สกว
TRF

รายงานวิจัยฉบับสมบูรณ์

โครงการการออกแบบโซ่อุปทานเชิงบูรณาการ
(Integrated Supply Chain Design)

โดย

ผศ. ดร. อำพล การุณสุนทวงษ์

ธันวาคม พ.ศ. 2555

สัญญาเลขที่ MRG5380066

รายงานวิจัยฉบับสมบูรณ์

โครงการการออกแบบโซ่อุปทานเชิงบูรณาการ
(Integrated Supply Chain Design)

ผศ. ดร. อำพล การุณสุนทวงษ์
ภาควิชาวิศวกรรมโยธา คณะวิศวกรรมศาสตร์
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

สนับสนุนโดยสำนักงานกองทุนสนับสนุนการวิจัย

Project Code (รหัสโครงการ): MRG5380066

Project Title (ชื่อโครงการ): การออกแบบโซ่อุปทานเชิงบูรณาการ
(Integrated Supply Chain Design)

Investigator (ชื่อนักวิจัย) : ผศ. ดร. อำพล การุณสุนทวงษ์
ภาควิชาวิศวกรรมโยธา คณะวิศวกรรมศาสตร์
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

E-mail Address : ampolk@gmail.com , ampol.kar@kmutt.ac.th

Project Period (ระยะเวลาโครงการ): 16 มิถุนายน 2553 - 15 มิถุนายน 2555

บทคัดย่อ

งานวิจัยนี้ศึกษาปัญหาการจัดการสินค้าคงคลังและการจัดเส้นทางขนส่งสินค้า ในโซ่อุปทานแบบสองระดับ ซึ่งประกอบด้วยโรงงานหนึ่งแห่งให้บริการกลุ่มคลังสินค้า ซึ่งให้บริการกลุ่มลูกค้าที่มีความต้องการสินค้าที่มีความไม่แน่นอน งานวิจัยนี้เสนอแบบจำลองคณิตศาสตร์แบบจำนวนเต็มไม่เชิงเส้นฐานการแบ่งเซต สำหรับปัญหาเชิงบูรณาการของการควบคุมสินค้าคงคลังแบบต่อเนื่องและการจัดเส้นทางขนส่งสินค้าแบบหลายท่ารถ ซึ่งคำนึงถึงความน่าจะเป็นของการละเมิดความจุสินค้าคงคลัง ความจุของปริมาณการสั่งซื้อ ระดับบริการ ข้อจำกัดความจุพาหนะและขีดจำกัดระยะเวลาของเส้นทางขนส่งสินค้า งานวิจัยนี้เสนอวิธีฮิวริสติกแบบทาบูเสิร์ชจำนวน 2 อัลกอริธึม ซึ่งแตกต่างกันที่วิธีการสร้างคำตอบตั้งต้น การทดสอบเชิงคำนวณสำหรับโครงข่ายทดสอบมาตรฐาน แสดงให้เห็นว่าการบูรณาการการตัดสินใจด้านการจัดการสินค้าคงคลังและการจัดเส้นทางขนส่งสินค้า โดยการหาคำตอบของปัญหาเชิงบูรณาการนี้ อาจให้การประหยัดต้นทุนสูงถึง 14 เปอร์เซนต์ เมื่อเทียบกับการหาคำตอบแบบตามลำดับ ซึ่งทำการหาคำตอบของปัญหาแบบแยกกัน ค่าฟังก์ชันวัตถุประสงค์ที่ดีที่สุดที่หาจากวิธีฮิวริสติกทาบูเสิร์ชถูกพบว่าจะมีค่ามากขึ้น เมื่อความแปรปรวนของความต้องการสินค้าของลูกค้ามีค่ามากขึ้น แต่มีค่าลดลง เมื่อความจุของปริมาณการสั่งซื้อ และ ขีดจำกัดระยะเวลาของเส้นทางขนส่งสินค้า มีค่ามากขึ้น ระดับสินค้าคงคลังปลอดภัย ระดับสินค้าสั่งซื้อใหม่ และต้นทุนถือครองสินค้า ถูกพบว่าจะมีค่ามากขึ้น เมื่อความแปรปรวนของความต้องการสินค้ามีค่ามากขึ้น ความจุสินค้าคงคลังที่เหลืออยู่ ถูกพบว่าจะมีค่าลดลงเมื่อความแปรปรวนความต้องการสินค้ามีค่ามากขึ้น ต้นทุนการสั่งซื้อจะมีค่ามากขึ้นเมื่อความจุของปริมาณการสั่งซื้อมีค่าลดลง ต้นทุนการถือครองสินค้าจะมีค่าลดลงเมื่อความจุของปริมาณการสั่งซื้อมีค่าลดลง ต้นทุนการจัดเส้นทางขนส่งสินค้ามีค่าเพิ่มขึ้นเมื่อขีดจำกัดระยะเวลาของเส้นทางมีค่าลดลง

คำหลัก : การควบคุมสินค้าคงคลังแบบต่อเนื่อง การจัดเส้นทางขนส่งสินค้าแบบหลายท่ารถ วิธีทาบูเสิร์ช

Abstract

This research studies the inventory management and routing problem in a two-level supply chain where a single plant serves a set of warehouses, which in turn serve a set of customers with stochastic demands. A set partitioning based probabilistic chance constrained nonlinear integer programming formulation is provided for the combined continuous inventory control and multi-depot vehicle routing problem while accounting for probability of inventory capacity violation, order quantity capacity, service levels, vehicle capacity restrictions and route duration limits. Two tabu search heuristics, differing in the way initial solutions are generated, are applied to solve the problem. Computational tests on standard tests networks reveal that integrating the inventory management and routing decisions by solving the combined inventory management and routing problem may yield cost savings of up to 14% over the sequential approach where both problems are solved separately. The best objective function value obtained by the tabu search heuristic was found to increase with increase in customer demand variance but decrease with increase in order quantity capacity and route duration limit. The safety stock levels, the reorder points and total holding costs were found to increase with increase in customer demand variance. The available inventory capacity was found to decrease with increase in customer demand variance. The total ordering costs in the best solution increases with the decrease of the order quantity capacity, whereas the total holding costs decreases with the decrease of the order quantity capacity. The routing costs increases with the decrease of route duration limit.

