2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

A Genetic Algorithm for the Minimum Latency Pickup and Delivery Problem

Xin-Lan Liao, Chih-Hung Chien, and Chuan-Kang Ting
Department of Computer Science and Information Engineering and
Advanced Institute of Manufacturing with High-tech Innovations
National Chung Cheng University
Chia-Yi 621, Taiwan
Email: {lhl99p, cchung100m, ckting}@cs.ccu.edu.tw

***Abstract*—The pickup and delivery problem combines vehicle routing and objects distribution to cope with logistic problems. While most research on PDP aims to minimize the transportation cost for the sake of service providers, this study proposes the minimum latency pickup and delivery problem (MLPDP) that seeks a low-latency route to transport commodities among nodes, where latency represents the sum of transportation time between demanders and the corresponding suppliers. The MLPDP is pertinent to time-sensitive services and logistics focusing on customer satisfaction. This study defines the latency of a customer as the average time elapsed aboard of goods received. The last-in-first-out loading method is employed to simulate real-world rear-loaded vehicles. This study further designs a genetic algorithm (GA) to resolve the MLPDP. In particular, we propose the edge aggregate crossover (EAC) and the reversely weighting technique to improve the performance of GA on the MLPDP. Experimental results show the effectiveness of the proposed GA. The results further indicate that EAC leads to significantly better performance than conventional crossover operators in solution quality and convergence speed on the MLPDP.**

I. Introduction

The pickup and delivery problem (PDP) combines vehicle routing and objects distribution. The goal of PDP is to find the optimal visiting schedules for vehicles to convey passengers or commodities from pickup nodes to delivery nodes. According to the properties of transportation endpoints, PDPs can be categorized into three structures: one-to-one, one-to-many-to-one, and many-to-many [1], [2], [3]. This study focuses on many-to-many PDP, where a delivery customer can have any source of supply considering the applications like farm fresh distribution that collects produce from farms and warehouses to serve grocers, restaurants, and institutions. While most research on PDP aims to minimize the transportation cost for the sake of service providers, we consider customer satisfaction and propose the minimum latency pickup and delivery problem (MLPDP). Specifically, latency represents the sum of transportation time between demanders and the corresponding suppliers; for the example of farm fresh distribution that conveys perishable freight, minimizing the latency can reduce the deterioration in nutrition so as to provide reliable produce. The MLPDP seeks a low-latency route to gather commodities from pickup nodes and distribute them to delivery nodes in the meanwhile. In addition, the vehicle load is bounded by its capacity and must be non-negative along the visiting tour to avoid overload and insufficient supply, respectively. The objective of the MLPDP is to optimize the visiting tour in terms of transportation time as well as customer satisfaction, and therefore meets the crucial requirements of time-sensitive services.

Latency is a common measurement of system performance in practice, which gauges the response time to a request or the delay experienced in a system. The requirement for minimizing latency arises in real-world applications such as multimedia processing and packet-switched network, in that the instruction set of processors as well as networking devices and transmission media cause delay and encumber overall efficiency in some cases. In logistics industry, service provider associates latency with the waiting time before the customer's demand is satisfied. Example scenarios include door-to-door freight services and humanitarian aid after natural disaster. This problem has been formulated as the cumulative capacitated vehicle routing problem (CCVRP) to minimize the total arrival time to reach customers through designated routes subject to capacity limitation [4], [5], [6], [7]. The CCVRP is closely related to the minimum latency problem (MLP) [8], [9], [10], which discards vehicle capacity and is applicable in managing the schedules for repairmen or house cleaning services. The Dial-a-Ride problem, moreover, specifies pairs of points to transport passengers from particular places to respective destinations and thus generalizes the routing problems. In such a demand-responsive transportation system, the objective takes waiting time and riding time of customers into account for their satisfaction [11], [12], [13], [14]. Instead of reducing the transportation cost that benefits the service provider, latency minimization strikes a balance between customer satisfaction and prime cost, leading to good brand reputation and attracting more potential customers. The above customer-oriented formulations in operations research therefore reconfirm the rationality of the minimum latency pickup and delivery problem.

To resolve the MLPDP, this study designs a genetic algorithm (GA) with novel edge aggregate crossover operator. GA is a nature-inspired global search approach and has succeeded in solving a variety of optimization problems. It utilizes a set of chromosomes representing candidate solutions and exchanges or infuses slight fluctuation into genetic information through crossover and mutation. The fitness evaluation determines the improvement direction for the search and assists selection of parents and survivors for advancing solution quality. The alternate exploration and exploitation in the search space introduced by variation operators and selection pressure respectively imitates evolution and leads the population toward the global optimum after generations. Instead of commonly used crossover operators for permutation representation, e.g., order crossover, cycle crossover, and partially mapped crossover, this study devises the *edge aggregate crossover* to construct offspring based on the information acquired from fitness evaluation. Furthermore, we design the *reversely weighting technique* to evaluate the fitness value with time complexity $\mathcal{O}(n^2)$. This evaluation along with the violation measure specifies fitter individuals regarding the feasibility and constitutes the constraint handling method that directs the search to low-latency feasible solution. Experiments compare the performance of edge aggregate crossover with other permutation-based crossover operators. The results validate the efficacy of the proposed operators in terms of both solution quality and convergence speed.

The organization of this study is as follows. Section II presents the formulation of MLPDP. The reversely weighting technique and edge aggregate crossover are proposed in Section III; Section IV examines their effectiveness through a series of experiments. Finally, we draw conclusions and recommend the directions for future work in Section V.

