

Pre- and End-Haulage of Intermodal Container Terminals Modelled as a Full Truckload Pickup and Delivery Problem with Time Windows

An Caris

Gerrit K. Janssens

Transportation Research Institute

Hasselt University - campus Diepenbeek

Wetenschapspark 5 - bus 6, 3590 Diepenbeek, Belgium

e-mail: {an.caris,gerrit.janssens}@uhasselt.be

KEYWORDS

Intermodal transport, vehicle routing, time windows, full truckload, heuristics.

ABSTRACT

Road transport makes up a large share of total costs of intermodal transport. In this paper pre- and end-haulage in the service area of intermodal container terminals is modelled as a Full Truckload Pickup and Delivery Problem. A single container is delivered from the terminal to a customer or picked up at a customer location and returned to the terminal. Customers impose hard time windows. A two-phase insertion heuristic is developed. In a first phase delivery customers are paired with pickup customers. These pairs of customers are assigned to routes in a second phase. The initial solution is improved by a local search procedure based on a CROSS operator. The construction heuristic and improvement heuristic are demonstrated by means of a numerical example.

I. INTRODUCTION

Pre- and end-haulage of intermodal container terminals involves the pickup or delivery of containers at customer locations. Road transport constitutes a relatively large share of intermodal transport costs. Therefore, the attractiveness of intermodal transport can be increased by organizing the road segment in the intermodal transport chain more efficiently.

The drayage of containers in the service area of an intermodal terminal can be modelled as a Full Truckload Pickup and Delivery Problem with Time Windows (FTPDPPTW). In this study a full truckload is assumed to be a single container. A delivery activity to a consignee starts from the intermodal terminal with a full container and a pickup activity returns a container to the intermodal terminal for shipment by barge.

II. RELATED LITERATURE

The Pickup and Delivery Problem (PDP) is an extension to the classical Vehicle Routing Problem (VRP) where customers may both receive and send goods. Dumas et al. [3] formulate the pickup and delivery prob-

lem with time windows and present an exact solution based on column generation. Savelsbergh and Sol [9] give a general description of the PDP. Less than truckload PDP are among others investigated by Nanry and Barnes [8], Landrieu et al. [6] and Lu and Dessouky [7]. Gronalt et al. [4] study the problem of transporting full truckloads between distribution centres. In their PDPTW goods are transported between customer locations, as opposed to our problem definition where all containers either originate or return to the terminal. A full truckload PDPTW is also considered by Currie and Salhi [1] and [2]. The problem studied in these papers differs from our setting with respect to the definition of tasks. Goods have to be picked up at works of a construction company and delivered to customers. The closest related article to our research is given by Imai et al. [5]. The authors present a heuristic based on Lagrangian relaxation for the drayage problem of intermodal container terminals, without taking time windows into account.

III. PROBLEM DEFINITION

The following optimization problem can be formulated in terms of a vehicle routing problem with full container load. Assuming a homogeneous container type and size, find the optimal assignment of delivery and pickup customer pairs to a fleet of vehicles, in order to minimize the total cost of servicing all customers, which includes fixed vehicle costs and travelling costs. In accordance with Dumas et al. [3], a fixed vehicle cost is introduced to minimize the fleet size. Each vehicle used incurs a fixed cost, which may vary with the vehicle. Fixed costs include depreciation of own vehicles or leasing costs if the vehicle is hired, insurance payments and fixed costs for hiring an extra truck driver. Travelling costs are proportional to the total time necessary to service all customers, which implies travelling time and truck waiting time at customer sites.

All orders are assumed to be known in advance, so the problem is studied in a static environment. An intermodal terminal is open during a pre-specified daily time window. All trucks k must return to the terminal before the end of their depot window L_k . A final as-

assumption underlying our problem definition is the existence of hard time windows at customer locations. This leads to the following integer programming formulation:

$$\text{Min} \sum_{i \in (V^D \cup 0)} \sum_{j \in (V^P \cup 0)} \sum_{k \in K} CR_{ijk} x_{ijk} + \sum_{k \in K} FC_k$$

subject to

$$\sum_{i \in (V^D \cup 0)} \sum_{k \in K} x_{ijk} = 1 \quad \forall j \in V^P \quad (1)$$

$$\sum_{j \in (V^P \cup 0)} \sum_{k \in K} x_{ijk} = 1 \quad \forall i \in V^D \quad (2)$$

$$E_i \leq b_i \leq L_i \quad \forall i \in V^D \quad (3)$$

$$E_j \leq b_j \leq L_j \quad \forall j \in V^P \quad (4)$$

$$x_{ijk} \in \{0, 1\} \quad \forall i \in (V^D \cup 0), j \in (V^P \cup 0), k \in K \quad (5)$$