Keywords : Continuous Inventory Control, Multi-Depot Vehicle Routing Problem, Tabu Search

EXECUTIVE SUMMARY

This research studies a two-level supply chain where a single plant supplies a single commodity to a set of warehouses which in turn serve a set of customers with stochastic demands. This research provides a combined stochastic chance constrained nonlinear integer programming formulation modeling the inventory management decisions at the warehouses and the routing of goods from the warehouses to the customers. The warehouses are assumed to manage the inventory using a continuous inventory policy. The model accounts for the service level at each warehouse which reflects the probability of available inventory meeting the demand during the lead time, probability of violation of inventory capacity, and restrictions on order quantity volume. The routing of goods from warehouse to customers is modeled as a route duration constrained capacitated multi-depot vehicle routing problem. Two tabu search heuristics – type 1 and type 2, differing primarily in the way initial solutions are generated are developed to solve the combined model. The optimal order quantity at each warehouse is approximated using the KKT conditions. Computational runs are conducted on variations of the standard Solomon test instances available for vehicle routing problems with time windows. Type 2 tabu search was found to outperform type 1 tabu search for the 100 customer instance. For smaller customer instances, both the heuristics were found to perform equally well. Integrating the inventory management and routing decisions by solving the combined inventory management and routing problem was found to yield cost savings of up to 14% over the sequential approach where both problems are solved separately. The best objective function value obtained by the tabu search heuristic was found to increase with increase in customer demand variance, decrease with increase in order quantity capacity and route duration limit. Variance of the customer demand was found to have significant impact on the solution quality. The safety stock levels, the reorder points and the total holding costs were found to increase with increase in customer demand variance. As expected, the available inventory capacity was found to decrease with increase in customer demand variance. It is unclear how the routing and ordering costs change with the demand variances. This is because the demand variance can influence the customer assignments to different warehouses, resulting in different routing costs and ordering costs. We found that the order quantity capacity and inventory capacity play a role in the trade-off between total holding costs and

total ordering costs. The total ordering costs in the best solution increases with the decrease of the order quantity capacity, whereas the total holding costs decreases with the decrease of the order quantity capacity. The routing costs increase with the decrease of route duration limit. Thus, the combined inventory management and routing model can be used to study the tradeoffs between inventory holding costs, ordering costs, and routing costs.

Table of Contents

Chapter 1 Introduction

Fierce competition in today's global market together with the global economic recession and fuel-price fluctuation is forcing companies to better design and manage their supply chain networks. An efficient supply chain design can decrease the system costs such as inventory control costs and transportation costs, and more importantly it helps save energy and reduce emissions. In this report, we consider a two-level supply chain, in which a single plant serves a set of warehouses, which in turn serve a set of end customers with stochastic demands. Inventory control decisions and vehicle routing decisions are made at the operational level for each warehouse. The inventory control problem (ICP) determines optimal order quantity, reorder point and safety stock, so that the total ordering and holding costs are minimal. The multi-depot vehicle routing problem (MDVRP) determines an optimal set of vehicle routes for each depot to satisfy demands such that the routing costs are minimal. Typically, these two problems are solved sequentially. Indeed, ICP and MDVRP are interrelated. The inventory control decisions for a warehouse depend on the demands incurred at this warehouse, which are determined from the demands of customers assigned to this warehouse. The MDVRP decisions aim at minimizing routing costs without considering the impact of the customer assignment on the ordering and holding costs at warehouses. Therefore there is significant potential to optimize the supply chain costs by solving ICP and MDVRP simultaneously (a.k.a. inventory routing problem: IRP).

1.1 Literature Review

Depending on the nature of the application, several variants of IRPs have been studied in the literature. Numerous studies focus on IRP application in a Vendor Managed Inventory (VMI) setting where a single vendor delivers goods to multiple customers and coordinates the routing and delivery decisions so that the customer always has sufficient inventory (Bertazzi et al., 2002; Campbell and Savelsbergh, 2004). Depending on the nature of the time horizon for the decision making – IRP can be classified into single day (Beltrami and Bodin, 1974; Federgruen and Zipkin, 1984), multi-day (Dror et al., 1985; Dror and Ball, 1987) or an long term horizon operational problem (Anily and Federgruen, 1993; Bard et al., 1998; Jaillet et al., 2002; Gaur and Fisher, 2004). Normally the long term horizon problem use frequency as the decision variable and the shorter duration studies are normally time based. In the context of long term operational problem, several studies have evaluated the effectiveness of delivery policies using asymptotic analysis in an infinite time period (Anily and Federgruen, 1990; Gallego and Simchi-Levi, 1990; Bramel and Simchi-Levi, 1995). Note that due to the complexities of the IRP and based on the nature of the application, several studies have focused on optimally timing the deliveries to a single customer (Dror et al., 1985; Dror and Ball, 1987; Bard et al., 1998). Savelsbergh and Song (2007, 2008) studied variants where the customers can be served by multiple facilities depending on product availability. Federgruen and Simchi-Levi (1995), Campbell et al. (1998) and Bertazzi et al. (2008) provide a detailed review of the IRP variants and their solution methods. This research is different from the past works as in our work the customers can be served by one among multiple warehouses. Moreover we do not adopt a VMI approach. In our model, the inventories are located at warehouses and not at the customers.