II. Problem Formulation

The MLPDP aims to schedule a visiting route with lowest total latency to transport commodities from pickup nodes to delivery nodes under the constraint on vehicle load. Given a complete graph $G = (V, A)$ with $V = \{v_0, ..., v_n\}$ and $A = \{(v_i, v_j)|v_i, v_j \in V, v_i \neq v_j\}$, in which each vertex $v_i \in V$ is associated with a demand d_i, and each arc $(v_i, v_j) \in A$ has a non-negative cost $c_{i,j} > 0$, representing the transportation time among nodes. The vertex set V is the union of the starter v_0 with $d_0 = 0$ and two disjoint sets, namely $V^+ = \{v_i|v_i \in V, d_i > 0\}$ of pickup nodes and $V^- = \{v_i|v_i \in V, d_i < 0\}$ of delivery nodes. The objective of the MLPDP is to find a feasible permutation $\pi = (v_0, v_{(1)}, ..., v_{(n)})$, where $v_{(i)}$ denotes the ith visited customer, such that the total latency experienced by all delivery nodes is minimum. Formally,

$$\min \sum_{v_i \in V^-} \omega_i \cdot \mathrm{latency}(\pi, v_i), \tag{1}$$

where ω_i specifies the importance of a customer. This study considers all delivery nodes equally important and defines the latency as the weighted sum of transportation time within a demander and its corresponding suppliers; to be specific, each supplier contributes to a portion of demands of a delivery node, and the latency of particular freight is the product of time elapsed aboard and the ratio of its amount to the total commodities requested. In other words, let $x_{i,j}$ be the decision variable with

$$x_{i,j} = \begin{cases} 1 & \text{vehicle travels from } v_i \text{ to } v_j \\ 0 & \text{otherwise} \end{cases}$$

The objective function (1) is defined by

$$\min \sum_{v_i \in V^-} \sum_{v_j \in V^+} \frac{\sigma_{j,i}}{|d_i|} \cdot t_{j \to i} \tag{2}$$

s.t.

$$\sum_{v_i \in V} x_{i,j} = 1, \quad \forall v_j \in V \tag{3}$$

$$\sum_{v_i, v_j \in S} x_{i,j} < |S|, \quad \forall S \subset V \tag{4}$$

$$0 \leq \lambda_{(i)} \leq Q, \quad \forall i \in \{0, \dots, n\} \tag{5}$$

$$x_{i,j} \in \{0, 1\} \tag{6}$$

where pickup node v_j supplies delivery node v_i with commodities amounted to $\sigma_{j,i}$, which is a non-negative integer and keeps conservation of quantity: $\sum_{v_j \in V^+} \sigma_{j,i} = |d_i|$ for each $v_i \in V^-$ and $\sum_{v_i \in V^-} \sigma_{j,i} = d_j$ for each $v_j \in V^+$; $t_{j \to i}$ denotes the transportation time from pickup node v_j to delivery node v_i via the resultant route, i.e., $t_{i \to i} = 0$ and $t_{j \to i} = \min_{0 \leq k \leq n} \frac{c_{j,k}}{x_{j,k}} + t_{k \to i}$, recursively taking steps toward particular demander. The objective function (2) presents the latency regarding the composition of freight, where normalization indicates the average transportation time consumed to satisfy customer requests. Constraints (3) and (4) guarantee to visit each node exactly once and eliminate subtours, respectively. The vehicle load limits are presented in (5), where $\lambda_{(i)}$ indicates the vehicle load at $v_{(i)}$ along the visiting order, i.e., $\lambda_{(i)} = \lambda_{(i-1)} + d_{(i)}$ with $\lambda_{(0)} = 0$. Notably, $\sigma_{j,i}$ not only records the amount of commodities supplied to v_i but indicates their source v_j. In this study, the carrier discharges goods in a last-in-first-out (LIFO) manner to simulate rear-loaded vehicles in practice. Figure 1 illustrates an example route $\pi = (v_0, v_4, v_5, v_3, v_1, v_2)$ for the MLPDP. Given the visiting order and LIFO loading, v_3, v_5 and v_4 contribute $\sigma_{3,1} = 8$, $\sigma_{5,1} = 1$, and $\sigma_{4,1} = 1$ to v_1, respectively. As a result, v_1 encounters latency amounted to $\frac{8}{10} \cdot 4 + \frac{1}{10} \cdot (4+3) + \frac{1}{10} \cdot (4+3+6)$ in accordance with the objective function. On the other hand, v_4 satisfies the demand of v_2, taking $\frac{5}{5} \cdot (9+4+3+6)$ of latency; the total latency of the route is therefore 27.2 from all delivery nodes. Note that formula (1) is a general-purpose objective function for the MLPDP; other practical definitions of latency function and viable loading methods are also applicable.