V^D = set of delivery points (index i)

V^P = set of pickup points (index j)

K = set of trucks (index k)

CR_{ijk} = cost of performing route (i, j) by truck k

FC_k = fixed cost of truck k

$x_{ijk} = 1$ if delivery i and pickup j are served in one route by truck k

L_k = depot window of truck k

E_i = earliest start time of delivery i

L_i = latest start time of delivery i

E_j = earliest start time of pickup j

L_j = latest start time of pickup j

b_i = actual time delivery i begins

b_j = actual time pickup j begins

t_{0i} = travel time from terminal 0 to delivery i

t_{ij} = travel time from delivery i to pickup j

t_{j0} = travel time from pickup j to terminal 0

s_i = service time of delivery i

s_j = service time of pickup j

Pickup and delivery customers can be combined in one route or can be serviced separately. In the latter case either the delivery point i or the pickup point j is represented by the depot 0 in our formulation. Equations (1) and (2) guarantee that all pickups and deliveries are visited only once. Constraints (3) and (4) state the hard time windows of customers. Constraint (5) defines the problem as a binary integer program. The presence of hard time windows implies:

$$\sum_{k \in K} x_{ijk} = 1 \quad \Rightarrow \quad b_i + s_i + t_{ij} \leq b_j \quad \forall i \in V^D, \forall j \in V^P \quad (6)$$

Constraint (6) is linearized to solve the problem to optimality:

$$b_i + s_i + t_{ij} - (L_i + s_i + t_{ij} - E_j) \cdot (1 - \sum_{k \in K} x_{ijk}) \leq b_j \quad \forall i \in V^D, \forall j \in V^P \quad (7)$$

When delivery point i and pickup point j are serviced in a single route, expression (7) equals (6). When delivery point i and pickup point j are not visited in a single route, expression (7) reduces to:

$$b_j - E_j \geq b_i - L_i$$

This inequality is always satisfied since the left-hand side is positive and the right-hand side is negative.

Hard time windows are also imposed on the total service time of a route k . These depot time windows are expressed by the following constraints:

$$\sum_{i \in (V^D \cup 0)} x_{ijk} = 1 \quad \Rightarrow \quad b_j + s_j + t_{j0} \leq L_k \quad \forall j \in V^P, \forall k \in K \quad (8)$$

$$x_{i0k} = 1 \quad \Rightarrow \quad b_i + s_i + t_{i0} \leq L_k \quad \forall i \in V^D, \forall k \in K \quad (9)$$

Constraints (8) and (9) can be linearized in a similar way:

$$b_j + s_j + t_{j0} - (1 - \sum_{i \in (V^D \cup 0)} x_{ijk}) \cdot (L_j + s_j + t_{j0}) \leq L_k \quad \forall j \in V^P, \forall k \in K \quad (10)$$

$$b_i + s_i + t_{i0} - (1 - x_{i0k}) \cdot (L_i + s_i + t_{i0}) \leq L_k \quad \forall i \in V^D, \forall k \in K \quad (11)$$

A final set of constraints ensures feasibility of the routes. Two additional variables are defined:

p_{ik} = position of delivery customer i in route k

p_{jk} = position of pickup customer j in route k

The position is equal to zero if a customer is not inserted in the route. Otherwise the position takes on a positive integer value.

$$\sum_{i \in (V^D \cup 0)} x_{ijk} = 0 \quad \Rightarrow \quad p_{jk} = 0 \quad \forall j \in V^P, \forall k \in K \quad (12)$$

$$\sum_{j \in (V^P \cup 0)} x_{ijk} = 0 \quad \Rightarrow \quad p_{ik} = 0 \quad \forall i \in V^D, \forall k \in K \quad (13)$$

Each pickup customer in a pair is serviced after the delivery customer.

$$x_{ijk} = 1 \quad \Rightarrow \quad p_{jk} = p_{ik} + 1 \quad \forall i \in V^D, \forall j \in V^P, \forall k \in K \quad (14)$$

Service of a delivery customer cannot begin before the pickup customer in a preceding pair is serviced by the same truck.

$$p_{ik} > p_{jk} \quad \Rightarrow \quad b_i - t_{0i} \geq b_j + s_j + t_{j0} \quad \forall i \in V^D, \forall j \in V^P, \forall k \in K \quad (15)$$

The next section proposes a two-phase insertion heuristic to find near-optimal solutions. An improvement heuristic is presented in section V. Section VI demonstrates both heuristics by means of a numerical example. Finally, conclusions are drawn in section VII.

