system can sometimes describe those algorithms. For example, if we suppose that chromosomes are represented by string objects in a membrane system, the mutation operator and the crossover operator can be described by rewriting rules.

Structure design is a very important research topic in the field of optimization algorithm. Our study focuses on the effects of algorithmic structure variation on the performance of genetic operators. In this work, the novel membrane algorithm PHEA is proposed. It contains a tissue-like membrane structure with separation rules and communication rules. Membrane structure variation is introduced in this algorithm. Several subsystems are initially produced via the separation rules. In each subsystem, chromosomes evolve

through genetic operators in a parallel way. Antiport rules are used between subsystems to exchange some number of individuals. This type of rule can increase the diversity of the population in each subsystem and improve the performance of genetic operators. To study VRPTW, we also designed a special improvement strategy to speed up the search process in the subsystems. The proposed algorithm was tested using Solomon's benchmark problems. The experimental results show that our algorithm is very competitive with the classical algorithms described in the literature.

Further research may include investigations into scalable methods for large VRPTW or other types of optimization problems. The vehicle routing problem considered in our paper has one depot and hard time windows. Other types of VRPTW, such as multi-depot vehicle routing problem with time windows [46, 47], vehicle routing problem with soft time windows [48, 49], and vehicle routing problem with split deliveries [50], can be tested by our algorithms with some changes. On the other hand, the influence of other characters of membrane system on the performance of approximate algorithms should be considered in the future, for example, choosing rules in a nondeterministic manner.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The authors are supported by the National Natural Science Foundation of China (61127005, 61272071, and 61373066).

References

- [1] G. Păun, "Computing with membranes," *Journal of Computer and System Sciences*, vol. 61, no. 1, pp. 108–143, 2000.
- [2] L. Pan and C. Martín-Vide, "Solving multidimensional 0-1 knapsack problem by P systems with input and active membranes," *Journal of Parallel and Distributed Computing*, vol. 65, no. 12, pp. 1578–1584, 2005.
- [3] L. Pan, D. P. Daniel, and M. J. Pérez-Jiménez, "Computation of Ramsey numbers by P systems with active membranes," *International Journal of Foundations of Computer Science*, vol. 22, no. 1, pp. 29–38, 2011.
- [4] C. Martín-Vide, G. Păun, J. Pazos, and A. Rodríguez-Patón, "Tissue P systems," *Theoretical Computer Science*, vol. 296, no. 2, pp. 295–326, 2003.
- [5] G. Păun, M. J. Pérez-Jiménez, and A. Riscos-Núñez, "Tissue P systems with cell division," *International Journal of Computers, Communications and Control*, vol. 3, no. 3, pp. 295–303, 2008.
- [6] L. Pan and G. Păun, "Spiking neural P systems: an improved normal form," *Theoretical Computer Science*, vol. 411, no. 6, pp. 906–918, 2010.
- [7] L. Pan, G. Păun, and M. J. Pérez-Jiménez, "Spiking neural P systems with neuron division and budding," *Science China Information Sciences*, vol. 54, no. 8, pp. 1596–1607, 2011.