Similar to Miranda and Garrido (2004, 2006), we assume that each warehouse follows the continuous inventory control policy, and we explicitly consider the probabilities of unfulfilled demands, the probabilities of inventory capacity violation and the order quantity capacity. The considered policy does not penalize unfulfilled demands. Rather, a reorder point is determined such that after order submission to the plant the inventory level should cover the demand generated during the lead time with probability. Since the cost of alternative storage space especially in the urban areas is high, it is essential to control the level of service associated with the inventory capacity. The probabilities of inventory capacity violation are employed in the chance constrained stochastic programming framework. The vehicle capacity restrictions are common in the urban areas, and this can be taken into account by setting order quantity capacity and through capacity constraints in the routing problem. In MDVRP, we explicitly consider the route duration limit which arises in a number of applications such as perishable goods delivery problems (Gorr et al., 2001) and time-critical delivery problems (Berger et al., 2007).

1.2 Objectives

The objectives of this study are three-fold. The first objective is to formulate the model for the combined continuous inventory control and MDVRP accounting for route duration limits and stochastic inventory capacity constraints. The second objective is to develop tabu search heuristics to solve the problem. The third objective is to compare the performances of the proposed tabu search algorithms with each other as well as against the sequential approach on hypothetical test networks based on Solomon (1987)'s test problems.

Chapter 2 Mathematical Formulation

The inventory routing model is developed based on the works by Miranda and Garrido (2006) and Berger et al. (2007). This combined model is a set partitioning-based formulation that has the stochastic inventory capacity constraints and the order quantity capacity constraints. Daily delivery demands of customers are assumed independent and normally distributed. Each customer is served on exactly a route by a warehouse, and a single commodity is considered.

2.1 Continuous Inventory Control Policy

The proposed model embeds the continuous inventory control policy, which is briefly reviewed here. At any warehouse *i*, we assume a continuous inventory control policy (Q_i, RP_i) to meet normally distributed random demand $\tilde{D}_i$ with the mean of ED_i (product units per day) and the variance of VD_i (squared product unit per day). ED_i and VD_i are variables, since they depend on the customers assigned to each warehouse *i*. Q_i is the order quantity at warehouse *i*, and RP_i is the reorder point at warehouse *i*. The plant takes a lead time LT_i to fulfill an incoming order from warehouse *i*. The evolution of the inventory level at warehouse *i* is shown in Figure 2.1. When the inventory level falls below RP_i, an order of Q_i units is triggered, which is received after LT_i time units. When an order is submitted to the plant, the inventory level should cover the demand generated during the lead time LT_i, with probability 1-α. This probability is known as the service level for the system Miranda and Garrido (2004).

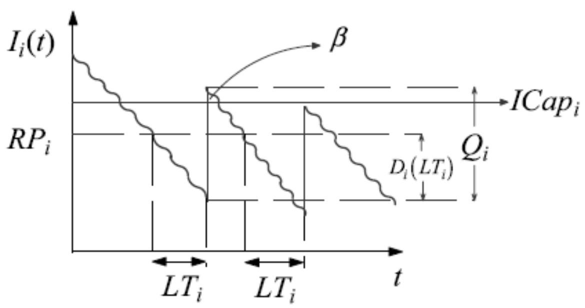


Figure 2.1. Continuous Inventory Control Policy with Stochastic Inventory Capacity Constraint (Miranda and Garrido, 2006)

The service-level constraint is:

$$\text{Prob}(\tilde{D}_i(LT_i) \le RP_i) = 1-\alpha \tag{2.1}$$

where $\tilde{D}_i \cdot LT_i$ is the normally distributed random demand generated during the lead time at warehouse i with the mean of $ED_i \cdot LT_i$ and the variance of $VD_i \cdot LT_i$. Eq.(2.1) can be standardized:

$$\text{Prob}(z = \frac{\tilde{D}_i \cdot LT_i - ED_i \cdot LT_i}{\sqrt{VD_i \cdot LT_i}} \le \frac{RP_i - ED_i \cdot LT_i}{\sqrt{VD_i \cdot LT_i}} = z_{1-\alpha}) = 1-\alpha$$

where $z_{1-\alpha}$ is the value of the standard normal distribution, which accumulates a probability of 1-α. Then, RP_i can be determined by the equation:

$$RP_i = ED_i \cdot LT_i + z_{1-\alpha}\sqrt{LT_i}\sqrt{VD_i} \tag{2.2}$$

The parameter $z_{1-\alpha}$ is assumed fixed for the entire network, determining a homogeneous service level for the whole system. The first term in Eq.(2.2) is the average demand during the lead time, and the second term ($Z_{1-\alpha}\sqrt{LT_i}\sqrt{VD_i}$) is the average safety stock. Given that HC_i is the holding cost per time unit for warehouse i (\$/unit/day), and OC_i is the fixed ordering cost at warehouse i (\$/order), the expected holding and ordering cost rate for each warehouse i (\$/day) is:

$$HC_i \cdot z_{1-\alpha}\sqrt{LT_i}\sqrt{VD_i} + \frac{1}{2}HC_i \cdot Q_i + \frac{OC_i}{Q_i}ED_i \tag{2.3}$$

The first term in Eq.(2.3) is the average safety stock cost at warehouse i. The last two terms in Eq.(2.3) represent the costs of the known Economic Order Quantity (EOQ) model (Erlenkotter, 1990). This is the average inventory and ordering cost incurred due to the ordering process, if the order size is always Q_i. The peak inventory levels take place when the orders arrive at warehouse, and equal to $RP_i - \tilde{D}_i \cdot LT_i + Q_i$.

When setting maximum probability β to violate the inventory capacity $I_i^{\max}$ at peak levels, the inventory capacity constraint can be written as chance constraints (Miranda and Garrido, 2006):

$$\text{Prob}(RP_i - \tilde{D}_i \cdot LT_i + Q_i \leq I_i^{\max}) \geq 1 - \beta \quad \forall i \in V_{WH} \tag{2.4}$$

Eq.(2.4) can be rewritten as nonlinear inequalities (Miranda and Garrido, 2004, 2006), which are the stochastic inventory capacity constraints:

$$Q_i + (z_{1-\alpha} + z_{1-\beta}) \cdot \sqrt{LT_i} \cdot \sqrt{VD_i} \leq I_i^{\max} \quad \forall i \in V_{WH} \tag{2.5}$$

2.2 Proposed Mathematical Program

The sets, parameters and decision variables are defined, followed by the proposed mathematical formulation.