IV. TWO-PHASE INSERTION HEURISTIC

The VRP belongs to the class of NP-hard problems. Problem sizes which exact models are able to solve are relatively small. In practice heuristics are used to solve problems of realistic size. In this section an insertion heuristic is developed, which consists of two phases to obtain a near-optimal solution. In a first phase pickups and deliveries are combined into pairs. These pairs of customers are inserted into routes in a second phase.

A. Phase 1: Pairing pickups and deliveries

Due to the existence of hard time windows, not every pickup customer and delivery customer can be combined into a feasible route. Inconsistencies in time windows are checked first:

$$E_i + s_i + t_{ij} \leq L_j \quad \forall i \in V^D, j \in V^P$$

This check results in a list of feasible combinations. The waiting time between delivery i and pickup j can be limited to a maximum amount $MAXWAIT$. A feasible pair of customers is discarded from the list if the minimum waiting time $MINWAIT_{ij}$ is larger than allowed.

$$MINWAIT_{ij} = \begin{cases} 0 & \text{if } E_j \leq L_i + s_i + t_{ij} \\ E_j - (L_i + s_i + t_{ij}) & \text{else} \end{cases}$$

A pickup customer j and delivery customer i are allowed in one tour only if:

$$MINWAIT_{ij} \leq MAXWAIT$$

Second, interesting combinations of customers are selected. In forming pairs of pickups and deliveries, both spatial and temporal aspects are to be taken into account. The savings in travel time obtained from serving delivery i and pickup j together should be as large as possible. The following expression for savings in travel time should be maximized:

$$(t_{i0} + t_{0j} - t_{ij})$$

The time window slack between customers i and j should be as small as possible, which implies a minimization of:

$$(L_j - E_i - s_i - t_{ij})$$

Therefore, the pair of pickup and delivery customers with the lowest value for the following criterion is selected first:

$$w_1 \cdot (L_j - E_i - s_i - t_{ij}) + w_2 \cdot (t_{ij} - t_{i0} - t_{0j}) \quad (16)$$

The weights w_1 and w_2 reflect the importance given to each objective. These weights are parameters of the

construction heuristic. Depending on the nature of the problem, more weight should be given to the savings in waiting time or the savings in travel time. The process of pairing customers is repeated until no further feasible combinations exist between the remaining pickup customers and delivery customers. The remaining customers are inserted into individual routes and form an imaginary pair with a dummy customer.

An opportunity cost for not choosing the best combination for a delivery i or pickup j can also be taken into account. Gronalt et al. [4] argue that this regret approach leads to significant improvements in the best solution. The opportunity cost $OC1_i$ (respectively $OC1_j$) can be defined as the difference in savings in travel time achieved by the best combination for delivery i (pickup j) and the currently selected combination. Selection criterion (16) can be extended with this third objective.

$$w_1 \cdot (L_j - E_i - s_i - t_{ij}) + w_2 \cdot (t_{ij} - t_{i0} - t_{0j}) + w_3 \cdot (OC1_i + OC1_j) \quad (17)$$

The opportunity cost related to the time window slack can also be added to this selection criterion. This opportunity cost $OC2_i$ (respectively $OC2_j$) is defined as the difference between the time window slack of the current combination and the smallest time window slack of delivery i (pickup j) in any combination.

$$w_1 \cdot (L_j - E_i - s_i - t_{ij}) + w_2 \cdot (t_{ij} - t_{i0} - t_{0j}) + w_3 \cdot (OC1_i + OC1_j) + w_4 \cdot (OC2_i + OC2_j) \quad (18)$$

B. Phase 2: Route construction

In a second phase routes are constructed consecutively. The first route is serviced by the vehicle with the lowest fixed cost FC_k . Vehicles are used in increasing order of their fixed costs. Pairs of customers are inserted into routes in increasing order of their latest start time L_{ij} .