III. Methodology

This study designs special genetic operators for GA to resolve the MLPDP. GA encodes candidate solutions into chromosomes, forming a population and enabling the search for global optima in the solution space. To simulate the evolution in Nature, the algorithm iteratively improves candidate solutions following the principle of "Survival of the Fittest". In GA, crossover and mutation are variation operators infusing

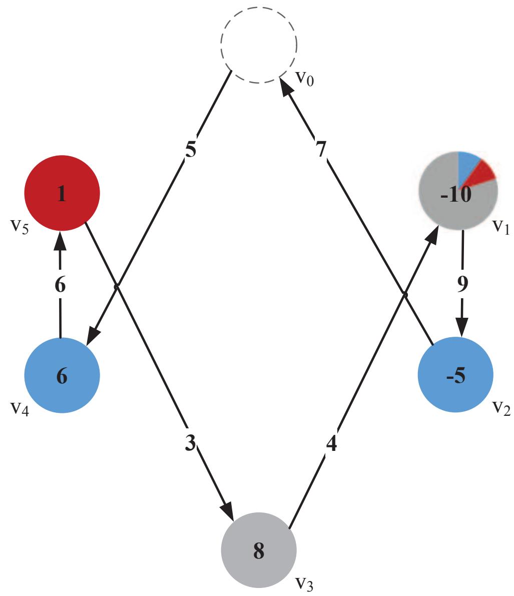


Figure 1. Example route for the MLPDP. The figures inside circles and the numbers on arcs denote d_i and transportation time between two vertices, respectively. The commodities held by each pickup node are denoted with distinct color, and the colors in a delivery node indicate the sources of the goods obtained as well as respective proportions to the request.

diversity into the population, where a pair of chromosomes is chosen for crossover operator to recombine the parental information, and the mutation then slightly changes the offspring in order to escape from local optima. The fitness function evaluates solution quality and assists selection operator in distinguishing fitter individuals for the given problem. GA follows the procedure of parent selection, crossover, and mutation; afterward, the offspring with (or without) the parent population compete to survive into next generation according to the fitness value. The evolutionary process repeats till the termination criterion is satisfied.

To explore the solution space in the course of evolution, the proposed GA represents visiting tours by order-based chromosome. The order-based representation reflects permutation of nodes, which guarantees to visit each node exactly once and forms the candidate solutions to the MLPDP. Instead of summing up the latency experienced by each delivery node, we propose a reversely weighting technique, which derives the weights of the traveled edges so that the total latency is decomposed and, particularly, the feature of supply along the route emerges. The edge aggregate crossover is accordingly introduced to inherit useful parental information for the MLPDP, taking advantage of the characteristics obtained from fitness evaluation. The following subsections describe the proposed GA in detail.

A. Fitness Evaluation and Constraint Handling

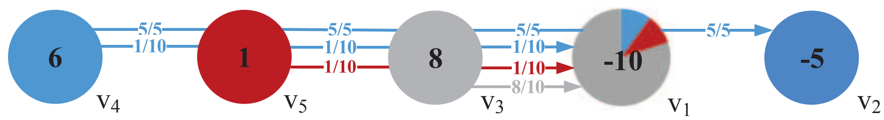
Fitness evaluation determines the quality of candidate solutions and directs the evolutionary process toward the optima. This study adopts the objective function (2) that totals the latency of each request from delivery node; namely,

Algorithm 1 Reversely weighting technique.

function $f(\pi)$
  $\delta \leftarrow 0$
  stack $\leftarrow \emptyset$
  feasible $\leftarrow$ true
  $i \leftarrow n$
  loop
    if feasible **then**
      if $d_{(i)} < 0$ **then**
        push $(v_{(i)}, d_{(i)})$ to stack
        $\delta \leftarrow \delta + 1$
      else
        $s \leftarrow d_{(i)}$
        loop
          if stack.size() $\neq 0$ **then**
            pop (v_j, d') from stack
            $\delta \leftarrow \delta - \frac{\min\{s, |d'|\}}{|d_j|}$
            $s \leftarrow s + d'$
          else
            $s \leftarrow 0$
            feasible $\leftarrow$ false
            $\delta \leftarrow 1$
          end if
        until $s \leq 0$
        if $s < 0$ **then**
          push (v_j, s) to stack
        end if
      end if
    end if
    $\ell_{(i-1,i)} \leftarrow \delta \cdot c_{(i-1,i)}$
    $i \leftarrow i - 1$
  until $i = 0$
return $\sum_{i=1}^{n} \ell_{(i-1,i)}$

$$f(\pi) = \sum_{v_i \in V^-} \sum_{v_j \in V^+} \frac{\sigma_{j,i}}{|d_i|} \cdot t_{j \to i}.$$

The double summation engages $\mathcal{O}(n^2)$ of time complexity on fitness evaluation. Restated, as tracing the visiting order, vehicle arrival time of all nodes are recorded, and each pickup node along with the number of commodities it holds is kept in a stack for subsequent implementation of LIFO loading method. In the meantime, delivery nodes pop the stock in the stack to obtain goods, determining the proportion of supplies to the corresponding requests. Furthermore, transportation time is calculated by subtracting vehicle arrival time of the supplier from that of the delivery customer receiving its freight. For example, route $\pi = (v_0, v_4, v_5, v_3, v_1, v_2)$ in Fig. 1 results in a sequence of arrival time as $(0, 5, 11, 14, 18, 27)$. The vehicle discharges freight collected from v_3, v_5 and v_4; hence v_1 experiences



(a) Supply flow of π.

i	$v_{(i)}$	$v_{(i-1)}$	δ	$c_{(i-1,i)}$	$\ell_{(i-1,i)}$	stack
5	v_2	v_1	1.0	9	9.0	$(\mathbf{v_2}, \mathbf{-5})$
4	v_1	v_3	2.0	4	8.0	$(v_2, -5), (\mathbf{v_1}, \mathbf{-10})$
3	v_3	v_5	1.2	3	3.6	$(v_2, -5), (\mathbf{v_1}, \mathbf{-2})$
2	v_5	v_4	1.1	6	6.6	$(v_2, -5), (\mathbf{v_1}, \mathbf{-1})$
1	v_4	v_0	0.0	5	0.0	$\emptyset$
				$\sum_{i=1}^{n} \ell_{(i-1,i)}$	27.2	

(b) Procedure of reversely weighting technique. Total latency of π: 27.2.