Sets

V_{CUS} = set of customer locations = {1,2,…, n_{CUS}}

V_{WH} = set of warehouse locations = {1,2,…, n_{WH}}

P_i = set of all feasible routes (with respect to route duration limit and vehicle capacity restriction) associated with warehouse *i*

Parameters

μ_j =mean of the daily demand for customer *j*

σ_j^2 = variance of the daily demand for customer *j*

n_{WH} = number of warehouse locations

n_{CUS} = number of customers to be served

RC_i = transportation unit cost between the plant and warehouse *i* (\$/unit/day)

$Q_i^{\max}$ = maximum order quantity (order quantity capacity) for warehouse *i*

$I_i^{\max}$ = inventory capacity for warehouse *i*

LT_i = lead time that the plant takes to fulfill an incoming order from warehouse i

OC_i = fixed ordering cost at warehouse i ($/order)

HC_i = holding cost per day per product unit at warehouse i ($/unit-day)

$Z_{1-\alpha}$ = value of standard normal distribution that accumulates the probability 1-α

$Z_{1-\beta}$ = value of standard normal distribution that accumulates the probability 1-β

a_{jik} = 1 if route k associated with warehouse i visits customer j; 0 otherwise

d_{ik} = cost of route k associated with warehouse i

Decision Variables

y_{ik} = 1 if route k associated with warehouse i is chosen; 0 otherwise.

Q_i = order quantity for warehouse i

ED_i = mean of the served daily demand by warehouse i

VD_i = variance of the served daily demand by warehouse i

Z = total costs

Model

$$\min Z = \sum_{i \in V_{WH}} \sum_{k \in P_i} d_{ik} \cdot y_{ik} + \sum_{i \in V_{WH}} \sum_{j \in V_{CUS}} \sum_{k \in P_i} a_{jik} \cdot RC_i \cdot \mu_j \cdot y_{ik} + \sum_{i \in V_{WH}} \left(\frac{OC_i}{Q_i} ED_i \right)$$

$$+ \sum_{i \in V_{WH}} \left(HC_i \frac{Q_i}{2} + HC_i \cdot Z_{1-\alpha} \cdot \sqrt{LT_i} \cdot \sqrt{VD_i} \right) \qquad (2.6.1)$$

Subject to

$$\sum_{i \in V_{CWH}} \sum_{k \in P_i} a_{jik} \cdot y_{ik} = 1 \qquad \forall j \in V_{CUS} \qquad (2.6.2)$$

$$Q_i + (Z_{1-\alpha} + Z_{1-\beta}) \cdot \sqrt{LT_i} \cdot \sqrt{VD_i} \le I_i^{\max} \qquad \forall i \in V_{WH} \qquad (2.6.3)$$

$$\sum_{j \in V_{CUS}} \sum_{k \in P_i} \mu_j \cdot a_{jik} \cdot y_{ik} = ED_i \qquad \forall i \in V_{WH} \qquad (2.6.4)$$

$$\sum_{j \in V_{CUS}} \sum_{k \in P_i} \sigma_j^2 \cdot a_{jik} \cdot y_{ik} = VD_i \qquad \forall i \in V_{WH} \qquad (2.6.5)$$

$$y_{ik} \in \{0,1\} \qquad \forall i \in V_{WH}, \forall k \in P_i \qquad (2.6.6)$$

$$0 \le Q_i \le Q_i^{\max} \qquad \forall i \in V_{WH} \qquad (2.6.7)$$

Objective (2.6.1) calculates the total costs Z composed four terms - total MDVRP costs, total direct transportation costs between the plant and warehouses, total expected ordering costs and total expected holding costs, respectively. Constraints (2.6.2) enforce that each customer is served on exactly a route by a warehouse. Constraints (2.6.3) are non-linear constraints assuring that the inventory capacity for each warehouse is satisfied at least with probability 1-β and that the reorder point can cover the stochastic demand during the lead time with probability 1-α. Constraints (2.6.4) determine the mean of the served demands assigned to each warehouse. Constraints (2.6.5) determine the variance of the served demands assigned to each warehouse. Constraints (2.6.4) and (2.6.5) result from the assumption that demands are independent and normally distributed across the customers; thus all the covariance terms are zero. Constraints (2.6.7) constrain the order quantity to be within the order quantity capacity, which is assumed homogeneous for each warehouse, and can be set as the vehicle (from plant to warehouse) capacity.

The VRP is NP-hard (Lenstra and Rinnooy Kan, 1981), which is a special case of the IRP. Thus, IRP is also NP-hard. The proposed formulation potentially contains an exponential number of variables (y_{ik}), and there exists nonlinearity in Eq.(2.6.1) and Eq.(2.6.3). In effect, there is not an efficient solution method that guarantees an optimal solution, and this essentially requires a metaheuristic approach. In this research, we propose tabu search heuristics.

Chapter 3 Tabu Search Heuristics

The overview of tabu search can be found in Glover and Laguna (1997). It integrates a hill-climbing search technique, which is based on a set of elementary moves, and a heuristic to avoid the stops at local optima and the occurrence of cycles. The tabu search was initially created with constant tabu tenure by Glover (1989); then, the proper choice of tabu tenure is critical to the success of the algorithm. The tabu tenure is the number of iterations that the algorithm prohibits past moves (a move is a process that the algorithm uses to change the current solution to the new solution), so that the algorithm will not visit the same past solutions (so a cycle is prevented) and will be able to depart from local optima. The tabu tenure should be sufficiently long to prevent cycles but short enough such that the search is not overly constrained.