$$L_{ij} = \text{Min}\{L_i - t_{0i}; L_j - t_{ij} - s_i - t_{0i}\}$$

A pair of customers can be inserted in an existing route k if it can start later than the time necessary to service the customers already assigned to the vehicle k and if the vehicle is able to return to the terminal within its depot window. The total time necessary to service customers assigned to a vehicle k is defined as the route service time RS_k .

$$RS_k \leq L_{ij} \text{ and } \text{Max}(RS_k, E_{ij}) + RS_{ij} \leq L_k$$

The route service time RS_k is initially set to 0. The time necessary to service pair (i, j) , RS_{ij} , is the sum of travel times, service times and the minimum waiting time.

$$RS_{ij} = s_i + s_j + t_{0i} + t_{ij} + t_{j0} + MINWAIT_{ij}$$

In case insertion into multiple existing routes is feasible, the pair of customers is added to the existing route with

the smallest waiting time between the previous pair. If no feasible insertions in existing routes are possible, the pair of customers is assigned to an unused vehicle to create a new route.

Finally the route service time RS_k is updated. Define the earliest starting time E_{ij} as the earliest time a vehicle can leave the depot for servicing pair (i, j) without unnecessary waiting between delivery i and pickup j .

$$E_{ij} = \begin{cases} L_i - t_{0i} & \text{if } L_i \leq (E_j - t_{ij} - s_i) \\ E_j - t_{ij} - s_i - t_{0i} & \text{if } E_i \leq (E_j - t_{ij} - s_i) \leq L_i \\ E_i - t_{0i} & \text{if } (E_j - t_{ij} - s_i) \leq E_i \end{cases}$$

This leads to the following expression for updating the route service time RS_k after inserting pair (i, j) .

$$RS_k = \begin{cases} RS_k + RS_{ij} & \text{if } E_{ij} < RS_k \\ E_{ij} + RS_{ij} & \text{else} \end{cases}$$

V. IMPROVEMENT HEURISTIC

In this section a local search heuristic is proposed to improve a feasible solution found by the construction heuristic described above. Considering the nature of the problem, the neighborhood of the local search procedure is defined as follows. Two pairs of pickup and delivery customers, for example (g, h) and (i, j) , are selected from two different routes. These pairs are recombined into new pairs of pickup and delivery customers, (g, j) and (i, h) . This move is further denoted as CROSS operator. The local search heuristic first lists all feasible CROSS moves. A CROSS move is feasible if the pickup customers and delivery customers can be combined into new pairs, taking into account their time windows. Second, it is checked whether these new pairs of customers can be reinserted into the routes. Either (g, j) is inserted into the first route and (i, h) into the second or the other way round. Next, the local search heuristic selects the CROSS move with the largest improvement I_{ghij} or smallest deterioration in route service times RS_k .

$$I_{ghij} = RS_{gh} + RS_{ij} - RS_{gj} - RS_{ih}$$

The improvement heuristic stops after a predefined number of iterations without any reduction in total costs of servicing all customers.

VI. NUMERICAL EXAMPLE

A numerical example is discussed to demonstrate the mechanism of the construction heuristic and improvement heuristic. In this example an intermodal terminal has to deliver containers to five customer sites and pick up containers at five other customer sites. The terminal is open during six hours per day. Trucks have to return after 360 minutes. Service at customer sites takes eight minutes. The problem is studied in a static environment. Travel times, waiting times and service times are therefore assumed to be constant. Table I presents the time windows imposed by pickup customers and delivery customers.

delivery i	E_i	L_i	pickup j	E_j	L_j
1	10	100	1	0	100
2	50	250	2	100	280
3	80	180	3	200	260
4	50	360	4	80	320
5	100	300	5	5	80

TABLE I: Customer time windows

Distances, expressed in time units, from the depot 0 to customers and between customers are given in table II. Table III mentions the cost CR_{ijk} of servicing each pair of customers. Fixed costs are assumed to be equal for all vehicles. Since all solutions found for this example require the same number of vehicles, fixed vehicle costs are left out of the comparison.

distance $i j$	0	1	2	3	4	5
0	0	63	48	49	12	44
1	9	57	40	41	14	40
2	47	42	5	11	48	50
3	36	45	13	16	37	43
4	18	56	32	32	18	35
5	24	82	55	53	12	26