Figure 2. Fitness evaluation of example route $\pi = (v_0, v_4, v_5, v_3, v_1, v_2)$ with reversely weighting technique (cf. Fig. 1): (a) Supply flow of π. The flows with distinct colors specify their respective sources of commodities, and the fractions decompose the weights on traveled edges, where the numerator and denominator indicate the supplies from pickup node ($\sigma_{j,i}$) and the demand of the delivery customer receiving the particular freight ($|d_i|$), respectively. (b) Procedure of reversely weighting technique. The data with boldface denotes the latest record in stack.

latency amounted to $\frac{8}{10} \cdot (18-14) + \frac{1}{10} \cdot (18-11) + \frac{1}{10} \cdot (18-5)$. As providing v_2 with the remaining commodities in the stack, the total latency of the tour increases by $\frac{5}{5} \cdot (27-5)$. Note that the number of nodes and the stack size determine the time complexity of this evaluation process, which is $\mathcal{O}(n^2)$ in worst case.

However, direct fitness evaluation dilutes the composition of total latency since the delay on each visited edge is obscure. This study therefore proposes the reversely weighting technique to analyze the weight on edge with the same time complexity. In addition, the proposed evaluation process can simultaneously examine vehicle load along the route to assist in handling constraint violation. The reversely weighting technique not only offers genetic information but facilitates the violation measure, which are adopted by variation and selection operators in GA.

1) Reversely Weighting Technique: The reversely weighting technique accumulates the proportion of supplies and weights each edge in reverse order. In the MLPDP, the commodities required assemble before visiting a delivery node to satisfy the request. This property implies that one weight is appended to the edges on the visiting path between the specific delivery node and its latest supplier according to the objective function. This study therefore traces the route reversely to perceive node type and weight each visited edge based on the aforementioned observation. Specifically, the reversely weighting technique adopts a variable δ, which is initialized to zero and increases by one when stepping on delivery node. In addition, each individual request is stacked before fully satisfied by preceding pickup nodes, and δ decreases as reaching pickup node to reflect the contribution of the supply to delivery customers. Notably, the value of δ at that time indicates the weight of passing edge. Algorithm 1 presents the procedure of reversely weighting technique, where the double loops enable reverse visits and stack manipulation, taking $\mathcal{O}(n^2)$ of time complexity in worst case.

To further utilize the weights on edges, the proposed reversely weighting technique maintains an edge table for each chromosome in the course of fitness evaluation. Figure 2 takes the aforementioned route $\pi = (v_0, v_4, v_5, v_3, v_1, v_2)$ for example and illustrates the evaluation procedure as well as edge table generation. In Fig. 2a, we decompose the weights on the traveled edges for investigation into the variation of δ. Specifically, δ increases by one in the first two steps, i.e., $i = 5$ and 4, to reflect the assemblage of supply flows before delivery nodes v_2 and v_1; then $\frac{8}{10}$ and $\frac{1}{10}$ are subtracted from δ according to Algorithm 1 when i shifts to 3 and 2, respectively, trimming the flows after passing through the corresponding sources v_3 and v_5 as shown in Fig. 2a. Lastly, v_4 serves the remaining requests of both delivery customers, leaving edge (v_0, v_4) zero weight. In addition, each node along with its predecessor and the corresponding share of the total latency ($\ell_{(i-1,i)}$) is kept in the edge table. Note that the pseudo-code of reversely weighting technique deals with exceptions triggered by infeasible solutions so as to generate edge table for future reference.

2) Constraint Handling Method: The MLPDP seeks a route with lowest total latency to transport commodities from pickup nodes to delivery nodes subject to vehicle capacity and non-negative load. In the course of search process, the operators in GA may produce infeasible solutions that exceed capacity limitation or hold insufficient freight for some delivery requests during the visits. The reversely weighting technique examines vehicle load while evaluating visiting tour in reverse order. Figure 3 illustrates the vehicle load variation of example route $\pi = (v_0, v_4, v_5, v_3, v_1, v_2)$. This example reveals the fact that the variation measured in reverse order mirrors the load variation of actual visiting direction. According to this observation, the proposed fitness evaluation determines feasibility through negative $\lambda_{(i)}$, which coincides with the real

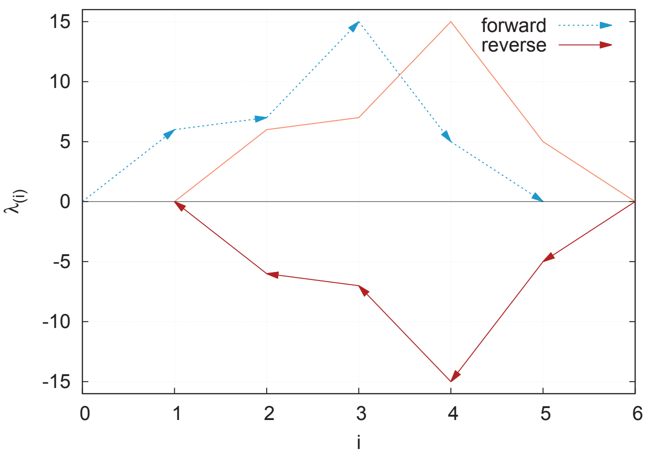


Figure 3. Vehicle load variation of example route $\pi = (v_0, v_4, v_5, v_3, v_1, v_2)$. The arrows indicate evaluation direction; the dotted and solid vectors represent vehicle load as visiting particular $v_{(i)}$ in forward and reverse order, respectively. Additionally, the solid line plots the negative quantity of the load variation measured in reverse order. Note that in this example, $v_{(6)}$ corresponds to depot v_0.