In this report, we modify the tabu search heuristic for MDVRP by (Renaud et al., 1996) in order to incorporate the continuous inventory control policy for warehouses in the two-level supply chain, accounting for route duration limits and stochastic inventory capacity constraints. Let $G=(V,A)$ be a directed graph. $V=\{V_{WH},V_{CUS}\}$ is a vertex set where $V_{WH}=\{v_{01},v_{02},...,v_{0n_{WH}}\}$ is the set of warehouse (or depot) locations and $V_{CUS}=\{v_1,v_2,...,v_{n_{CUS}}\}$ is the set of customers. $A=\{(v_i,v_j):i\neq j\}$ is an arc set. Vertex $v_{0i}\in V_{WH}$ denotes a warehouse where m_i identical vehicles are based. m_i is assumed unlimited. Vertex $v_j\in V_{CUS}$ denotes a customer. With every arc (v_i,v_j) is associated a fixed nonnegative distance c_{ij}. $V^i=\{v_0^i,v_1^i,...,v_{n_{CUS}^i}^i\}$ is the vertex set associated with warehouse *i*; v_0^i a warehouse vertex; n_{CUS}^i the number of customers assigned to warehouse *i*.

A least cost solution is determined such that:

- Total cost is minimized, including direct transport cost between the plant and warehouses, MDVRP costs from warehouses to customers, ordering costs, and inventory holding costs.
- The order quantity from warehouse v_{0i} to the plant may not exceed its maximum value $Q_{0i}^{\max}$.

- When an order is submitted to the plant by a warehouse, the reorder point can cover the stochastic demand generated during the lead time with probability 1-α.
- For each warehouse, the inventory level at peak levels may violate the inventory capacity with the maximum probability β.
- A route starts and ends at a warehouse.
- Each Customer in V_{CUS} is visited exactly once by exactly a vehicle based at a warehouse.
- Customer v_j has an independent and normally distributed demand with the mean of μ_j and variance of σ_j^2, whereas each warehouse v_{0i} has a fixed zero demand.
- The total average daily demands served by a vehicle based at warehouse v_{0i} may not exceed the vehicle capacity $RD_{0i}^{\max}$.
- Each city v_j requires a fixed service time δ_j, and each warehouse v_{0i} has no service time.
- The duration (travel plus service times) of any route beginning at warehouse v_{0i} and ending at the last customer visited on this route may not exceed the route duration limit $L_{0i}^{\max}$.

The tabu search algorithm consists of two phases: (1) construction of an initial solution and (2) solution improvement as shown in Figure 3.1. Inspired by Campbell and Savelsbergh (2004), we maintain the following information in our implementation in order to save computational efforts:

- For every route r_1 and warehouse i_1, the sum of the average delivery quantities currently assigned to this route is $q_{r_1}^{i_1}$; the duration (travel plus service time) of round-trip route r_1 beginning and ending at warehouse i_1 is $rl_{r_1}^{i_1}$; the duration (travel plus service time) of route r_1 beginning at warehouse i_1 and ending at the last customer visited in route r_1 is $pl_{r_1}^{i_1}$.

- For every warehouse i_1, the sum of average currently served demands is ED_{i_1}; the sum of currently served demand variances is VD_{i_1}.

With such information maintained, it is easy to verify the route feasibility of inserting a customer into route r_1 associated with warehouse i_1; i.e., check whether $pl_{r_1}^{i_1} \le L_{0i}^{\max}$ and $q_{r_1}^{i_1} \le RD_{0i}^{\max}$.

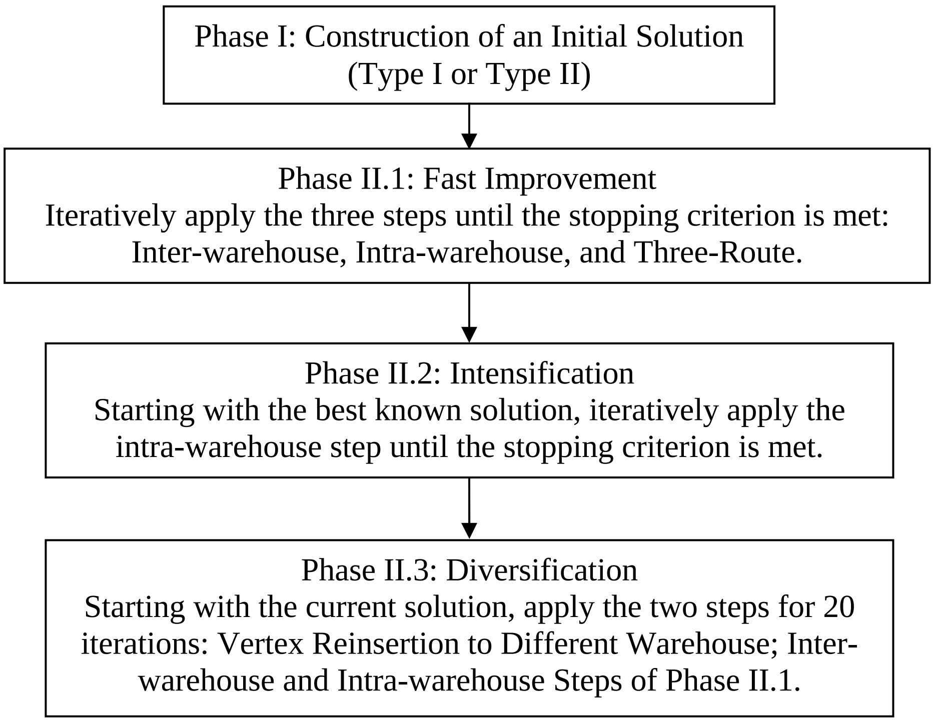


Figure 3.1 Flowchart of Proposed Tabu Search Heuristics

Before describing the two phases of tabu search algorithm, the heuristic approximation for the continuous ICP with order quantity capacity and stochastic inventory capacity constraints is described when the currently served average demands (ED_i) and demand variances (VD_i) at warehouses are known.