- [8] T. Y. Nishida, "Membrane algorithm: an approximate algorithm for NP-complete optimization problems exploiting P-systems," in *Proceedings of the 6th International Workshop on Membrane Computing (WMC '05)*, pp. 26–43, Vienna, Austria, 2005.
- [9] A. Leporati and D. Pagani, "A membrane algorithm for the min storage problem," in *Membrane Computing: 7th International Workshop, WMC 2006, Leiden, The Netherlands, July 17–21, 2006, Revised, Selected, and Invited Papers*, vol. 4361 of *Lecture Notes in Computer Science*, pp. 443–462, Springer, Berlin, Germany, 2006.
- [10] D. Zaharie and G. Ciobanu, "Distributed evolutionary algorithms inspired by membranes in solving continuous optimization problems," in *Membrane Computing: 7th International Workshop, WMC 2006, Leiden, The Netherlands, July 17–21, 2006, Revised, Selected, and Invited Papers*, vol. 4361 of *Lecture Notes in Computer Science*, pp. 536–553, 2006.
- [11] Y. Zhang and L. Huang, "A variant of P systems for optimization," *Neurocomputing*, vol. 72, no. 4–6, pp. 1355–1360, 2009.
- [12] T. Y. Nishida, "Membrane algorithm with Brownian subalgorithm and genetic subalgorithm," *International Journal of Foundations of Computer Science*, vol. 18, no. 6, pp. 1353–1360, 2007.
- [13] L. Huang and I. H. Suh, "Controller design for a marine diesel engine using membrane computing," *International Journal of Innovative Computing, Information and Control*, vol. 5, no. 4, pp. 899–912, 2009.
- [14] J. Zhao and N. Wang, "A bio-inspired algorithm based on membrane computing and its application to gasoline blending scheduling," *Computers and Chemical Engineering*, vol. 35, no. 2, pp. 272–283, 2011.
- [15] L. Huang, I. H. Suh, and A. Abraham, "Dynamic multi-objective optimization based on membrane computing for control of time-varying unstable plants," *Information Sciences*, vol. 181, no. 11, pp. 2370–2391, 2011.
- [16] G. Zhang, M. Gheorghe, and C. Wu, "A quantum-inspired evolutionary algorithm based on P systems for knapsack problem," *Fundamenta Informaticae*, vol. 87, no. 1, pp. 93–116, 2008.
- [17] G. Zhang, C. Liu, and M. Gheorghe, "Diversity and convergence analysis of membrane algorithms," in *Proceedings of the 5th IEEE International Conference on Bio-Inspired Computing: Theories and Applications*, pp. 596–603, September 2010.
- [18] G. Zhang, J. Cheng, M. Gheorghe, and Q. Meng, "A hybrid approach based on differential evolution and tissue membrane systems for solving constrained manufacturing parameter optimization problems," *Applied Soft Computing Journal*, vol. 13, no. 3, pp. 1528–1542, 2013.
- [19] M. M. Solomon, "Algorithms for the vehicle routing and scheduling problems with time window constraints," *Operations Research*, vol. 35, no. 2, pp. 254–265, 1987.
- [20] B. Garcia, J. Potvin, and J. Rousseau, "A parallel implementation of the Tabu search heuristic for vehicle routing problems with time window constraints," *Computers and Operations Research*, vol. 21, no. 9, pp. 1025–1033, 1994.
- [21] W. Chiang and R. A. Russell, "A reactive tabu search meta-heuristic for the vehicle routing problem with time windows," *INFORMS Journal on Computing*, vol. 9, no. 4, pp. 417–430, 1997.
- [22] J. Wy, B. Kim, and S. Kim, "The rollon-rolloff waste collection vehicle routing problem with time windows," *European Journal of Operational Research*, vol. 224, no. 3, pp. 466–476, 2013.
- [23] N. Bhusiri, A. G. Qureshi, and E. Taniguchi, "The trade-off between fixed vehicle costs and time-dependent arrival penalties in a routing problem," *Transportation Research E: Logistics and Transportation Review*, vol. 62, pp. 1–22, 2014.