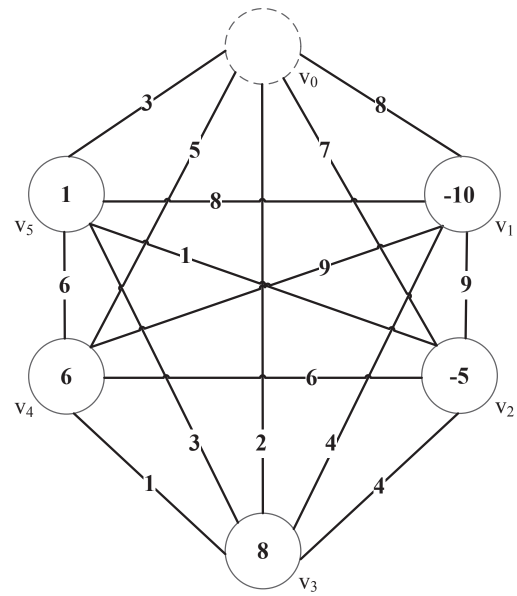


Figure 4. An example MLPDP. The figures inside circles and the numbers on arcs denote d_i and transportation time between two vertices, respectively.

load variation if shifted to one step forward.

Furthermore, this study adopts the constraint handling method proposed by Deb [15] to guide the search process toward low-latency feasible solution. The selection operators in GA consider feasible solutions better than infeasible ones; moreover, the comparison among feasible individuals is based on fitness function $f(\pi)$, while the quality of infeasible solutions is determined by violation measure $g(\pi)$. In this study, we employ the violation measure designed by Ting and Liao [16]. Formally,

$$g(\pi) = \lambda_{\text{exc}} + |\lambda_{\text{neg}}|$$

with

$$\lambda_{\text{exc}} = \max_{i\epsilon\{1,\dots,n\}}(\lambda_{(i)}, Q) - Q$$
$$\lambda_{\text{neg}} = \min_{i\epsilon\{1,\dots,n\}}(\lambda_{(i)}, 0)$$

where λ_{exc} indicates the maximal amount of exceeding load and λ_{neg} denotes the extreme shortage of commodities on board. Note that both $f(\pi)$ and $g(\pi)$ are to be minimized during the evolution process. Through this constraint handling method, the selection operators enable low-latency feasible solutions to have higher probability to generate offspring and survive until subsequent generations, leading to the optimal visiting tour.

B. Edge Aggregate Crossover

The conventional order-based crossover operators such as order crossover, cycle crossover, and partially mapped crossover guarantee to visit each node exactly once [17]. In addition to permutation attribute preservation, this study attempts to inherit useful genetic information from parents to improve the performance of GA for the MLPDP. The proposed crossover is called edge aggregate crossover (EAC). The EAC takes advantage of the edge table generated by reversely weighting technique to construct offspring. Since the edge table records the composition of total latency experienced through the duration of the visits, the EAC can choose edges for offspring according to probable effects occurred in parents. Specifically, the proposed EAC begins with a randomly selected delivery node and determines the route in reverse order owing to the direction of fitness evaluation. To retain parental features, the edge of either parent with lower $\ell_{(i-1,i)}$ is involved. Nevertheless, both choices may destroy permutation attribute; the EAC then follows the visiting order of a certain parent with probability of 0.5 or greedily chooses the edge consuming least transportation time among those connecting to unvisited nodes. Figure 5a shows a pair of parents and their respective edge tables on the MLPDP instance in Fig. 4. Notably, the chromosome representation excludes depot v_0 to reduce the solution space. Figure 5b describes the procedure of the proposed EAC, where we randomly select a node from one of the parents since edges (v_3, v_1) and (v_5, v_1) have the same share in total latency of p_1 and p_2, respectively. Moreover, in steps 3 and 5, the EAC attaches the worse candidate due to feasibility of representation. The EAC introduces new edges into the population in some cases and enables offspring to possess features of both parents, e.g., $c_1 = (v_0, v_4, v_2, v_5, v_3, v_1)$.

IV. Experimental Results

This study conducts a series of experiments to examine the effectiveness of the proposed edge aggregate crossover (EAC) in comparison to conventional order-based recombinations, i.e., order crossover (OX), cycle crossover (CX) and partially mapped crossover (PMX). Table I summarizes the parameter setting of GA in the experiments. The empirical study uses the benchmark introduced in [16] and conducts experiments on Intel i7-920 machines. Each experiment includes 30 independent runs due to the stochastic nature of GA. All problem instances

p_1 | v_4 | v_5 | v_3 | v_1 | v_2 |

$v_{(i)}$	$v_{(i-1)}$	$\ell_{(i-1,i)}$
v_1	v_3	8.0
v_2	v_1	9.0
v_3	v_5	3.6
v_4	v_0	0.0
v_5	v_4	6.6

p_2 | v_3 | v_4 | v_2 | v_5 | v_1 |

$v_{(i)}$	$v_{(i-1)}$	$\ell_{(i-1,i)}$
v_1	v_5	8.0
v_2	v_4	11.4
v_3	v_0	0.0
v_4	v_3	0.8
v_5	v_2	0.9

(a) Selected parents and their respective edge tables.

step	candidate_1	candidate_2	action	current state
1	V^-		random	(v_1)
2	v_3 (8.0)	v_5 (8.0)	random	(v_3, v_1)
3	v_5 (3.6)	v_0 (0.0)	candidate_1	(v_5, v_3, v_1)
4	v_4 (6.6)	v_2 (0.9)	candidate_2	(v_2, v_5, v_3, v_1)
5	v_1 (9.0)	v_4 (11.4)	candidate_2	c_1 | v_4 | v_2 | v_5 | v_3 | v_1

(b) Procedure of edge aggregate crossover.