3.1 Heuristic Approximation for (Q_i, RP_i)

When the means and variances of currently served demands (ED_i and VD_i) for warehouses are known, the continuous inventory control policies (Q_i, RP_i) with stochastic inventory capacity constraints and order quantity capacity constraints can be heuristically approximated. In the inventory location models studied in Miranda and Garrido (2004) and Daskin et al. (2002), the order quantity (in the former) and the total orders (in the latter) are obtained through the first order optimality conditions of the objective function. Thus, the outcome is analogous to the result of the EOQ model and the corresponding ordering decisions (Q_i) are not variables and hence eliminated from the model. However, in our formulation, there are constraints on Q_i (see constraints 2.6.3 and 2.6.7). Two decision variables for the continuous ICP are order quantities (Q_i) and reorder points (RP_i). RP_i can be determined Eq.(2.2) when ED_i and VD_i are known. The heuristic approximation of an optimal order quantity for warehouse i (Q_i^*) is described below.

If constraints (2.6.3) and (2.6.7) are removed, Q_i^* can be approximated through the first order optimality condition. When the constraints on Q_i are taken into account, the first order optimality conditions for a constrained minimum is employed to approximate Q_i^*. The standard form of a minimization program is

$$\min Z(x)$$
$$\text{subject to} \quad g_j(x) \geq b_j \quad \forall j \quad : u_j$$

where $Z(x)$ is the objective function; x is the vector of decision variables; b_j is a constant; $g_j(x)$ is the function of x in constraint j; u_j is the dual variable associated with constaint j. Then, the first-order conditions (a.k.a. Karush-Kuhn-Tucker Conditions) (Luenberger, 1973) are:

$$\frac{\partial Z(x^*)}{\partial x_i} = \sum_j u_j \frac{\partial g_j(x^*)}{\partial x_i} \quad \forall i$$
$$u_j \geq 0 \qquad \forall j$$
$$u_j(b_j - g_j(x^*)) = 0 \qquad \forall j$$
$$g_j(x^*) \geq b_j \qquad \forall j$$

The order quantity Q_i correspond to x in the standard form. Constraints (2.6.3) and (2.6.7) can be written in the standard form as:

$$-Q_i \geq (z_{1-\alpha} + z_{1-\beta})\sqrt{LT_i}\sqrt{VD_i} - I_i^{\max} \qquad \forall i \in V_{WH} : u_{1i} \qquad (3.1.1)$$

$$-Q_i \geq -Q_i^{\max} \qquad \forall i \in V_{WH} : u_{2i} \qquad (3.1.2)$$

$$Q_i \geq 0 \qquad \forall i \in V_{WH} : u_{3i} \qquad (3.1.3)$$

where u_{1i}, u_{2i} and u_{3i} are dual variables associated with Eq.(3.1.1)-(3.1.3). The Karush-Kuhn-Tucker (KKT) conditions for the minimum program (2.6.1), (3.1.1)-(3.1.3) where only Q_i are decision variables, are:

$$\frac{\partial Z(Q^*)}{\partial Q_i} = -u_{1i} - u_{2i} + u_{3i} \qquad \forall i \in V_{WH} \qquad (3.2.1)$$

$$u_{1i} \geq 0; u_{2i} \geq 0; u_{3i} \geq 0 \qquad \forall i \in V_{WH} \qquad (3.2.2)$$

$$u_{1i} \cdot \left(Q_i^* + (z_{1-\alpha} + z_{1-\beta})\sqrt{LT_i}\sqrt{VD_i} - I_i^{\max}\right) = 0 \qquad \forall i \in V_{WH} \qquad (3.2.3)$$

$$u_{2i} \cdot (Q_i^* - Q_i^{\max}) = 0 \qquad \forall i \in V_{WH} \qquad (3.2.4)$$

$$u_{3i} \cdot Q_i^* = 0 \qquad \forall i \in V_{WH} \qquad (3.2.5)$$

$$-Q_i^* \geq (z_{1-\alpha} + z_{1-\beta})\sqrt{LT_i}\sqrt{VD_i} - I_i^{\max} \qquad \forall i \in V_{WH} : u_{1i} \qquad (3.2.6)$$

$$-Q_i^* \geq -Q_i^{\max} \qquad \forall i \in V_{WH} : u_{2i} \qquad (3.2.7)$$

$$Q_i^* \geq 0 \qquad \forall i \in V_{WH} : u_{3i} \qquad (3.2.8)$$

For any warehouse with served demands, the optimal order quantity is naturally greater than zero. Then, Eq.(3.2.5) implies that u_{3i} equal to 0. Then, Eq.(3.2.1) become:

$$\frac{\partial Z(Q^*)}{\partial Q_i} = -u_{1i} - u_{2i} \qquad (3.2.1a)$$

This implies that the stationary point with the property $\frac{\partial Z}{\partial Q_i} = 0$ can be either within the feasible range of Q_i or greater than the feasible range of Q_i. The stationary point cannot be less than the feasible range of Q_i; otherwise, $\frac{\partial Z(Q^*)}{\partial Q_i}$ becomes positive, given that $Z(Q)$ is assumed convex with respect to Q_i. When the stationary point is within the feasible range of Q_i, the minimal point is the stationary point. Eq.(3.2.3) and (3.2.4) imply that $u_{1i} = 0$ and $u_{2i} = 0$, and Eq.(3.2.1) yields $\frac{\partial Z(Q^*)}{\partial Q_i} = 0$. When the stationary point is greater than the feasible range of Q_i, the minimal point is not the stationary point. Eq.(3.2.1a) and the assumed convexity of $Z(Q)$ imply that the minimal point is at the boundary of either Eq.(3.2.6) or Eq.(3.2.7). Thus, Q_i^* can be determined from the equation:

$$Q_i^* = \min\left\{ \sqrt{\frac{2OC_i \cdot ED_i}{HC_i}}, \min\left\{ Q_i^{\max}, I_i^{\max} - (Z_{1-\alpha} + Z_{1-\beta})\sqrt{LT_i}\sqrt{VD_i} \right\} \right\} \quad (3.3)$$

3.2 Phase I: Construction of an initial solution

The first phase of the proposed tabu search constructs an initial solution as follows.

