

Hybrid Genetic Algorithm in Solving Vehicle Routing Problems with Time Window Constraints

Dr. Loo Hay Lee, Dr. Kay Chen Tan & Mr. Ke Ou
National University of Singapore

Abstract

This paper describes the research on developing a hybrid Genetic Algorithm (HGA) in solving Vehicle Routing Problems with Time Window Constraints (VRPTW). HGA uses intuitive integer string representation with group information attached. It also incorporates other techniques such as group information updating and local search improvement. The implemented heuristic was applied to solve Solomon's 56 VRPTW 100-customer instances, and yield 15 solutions better than or equivalent to the best solutions ever published in literature.

An integer string of length N is used as the representation of chromosome, where N is the number of customers. Each gene in the chromosome is the customer node number. The sequence of the genes is the order of visiting these customers. The following is one chromosome:

$$2\ 5\ 8\ 10\ 12\ 9\ 3\ 7\ 6\ 1\ 4\ 11 \quad (1)$$

For the same chromosome string, it may represent many feasible solutions as long as all the capacity and time window constraints are not violated, such as:

$$(2\ 5\ 8\ 10)\ (12\ 9\ 3\ 7)\ (6\ 1\ 4\ 11)\ \text{and}\ (2\ 5\ 8)\ (10\ 12\ 9\ 3\ 7\ 6)\ (1\ 4\ 11) \quad (2)$$

where each portion in the parentheses represents one route. The only difference between these representations is the *grouping* of customer nodes. With different grouping one chromosome can represent different solutions. In the hybrid GA, we simply associate the grouping information with each chromosome, then the chromosome representation is:

$$2\ 5\ 8\ 10\ 12\ 9\ 3\ 7\ 6\ 1\ 4\ 11\ [3\ 6\ 3] \quad (3)$$

In each of the new representation, the second part inside parentheses is the grouping information. For example, $[3\ 6\ 3]$ indicates the first 3 customer nodes is in the first route, the next 6 customer nodes is in the second route, and the last 3 customer nodes belong to the last route. The fitness value of the representation can be calculated easily according to this grouping information.

To obtain a better grouping for each chromosome, a local search for grouping is conducted by changing the grouping information until one feasible better grouping is found. For example, if the original grouping is $[6\ 5\ 1]$, we can check the new grouping combinations such as $[5\ 6\ 1]$, $[5\ 5\ 2]$ or $[4\ 5\ 3]$. The searching order and operators can be defined easily. Each of the new groupings will be first checked for feasibility, only feasible groupings will be under consideration. The search will stop when a new better grouping is found or the searching time is out.

After the grouping update process, chromosomes will go through the Local Search Improvement (LSI) process for further fine-tuning. The LSI process is defined as follows: a portion of the chromosomes are decoded into their route configurations, and then they will undergo λ -interchange local search procedure to improve solutions. The improved solutions are then encoded back to chromosome format. This process is called *Local Search Improvement (LSI)*. Only some randomly selected chromosomes will undergo this LSI process.

Every iteration before going through the normal selection and reproduction process, the chromosome will first go through the improvement procedures which includes both the grouping update process and LSI process. Notice that the grouping information will not be used in the normal GA process.

The population size of each generation is kept at 1000 in our problem. All the initial chromosomes are generated randomly. The total generation number is set at 500 to 1000, which is a

compromise between computational time and final results. The process of HGA is shown below, where P1 and P2 are two preset numbers:

Heuristic	Hybrid Genetic Algorithm (hGA)
HybridGA-1:	Initialize chromosomes and set the iteration counter $N=0$.
HybridGA-2:	Get initial grouping information for each chromosome.
HybridGA-3:	Set the regrouping counter M .
HybridGA-4:	Update chromosome grouping information using local search.
HybridGA-5:	Undergo LSI Process, $M = M+1$.
HybridGA-6:	If ($M > P1$)
	Go to HybridGA-7.
	Else
	Go to HybridGA-4.
HybridGA-7:	GA operations: selection, crossover and mutation. $N = N+1$.
HybridGA-8:	If ($N > P2$)
	Go to HybridGA-9.
	Else
	Go to HybridGA-2.
HybridGA-9:	Terminate.

This HGA was applied to six VRPTW data sets, i.e., R1, C1, RC1, R2, C2 and RC2 which was generated by Solomon consisting of 100 customers with Euclidean distance.

Table 1 compares the average distance obtained from nine different heuristics and the hybrid genetic algorithm. The best known results ever published in literature are also listed in the table.

Table 1: Comparison among different heuristics, hGA and best known results

Prob Class	Rochat and Taillard 1995	Russell 1995	Chiang and Russell 1996	Potvin et al. 1996	Potvin and Bengio 1996	Taillard et al. 1996	Bachem et al. 1997	Chiang and Russell 1997	Homberger and Gehring 1999	hGA	Best Known Results
C1	828.45	930.00	909.80	861.00	838.00	828.45	829.50	828.38	828.38	851.05	827.5
C2	590.32	681.00	684.10	602.50	589.90	590.30	591.88	591.42	589.86	620.12	589.9
R1	1197.42	1317.00	1308.82	1294.70	1296.83	1216.70	1264.24	1204.19	1228.06	1220	1174.7
R2	954.36	1167.00	1166.42	1185.90	1117.70	995.38	1100.33	986.32	969.95	985.69	942.0
RC1	1369.48	1523.00	1473.90	1465.00	1446.20	1367.51	1414.63	1397.44	1392.57	1366.62	1354.1
RC2	1139.79	1398.00	1401.50	1476.10	1360.60	1165.62	1341.35	1229.54	1144.43	1108.5	1081.3
All	57120	65827	65201	64679	62572	57993	61523	58502	57876	57903	56108

Notice that the best-known results are collectively gathered from different papers in literature covering the most recent research on the optimization of VRPTW in using Solomon's data sets. We obtained totally 15 better/equivalent solutions (26.8% out of 56 instances).

In conclusion, the HGA that we proposed in this paper not only uses a new chromosome representation scheme for the VRPWT, but also incorporate two improvement methods. From the result analysis, it can be seen that our heuristic is one of the best heuristics in solving VRPTW problems.