Figure 5. Example for edge aggregate crossover: (a) Selected parents p_1 and p_2 with respective edge tables. (b) Procedure of edge aggregate crossover. Columns candidate_1 and candidate_2 are predecessors of the latest attached node in p_1 and p_2, respectively; furthermore, the number inside parentheses presents the $\ell_{(i-1,i)}$ obtained from the corresponding parent.

Table I
PARAMETER SETTING OF GA.

Parameter	Value
Representation	Order-based
Initialization	Random
Population size	500
Parent selection	Binary tournament
Crossover	EAC, OX, CX, and PMX ($p_c = 1.0$)
Mutation	Swap ($p_m = 1.0$)
Survival selection	$(\mu + \lambda)$
Termination	20000 generations

are set to $\sum_{i=0}^{n} d_i = 0$ for the equilibrium of total supply and total demand. The name of instance consists of the original benchmark name and the vehicle capacity, e.g., n60mosBq146 indicating instance n60mosB with vehicle capacity 146.

First, we investigate the average total latency obtained from GA with different crossover operators. According to Table II, the proposed EAC achieves lowest latency on all tested instances. The EAC overwhelmingly outperforms CX with statistical significance. The insignificant difference in solution quality between EAC and OX as well as PMX on n20mosAq44 and n20mosBq40 could be caused by small search space. In general, the superiority of the proposed EAC over conventional recombinations increases as the number of nodes grows. Figures 6a and 6b illustrate the anytime behavior of GA using EAC and OX on n40mosBq85 and n60mosBq146, respectively. The OX is compared in that it obtains lower latency on most problem instances among the three conventional order-based recombinations. The results reflect that OX suffers from premature convergence and is trapped into local optima; by contrast, the proposed EAC keeps search potential and reaches quality visiting routes.

To examine the scalability of the EAC on resolving the MLPDP, this study further employs test bench containing more nodes and compares the performance of EAC with that of representative recombination, i.e., OX. The termination criterion of GA is extended to a million generations due to the increase of problem scale. In Table III, the proposed EAC significantly excels OX in solution quality on all instances. Additionally, OX still encounters premature convergence in Fig. 6c and 6d, yielding more delay during transportation. The smooth convergence of GA with EAC in Fig. 6d reveals that the best chromosomes generated by OX and EAC may reach similar solution quality in early stage of evolution on some problem instances. The proposed EAC, however, significantly surpasses OX in subsequent generations. These outcomes verify the effectiveness of extracting and inheriting genetic information in the proposed EAC for the MLPDP.

Table III
AVERAGE TOTAL LATENCY OF ROUTE OVER 30 TRIALS (AVE.) AND SAMPLE STANDARD DEVIATION (STD.) OBTAINED FROM GA WITH EDGE AGGREGATE CROSSOVER (EAC) AND ORDER CROSSOVER (OX). THE p-VALUE SHOWS THE RESULTS OF ONE-TAILED t-TEST BETWEEN EAC AND OX. BOLDFACE INDICATES THE LOWEST AVERAGE TOTAL LATENCY AND THE p-VALUE LESS THAN SIGNIFICANCE LEVEL ($\alpha = 0.05$).

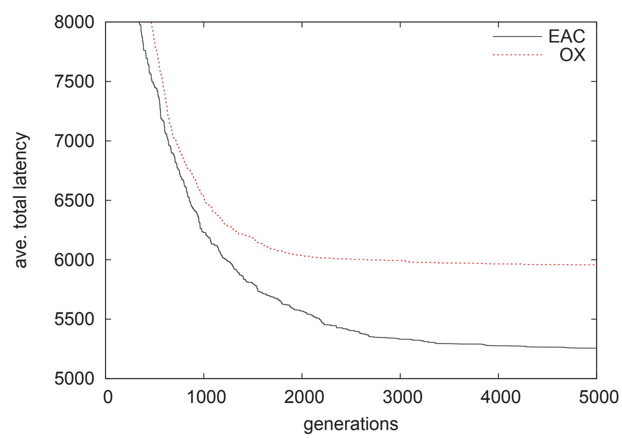
Instance	EAC		OX		
	ave.	std.	ave.	std.	p-value
n100mosAq218	**9365.10**	3.48E+02	12299.10	7.85E+02	**1.11E-21**
n100mosBq243	**9630.72**	5.65E+02	12401.25	9.37E+02	**1.22E-18**
n200mosAq460	**14882.01**	8.99E+02	19059.39	1.09E+03	**3.81E-23**
n200mosBq499	**17650.52**	1.27E+03	21875.05	1.24E+03	**3.58E-19**
n300mosAq721	**24321.75**	2.17E+03	27539.76	1.19E+03	**3.44E-09**
n300mosBq743	**25198.44**	1.99E+03	28250.29	1.59E+03	**9.84E-09**