TABLE II: Distance matrix

cost $i j$	0	1	2	3	4	5
0	0.0	178.7	138.7	141.3	42.7	128.0
1	34.7	193.3	150.7	221.3	68.0	145.3
2	136.0	224.0	154.7	164.0	164.0	209.3
3	106.7	213.3	150.7	156.0	134.7	185.3
4	58.7	204.0	152.0	153.3	85.3	150.7
5	74.7	246.7	190.7	189.3	85.3	146.7

TABLE III: Cost matrix

A. Construction heuristic

In the pairing phase of the heuristic a maximum waiting time, $MAXWAIT$, of 30 minutes is allowed between serving a delivery customer and a pickup customer. The maximum waiting time is a parameter of the construction heuristic. Its value should be large enough to allow flexibility in the pairing phase, but small enough in comparison with the depot time window of 360 minutes. Equal weights w_1 and w_2 of five are assigned to savings in waiting time and travel time in the first selection criterion (16). This results in the ranked list of feasible pairs of customers given in table IV. Selected pairs are highlighted in bold. In this example all customers can be combined into pairs; no dummy customers are necessary.

In the second phase of the heuristic the selected pairs of customers are inserted into routes. Results are presented in table V. Vehicle 1 returns to the depot after

delivery i	pickup j	selection criterion
2	1	-340
1	5	45
1	1	50
5	3	395
3	3	435
5	2	500
2	3	530
3	2	540
2	2	635
4	3	675
4	2	780
5	4	880
3	4	920
2	4	1015
1	2	1025
4	4	1160
1	4	1405

TABLE IV: Ranked pairs of customers

313 minutes, vehicle 2 after 287 minutes. Travelling costs of this solution are 794.7.

route 1	route 2
2-1	1-5
5-3	3-2
	4-4

TABLE V: Route construction

B. Parameter setting

The selection criterion in the first phase of the heuristic is a weighted combination of two sub-criteria. The initial solutions found by the construction heuristic depend on the values assigned to the weights in the selection criterion. A multi-start approach, assigning different values to these weights, can be applied to find the best overall solution. Solutions found by different parameter settings are presented in table VI. The best solution is obtained in this example when a large weight is assigned to the savings in travel time. The corresponding routes are given in table VII.

weight 1	weight 2	Travelling costs
0	1	776
0.092	0.908	776
0.093	0.907	762.7
0.188	0.812	762.7
0.189	0.811	801.3
0.462	0.538	801.3
0.463	0.537	794.7
1	0	794.7

TABLE VI: Parameter setting

route 1	route 2
2-1	1-5
4-2	3-3
	5-4

TABLE VII: Best routes found by construction heuristic

C. Opportunity costs

When the second selection criterion (17) is applied in the pairing phase of the construction algorithm, the same best solution with a cost of 762.7 is found. Also no further improvement in best solution is found when the opportunity costs of time window slack (18) are taken into account. Assigning various values to the weights in this selection criterion leads to the same best solution with a travelling cost of 762.7.

D. Improvement heuristic

The local search procedure described in section V is applied to the first solution of the construction heuristic given in table V. This initial solution is obtained by giving equal weights in selection criterion (16) and without taking opportunity costs into account. Two CROSS moves are possible in this solution. Table VIII lists the pairs of customers involved and resulting improvements. The second CROSS move is selected. Pair (i, h) is inserted in the first route and pair (g, j) in the second route. The resulting routes, presented in table IX, imply a travelling cost of 758.7, which is lower than the best solution found after parameter setting. The improvement algorithm stops after two further iterations without any improvement in the objective function value.

pair (g, h)	pair (i, j)	I_{ghij}
5-3	3-2	-5
5-3	4-4	27

TABLE VIII: CROSS moves

route 1	route 2
2-1	1-5
4-3	3-2
	5-4

TABLE IX: Best routes found by improvement heuristic

E. Bound

In order to evaluate the results of the heuristics, a lower bound for the optimal solution is computed. The problem is relaxed by ignoring the final group of constraints which determine the feasibility of the routes. The solution of the relaxed problem gives a lower bound to the problem including all constraints. The optimal

solution for the relaxed problem formulation has a travelling cost of 758.7. The routes are shown in table X. These routes are not feasible when taking constraints (12) to (15) into account.

route 1	route 2
2-1	1-5
3-2	
4-3	
5-4	

TABLE X: Best routes of relaxed problem definition

We can conclude that the improvement heuristic is able to find the optimal solution in this numerical example. A comparison of table IX and table X shows that the same pairs of customers are selected, but the assignment of pairs to routes differs. Whereas the routes are not feasible in the solution of the relaxed problem, they are in the solution of the improvement heuristic.