Step I.1. Each customer is assigned to its nearest warehouse. Then, for each warehouse, sort assigned customers in increasing order of the angle that they make with the warehouse and a horizontal line.

Step I.2. Create initial vehicle routes for each warehouse. This will be described in the next subsections.

Step I.3. Determine RP_i and Q_i, using Eq.(2.2) and (3.3), respectively.

Step I.4. Determine the objective function value of the initial solution, using Eq.(2.6.1)

We consider two alternatives to create initial routes in Step I.2. The initial solution type 1 is based on Cordeau et al. (1997) and the initial solution type 2 based on Gendreau et al. (1994).

Construction of Initial Solution Type 1

For each warehouse i=1,…, n_{WH} , do

(a) Let v_j^i be a customer randomly chosen among those closest to warehouse i (vertex v_0^i)

(b) Set m_i =1

(c) Using the customer vertex sequence $(v_j^i, v_{j+1}^i, ..., v_{n_{CUS}^i}^i, v_1^i, ..., v_{j-1}^i)$, perform the following steps for every customer assigned to warehouse i to obtain an initial routing solution, $S_{MDVRP} = \{S_{SDVRP}^i \forall i \in V_{WH}\}$:

- Insert each customer into the route m_i based at warehouse i (vertex v_0^i) using the generalized insertion (GENI) algorithm by Gendreau et al. (1992).
- If the insertion of customer in the route m_i would result in the violation of vehicle capacity or route duration limit, set m_i = m_i +1.

Construction of Initial Solution Type 2

For each warehouse i=1,…, n_{WH} , do

(a) Let v_j^i be a customer randomly chosen among those closest to the depot

(b) Using the customer vertex sequence $(v_j^i, v_{j+1}^i, ..., v_{n_{CUS}^i}^i, v_1^i, ..., v_{j-1}^i)$, construct a tour on all vertices assigned to warehouse i by means of GENI procedure and Unstringing and Stringing (US) procedure Gendreau et al. (1992).

(c) Start with warehouse i (vertex v_0^i), create m_i vehicle routes by following the tour. The first vehicle contains all customers starting from the first customer on the tour and up to, but excluding, the first customer v whose inclusion in the route would cause a violation of the capacity or route duration limit. This process is repeated, starting from the city v, and until all customers have been included into routes. The initial MDVRP solution is $S_{MDVRP} = \{S_{SDVRP}^i \forall i \in V_{WH}\}$.

3.3 Phase II: Solution Improvement

The initial solution generated in Phase I is used as an input in Phase II, which consists of 3 sub-phases: fast improvement, intensification, and diversification. Three basic procedures that are employed in these sub-phases are first described including one-route, two-route and three-route procedures, followed by the descriptions of three sub-phases. Then, the selection of routes for two-route and three route procedures in the three sub-phases is described.

One-Route Procedure

The one-route procedure is a post-optimizer on single-vehicle routes. In this study, the US algorithm by Gendreau et al. (1992) is employed while maintaining route duration feasibility and vehicle capacity feasibility. Since the procedure improves the sequence of customers on a particular route without reassigning any customer to different warehouses, ED_i and VD_i are unaffected. Thus, the optimal order quantity and reorder point as well as ordering and holding costs are not changed.

Two-Route Procedure

The two-route procedure moves vertices belonging to two different routes assigned to one or two warehouses. Let $(v_{h_1}, v_{j_1}, v_{k_1}, v_{l_1})$ and $(v_{h_2}, v_{j_2}, v_{k_2}, v_{l_2})$ be two sequences of four consecutive vertices (possibly including a warehouse) from route r_1 based at warehouse i_1 and route r_2 based at warehouse i_2, respectively. Similar to Renaud et al. (1996), the following 6 moves are attempted as long as a warehouse is not moved, and vehicle capacity feasibility and route duration feasibility are maintained. The six moves are described together with the calculation of changes in relevant q_r^i, rl_r^i, pl_r^i, ED_i, and VD_i.

(a)Insert v_{j_1} between v_{h_2} and v_{j_2}

The two vertex sequences become $(v_{h_1}, v_{k_1}, v_{l_1})$ and $(v_{h_2}, v_{j_1}, v_{j_2}, v_{k_2}, v_{l_2})$, respectively. The changes in the round-trip lengths are $\Delta_{rl_1} = -c_{h_1,j_1} - c_{j_1,k_1} + c_{h_1,k_1} - \delta_{j_1}$ and $\Delta_{rl_2} = -c_{h_2,j_2} + c_{h_2,j_1} + c_{j_1,j_2} + \delta_{j_1}$. If v_{j_1} is the last customer visited on route r_1, $\Delta_{pl_1} = -c_{h_1,j_1} - \delta_{j_1}$; otherwise, $\Delta_{pl_1} = \Delta_{rl_1}$. If v_{h_2} is the last customer visited on route r_2, $\Delta_{pl_2} = c_{h_2,j_1} + \delta_{j_1}$; otherwise, $\Delta_{pl_2} = \Delta_{rl_2}$. The changes in the average delivery demands are $\Delta q_{r_1}^{i_1} = -\mu_{j_1}$ and $\Delta q_{r_2}^{i_2} = \mu_{j_1}$. If $i_1 \neq i_2$, $\Delta ED_{i_1} = -\mu_{j_1}$, $\Delta ED_{i_2} = \mu_{j_1}$,

$\Delta VD_{i_1} = -\sigma_{j_1}^2$ and $\Delta VD_{i_2} = \sigma_{j_1}^2$. Otherwise, $\Delta ED_{i_1} = \Delta ED_{i_2} = 0$ and $\Delta VD_{i_1} = \Delta VD_{i_2} = 0$.