V. CONCLUSIONS

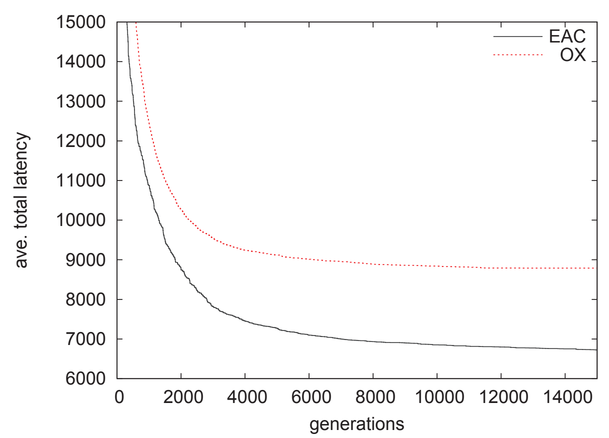
This study proposes the minimum latency pickup and delivery problem (MLPDP), which seeks a low-latency route to transport commodities from pickup nodes to delivery nodes. The MLPDP is applicable to time-sensitive services such as conveyance of perishable goods in farm fresh distribution. Moreover, it considers the quality of a visiting tour from

Table II
AVERAGE TOTAL LATENCY OF ROUTE OVER 30 TRIALS (AVE.) AND SAMPLE STANDARD DEVIATION (STD.) OBTAINED FROM GA WITH DIFFERENT RECOMBINATION OPERATORS, I.E., EDGE AGGREGATE CROSSOVER (EAC), ORDER CROSSOVER (OX), CYCLE CROSSOVER (CX), AND PARTIALLY MAPPED CROSSOVER (PMX). THE p-VALUE SHOWS THE RESULTS OF ONE-TAILED t-TEST BETWEEN EAC AND THE PARTICULAR CROSSOVER. BOLDFACE INDICATES THE LOWEST AVERAGE TOTAL LATENCY AMONG THE TESTED OPERATORS AND THE p-VALUE LESS THAN SIGNIFICANCE LEVEL ($\alpha = 0.05$).

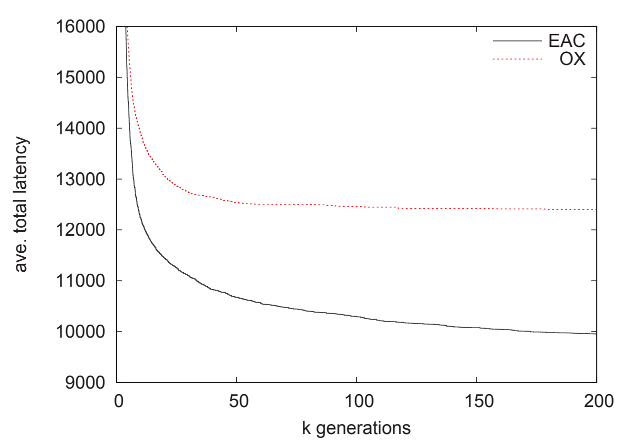
Instance	EAC		OX			CX			PMX		
	ave.	std.	ave.	std.	p-value	ave.	std.	p-value	ave.	std.	p-value
n20mosAq44	**3313.21**	9.87E+01	3307.36	9.39E+01	4.08E-01	3376.90	1.65E+02	**3.78E-02**	3355.81	1.63E+02	1.13E-01
n20mosBq40	**2855.61**	0.00E+00	2868.55	2.19E+01	1.63E-01	2927.43	2.09E+02	**3.49E-02**	2897.21	1.85E+02	1.14E-01
n30mosAq70	**5771.77**	2.18E+02	6006.62	2.83E+02	**2.93E-04**	6444.23	2.83E+02	**7.28E-11**	6317.04	3.40E+02	**8.01E-10**
n30mosBq62	**4385.07**	6.57E+01	4671.12	1.87E+02	**1.22E-09**	5004.64	3.05E+02	**1.61E-12**	4836.24	2.04E+02	**9.23E-14**
n40mosAq93	**4738.03**	1.47E+02	5264.73	3.30E+02	**3.73E-10**	5687.88	4.96E+02	**5.03E-12**	5641.98	4.40E+02	**6.55E-13**
n40mosBq85	**5209.32**	3.32E+02	5931.35	4.48E+02	**1.62E-09**	6490.59	5.17E+02	**9.02E-16**	6408.78	5.25E+02	**1.48E-14**
n50mosAq99	**3915.71**	9.98E+01	4614.86	4.03E+02	**3.19E-11**	5159.13	4.53E+02	**5.26E-16**	4941.35	5.15E+02	**2.90E-12**
n50mosBq122	**6712.44**	2.18E+02	8160.30	7.34E+02	**2.01E-12**	8663.25	6.69E+02	**2.66E-17**	8159.81	6.33E+02	**3.08E-14**
n60mosAq126	**5838.74**	3.01E+02	7235.31	6.35E+02	**2.84E-15**	8293.36	7.05E+02	**1.80E-20**	7676.47	6.47E+02	**1.26E-17**
n60mosBq146	**6662.92**	3.25E+02	8732.93	7.07E+02	**5.42E-18**	9203.83	7.44E+02	**2.96E-20**	8582.71	8.81E+02	**1.07E-13**



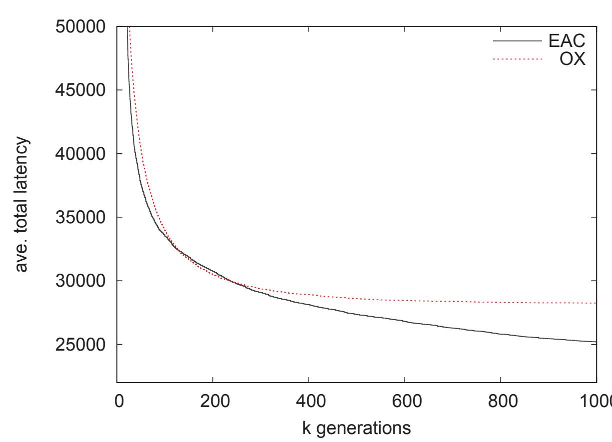
(a) n40mosBq85.