VII. CONCLUSIONS

In this study a special class of pickup and delivery problems has been explored, in which vehicles carry full truckloads to and from an intermodal terminal. An insertion heuristic consisting of two phases is proposed. The two-phase construction heuristic is able to find a feasible solution in a short time span. This solution can be further improved by a local search procedure based on the CROSS operator. Further research is necessary to test the heuristic on large problems. Other neighborhood definitions in the local search procedure may also be investigated. A tabu search algorithm based on reactive tabu search can be developed. Results of the tabu search procedure can be compared with the heuristic described in this paper. Finally, the problem can be investigated in a stochastic environment.

REFERENCES

- [1] R.H. Currie and S. Salhi. Exact and heuristic methods for a full-load, multi-terminal, vehicle scheduling problem with backhauling and time windows. *Journal of the Operational Research Society*, 54:390–400, 2003.
- [2] R.H. Currie and S. Salhi. A tabu search heuristic for a full-load, multi-terminal, vehicle scheduling problem with backhauling and time windows. *Journal of Mathematical Modelling and Algorithms*, 3:225–243, 2004.
- [3] Yvan Dumas, Jacques Desrosiers, and François Soumis. The pickup and delivery problem with time windows. *European Journal of Operational Research*, 54:7–22, 1991.
- [4] Manfred Gronalt, Richard F. Hartl, and Marc Reimann. New savings based algorithms for time constrained pickup and delivery of full truckloads. *European Journal of Operational Research*, 151:520–535, 2003.
- [5] Akio Imai, Etsuko Nishimura, and John Current. A Lagrangian relaxation-based heuristic for the vehicle routing with full container load. *European Journal of Operational Research*, 176(1):87–105, 2007.
- [6] Antoine Landrieu, Yazid Mati, and Zdenek Binder. A tabu search heuristic for the single vehicle pickup and delivery problem with time windows. *Journal of Intelligent Manufacturing*, 12:497–508, 2001.
- [7] Quan Lu and Maged M. Dessouky. A new insertion-based construction heuristic for solving the pickup and delivery problem with time windows. *European Journal of Operational Research*, 175:672–687, 2006.

- [8] William P. Nanry and J. Wesley Barnes. Solving the pickup and delivery problem with time windows using reactive tabu search. *Transportation Research Part B*, 34:107–121, 2000.
- [9] M.W.P. Savelsbergh and M. Sol. The general pickup and delivery problem. *Transportation Science*, 29(1):17–29, 1995.

BIOGRAPHY

An Caris graduated as Master of Business Economics with a major in Operations Management and Logistics at the Limburg University Centre (LUC), Belgium, in 2003. After one year of practical experience in inventory management at Reynaers Aluminium, she started as a teaching assistant in econometrics and operations research at the Hasselt University (UHasselt), Belgium, in September 2004. In addition she is preparing a Ph.D. in Applied Economic Sciences. She is a member of the research group Data Analysis and Modelling and the Transportation Research Institute (IMOB) of the Hasselt University. She takes a research interest in modelling intermodal freight transport networks with a focus on inland navigation.

Gerrit K. Janssens received degrees of M.Sc. in Engineering with Economy from the University of Antwerp (RUCA), Belgium, M.Sc. in Computer Science from the University of Ghent (RUG), Belgium, and Ph.D. from the Free University of Brussels (VUB), Belgium. After some years of work at General Motors Continental, Antwerp, he joined the University of Antwerp until the year 2000. Currently he is Professor of Operations Management and Logistics at Hasselt University within the Faculty of Business Administration. He also holds the CPIM certificate of the American Production and Inventory Control Society (APICS). During the last fifteen years he has repeatedly been visiting faculty member of universities in Thailand, Vietnam, Philippines, Cambodia and Zimbabwe. His main research interests include the development and application of operations research models in production and distribution logistics.