- [24] P. Amorim and B. Almada-Lobo, "The impact of food perishability issues in the vehicle routing problem," *Computers & Industrial Engineering*, vol. 67, pp. 223–233, 2014.
- [25] B. Melián-Batista, A. Santiago, F. Bello et al., "A bi-objective vehicle routing problem with time windows: a real case in Tenerife," *Applied Soft Computing*, vol. 17, pp. 140–152, 2014.
- [26] J.-F. Cordeau, G. Laporte, and A. Mercier, "A unified tabu search heuristic for vehicle routing problems with time windows," *Journal of the Operational Research Society*, vol. 52, no. 8, pp. 928–936, 2001.
- [27] C. A. Ullrich, "Integrated machine scheduling and vehicle routing with time windows," *European Journal of Operational Research*, vol. 227, no. 1, pp. 152–165, 2013.
- [28] H. C. Lau, M. Sim, and K. M. Teo, "Vehicle routing problem with time windows and a limited number of vehicles," *European Journal of Operational Research*, vol. 148, no. 3, pp. 559–569, 2003.
- [29] S. Thangiah, "Vehicle routing with time windows using genetic algorithms," in *Application Handbook of Genetic Algorithms: New Frontiers*, L. Chambers, Ed., pp. 253–277, CRC Press, Boca Raton, Fla, USA, 1995.
- [30] J. Homberger and H. Gehring, "Two evolutionary metaheuristics for the vehicle routing problem with time windows," *INFOR Journal*, vol. 37, pp. 297–318, 1999.
- [31] J. Homberger and H. Gehring, "A two-phase hybrid metaheuristic for the vehicle routing problem with time windows," *European Journal of Operational Research*, vol. 162, no. 1, pp. 220–238, 2005.
- [32] J. Berger, M. Barkaoui, and O. Bräysy, "A route-directed hybrid genetic approach for the vehicle routing problem with time windows," *INFOR Journal*, vol. 41, no. 2, pp. 179–194, 2003.
- [33] K. C. Tan, L. H. Lee, and K. Ou, "Artificial intelligence heuristics in solving vehicle routing problems with time window constraints," *Engineering Applications of Artificial Intelligence*, vol. 14, no. 6, pp. 825–837, 2001.
- [34] R. Bent and P. van Hentenryck, "A two-stage hybrid local search for the vehicle routing problem with time windows," *Transportation Science*, vol. 38, no. 4, pp. 515–530, 2004.
- [35] X. Zhang, S. Wang, Y. Niu, and L. Pan, "Tissue P systems with cell separation: attacking the partition problem," *Science China Information Sciences*, vol. 54, no. 2, pp. 293–304, 2011.
- [36] T. Vidal, T. G. Crainic, M. Gendreau, and C. Prins, "A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows," *Computers & Operations Research*, vol. 40, no. 1, pp. 475–489, 2013.
- [37] T. Chiang and W. Hsu, "A knowledge-based evolutionary algorithm for the multiobjective vehicle routing problem with time windows," *Computers & Operations Research*, vol. 45, pp. 25–37, 2014.
- [38] Y. Rochat and E. D. Taillard, "Probabilistic diversification and intensification in local search for vehicle routing," *Journal of Heuristics*, vol. 1, no. 1, pp. 147–167, 1995.
- [39] R. Baños, J. Ortega, C. Gil, A. L. Márquez, and F. de Toro, "A hybrid meta-heuristic for multi-objective vehicle routing problems with time windows," *Computers & Industrial Engineering*, vol. 65, no. 2, pp. 286–296, 2013.
- [40] R. Baños, J. Ortega, C. Gil, A. Fernández, and F. de Toro, "A simulated annealing-based parallel multi-objective approach to vehicle routing problems with time windows," *Expert Systems with Applications*, vol. 40, no. 5, pp. 1696–1707, 2013.
- [41] J. Michallet, C. Prins, L. Amodéo et al., "Multi-start iterated local search for the periodic vehicle routing problem with time windows and time spread constraints on services," *Computers & Operations Research*, vol. 41, pp. 196–207, 2014.
- [42] S. Thangiah, I. Osman, and T. Sun, "Hybrid genetic algorithm, simulated annealing and tabu search methods for vehicle routing problems with time windows," Technical Report UKC/OR94/4, Institute of Mathematics and Statistics, University of Kent, Canterbury, UK, 1994.
- [43] D. E. Goldberg, *Genetic Algorithms in Search Optimisation and Machine Learning*, Addison-Wesley, 1989.
- [44] S. Liu, W. Huang, and H. Ma, "An effective genetic algorithm for the fleet size and mix vehicle routing problems," *Transportation Research E: Logistics and Transportation Review*, vol. 45, no. 3, pp. 434–445, 2009.
- [45] <http://www.sintef.no/Projectweb/TOP/VRPTW/Solomon-benchmark/100-customers>.
- [46] R. Liu, Z. Jiang, and N. Geng, "A hybrid genetic algorithm for the multi-depot open vehicle routing problem," *OR Spectrum*, vol. 36, no. 2, pp. 401–421, 2014.
- [47] J. Luo and M. Chen, "Improved shuffled frog leaping algorithm and its multi-phase model for multi-depot vehicle routing problem," *Expert Systems with Applications*, vol. 41, pp. 2535–2545, 2014.
- [48] D. Taş, M. Gendreau, N. Dellaert, T. van Woensel, and A. G. de Kok, "Vehicle routing with soft time windows and stochastic travel times: a column generation and branch-and-price solution approach," *European Journal of Operational Research*, vol. 236, no. 3, pp. 789–799, 2014.
- [49] F. Guerriero, R. Surace, V. Loscrí, and E. Natalizio, "A multi-objective approach for unmanned aerial vehicle routing problem with soft time windows constraints," *Applied Mathematical Modelling*, vol. 38, no. 3, pp. 839–852, 2014.
- [50] P. Belfiore and H. T. Y. Yoshizaki, "Heuristic methods for the fleet size and mix vehicle routing problem with time windows and split deliveries," *Computers & Industrial Engineering*, vol. 64, no. 2, pp. 589–601, 2013.