(b) n60mosBq146.



(c) n100mosBq243.



(d) n300mosBq743.

Figure 6. Anytime behavior of EAC and OX.

the viewpoint of customer satisfaction rather than service provider; that is, the objective is to minimize the total latency encountered during freight transportation between demanders and the corresponding suppliers. A general-purpose problem formulation is presented for any specified latency function and loading method. This study takes the composition of the freight into account and considers the average time elapsed aboard of goods received as the latency of a delivery node. In addition, LIFO loading method is employed to simulate rear-loaded vehicle in practice. The proposed fitness evaluation adopts the reversely weighting technique, which decomposes the objective into the weighted edge cost rather than the latency experienced by each delivery node and achieves time complexity $\mathcal{O}(n^2)$. To tackle the MLPDP, we design the edge aggregate crossover (EAC), which constructs offspring according to genetic information on parental routes acquired from fitness evaluation.

A series of experiments is conducted to verify the effectiveness of the proposed EAC on resolving the MLPDP. Statistical test results reveal that EAC leads to significantly lower total latency than conventional order-based crossover, i.e., order crossover, cycle crossover, and partially mapped crossover. Moreover, the anytime behavior suggests that the EAC infuses diversity into population while retaining useful features of parents and thus enables escape from local optima. These preferable outcomes validate the effectiveness and advantages of the proposed EAC and reversely weighting technique.

Future work for the MLPDP includes several directions, for example, amelioration of the proposed EAC, heuristic enhancement on latency reduction, development of local search, and attempt at other constraint handling methods.

Acknowledgment

This work was supported by the National Science Council of Taiwan under contract NSC100-2628-E-194-009-MY3, and Ministry of Education, Taiwan.

References

[1] G. Berbeglia, J.-F. Cordeau, I. Gribkovskaia, and G. Laporte, "Static pickup and delivery problems: A classification scheme and survey," *TOP*, vol. 15, no. 1, pp. 1–31, 2007.

[2] S. N. Parragh, K. F. Doerner, and R. F. Hartl, "A survey on pickup and delivery problems—Part I: Transportation between customers and depot," *Journal für Betriebswirtschaft*, vol. 58, pp. 21–51, 2008.

[3] ——, "A survey on pickup and delivery problems—Part II: Transportation between pickup and delivery locations," *Journal für Betriebswirtschaft*, vol. 58, pp. 81–117, 2008.

[4] S. U. Ngueveu, C. Prins, and R. W. Calvo, "An effective memetic algorithm for the cumulative capacitated vehicle routing problem," *Computers & Operations Research*, vol. 37, no. 11, pp. 1877–1885, 2010.

[5] P. Chen, X. Dong, and Y. Niu, "An iterated local search algorithm for the cumulative capacitated vehicle routing problem," in *Technology for Education and Learning*, ser. Advances in Intelligent Systems and Computing, H. Tan, Ed. Springer Berlin Heidelberg, 2012, vol. 136, pp. 575–581.

[6] G. M. Ribeiro and G. Laporte, "An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem," *Computers & Operations Research*, vol. 39, pp. 728–735, 2012.

[7] L. Ke and Z. Feng, "A two-phase metaheuristic for the cumulative capacitated vehicle routing problem," *Computers & Operations Research*, vol. 40, no. 2, pp. 633–638, 2013.

[8] A. Blum, P. Chalasani, D. Coppersmith, B. Pulleyblank, P. Raghavan, and M. Sudan, "The minimum latency problem," in *Proceedings of the Twenty-sixth Annual ACM Symposium on Theory of Computing (STOC1994)*, 1994, pp. 163–171.

[9] K. Chaudhuri, B. Godfrey, S. Rao, and K. Talwar, "Paths, trees, and minimum latency tours," in *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science (FOCS2003)*, 2003, pp. 36–45.

[10] M. M. Silva, A. Subramanian, T. Vidal, and L. S. Ochi, "A simple and effective metaheuristic for the minimum latency problem," *European Journal of Operational Research*, vol. 221, no. 3, pp. 513–520, 2012.

[11] H. N. Psaraftis, "A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem," *Transportation Science*, vol. 14, no. 2, pp. 130–154, 1980.

[12] J.-F. Cordeau and G. Laporte, "The dial-a-ride problem: models and algorithms," *Annals of Operations Research*, vol. 153, no. 1, pp. 29–46, 2007.

[13] S. N. Parragh, K. F. Doerner, R. F. Hartl, and X. Gandibleux, "A heuristic two-phase solution approach for the multi-objective dial-a-ride problem," *Networks*, vol. 54, no. 4, pp. 227–242, 2009.

[14] G. Mauri, L. Antonio, and N. Lorena, "Customers' satisfaction in a dial-a-ride problem," *IEEE Intelligent Transportation Systems Magazine*, vol. 1, pp. 6–14, 2009.

[15] K. Deb, "An efficient constraint handling method for genetic algorithms," *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 2–4, pp. 311–338, 2000.

[16] C. K. Ting and X. L. Liao, "The selective pickup and delivery problem: Formulation and a memetic algorithm," *International Journal of Production Economics*, vol. 141, no. 1, pp. 199–211, 2013.

[17] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*, ser. Natural Computing Series. Springer Berlin Heidelberg, 2003.