(b)Insert v_{j_2} between v_{h_1} and v_{j_1}

The two vertex sequences become $(v_{h_1}, v_{j_2}, v_{j_1}, v_{k_1}, v_{l_1})$ and $(v_{h_2}, v_{k_2}, v_{l_2})$, respectively. The changes in the round-trip lengths are $\Delta_{rl_1} = -c_{h_1,j_1} + c_{h_1,j_2} + c_{j_2,j_1} + \delta_{j_2}$ and $\Delta_{rl_2} = -c_{h_2,j_2} - c_{j_2,k_2} + c_{h_2,k_2} - \delta_{j_2}$. If v_{h_1} is the last customer visited on route *r1*, $\Delta_{pl_1} = c_{h_1,j_2} + \delta_{j_2}$; otherwise, $\Delta_{pl_1} = \Delta_{rl_1}$. If v_{j_2} is the last customer visited on route *r2*, $\Delta_{pl_2} = -c_{h_2,j_2} - \delta_{j_2}$; otherwise, $\Delta_{pl_2} = \Delta_{rl_2}$. The changes in the average delivery demands are $\Delta q_{r_1}^{i_1} = \mu_{j_2}$ and $\Delta q_{r_2}^{i_2} = -\mu_{j_2}$. If $i_1 \neq i_2$, then $\Delta ED_{i_1} = \mu_{j_2}$, $\Delta ED_{i_2} = -\mu_{j_2}$, $\Delta VD_{i_1} = \sigma_{j_2}^2$ and $\Delta VD_{i_2} = -\sigma_{j_2}^2$. Otherwise, $\Delta ED_{i_1} = \Delta ED_{i_2} = 0$ and $\Delta VD_{i_1} = \Delta VD_{i_2} = 0$.

(c) Swap v_{j_1} and v_{j_2}

The two vertex sequences become $(v_{h_1}, v_{j_2}, v_{k_1}, v_{l_1})$ and $(v_{h_2}, v_{j_1}, v_{k_2}, v_{l_2})$. The changes in the round-trip lengths are $\Delta_{rl_1} = -c_{h_1,j_1} - c_{j_1,k_1} + c_{h_1,j_2} + c_{j_2,k_1} - \delta_{j_1} + \delta_{j_2}$ and $\Delta_{rl_2} = -c_{h_2,j_2} - c_{j_2,k_2} + c_{h_2,j_1} + c_{j_1,k_2} - \delta_{j_2} + \delta_{j_1}$. If v_{j_1} is the last customer visited on route *r1*, $\Delta_{pl_1} = -c_{h_1,j_1} + c_{h_1,j_2} - \delta_{j_1} + \delta_{j_2}$; otherwise, $\Delta_{pl_1} = \Delta_{rl_1}$. If v_{j_2} is the last customer visited on route *r2*, $\Delta_{pl_2} = -c_{h_2,j_2} + c_{h_2,j_1} - \delta_{j_2} + \delta_{j_1}$; otherwise, $\Delta_{pl_2} = \Delta_{rl_2}$. The changes in the average delivery demands are $\Delta q_{r_1}^{i_1} = -\mu_{j_1} + \mu_{j_2}$ and $\Delta q_{r_2}^{i_2} = \mu_{j_1} - \mu_{j_2}$. If $i_1 \neq i_2$, then $\Delta ED_{i_1} = -\mu_{j_1} + \mu_{j_2}$, $\Delta ED_{i_2} = \mu_{j_1} - \mu_{j_2}$, $\Delta VD_{i_1} = -\sigma_{j_1}^2 + \sigma_{j_2}^2$ and $\Delta VD_{i_2} = \sigma_{j_1}^2 - \sigma_{j_2}^2$. Otherwise, $\Delta ED_{i_1} = \Delta ED_{i_2} = 0$ and $\Delta VD_{i_1} = \Delta VD_{i_2} = 0$.

(d)Insert (v_{j_1}, v_{k_1}) between (v_{h_2}, v_{j_2})

The two vertex sequences become (v_{h_1}, v_{l_1}) and $(v_{h_2}, v_{j_1}, v_{k_1}, v_{j_2}, v_{k_2}, v_{l_2})$. The changes in the round-trip lengths are $\Delta_{rl_1} = -c_{h_1,j_1} - c_{j_1,k_1} - c_{k_1,l_1} + c_{h_1,l_1} - \delta_{j_1} - \delta_{k_1}$ and

$\Delta_{rl_2} = -c_{h_2,j_2} + c_{h_2,j_1} + c_{j_1,k_1} + c_{k_1,j_2} + \delta_{j_1} + \delta_{k_1}$. If v_{k_1} is the last customer visited on route r_1, $\Delta_{pl_1} = -c_{h_1,j_1} - c_{j_1,k_1} - \delta_{j_1} - \delta_{k_1}$; otherwise, $\Delta_{pl_1} = \Delta_{rl_1}$. If v_{h_2} is the last customer visited on route r_2, $\Delta_{pl_2} = c_{h_2,j_1} + c_{j_1,k_1} + \delta_{j_1} + \delta_{k_1}$; otherwise, $\Delta_{pl_2} = \Delta_{rl_2}$. The changes in the average delivery demands are $\Delta q_{r_1}^{i_1} = -\mu_{j_1} - \mu_{k_1}$ and $\Delta q_{r_2}^{i_2} = \mu_{j_1} + \mu_{k_1}$. If $i_1 \neq i_2$, $\Delta ED_{i_1} = -\mu_{j_1} - \mu_{k_1}$, $\Delta ED_{i_2} = \mu_{j_1} + \mu_{k_1}$, $\Delta VD_{i_1} = -\sigma_{j_1}^2 - \sigma_{k_1}^2$ and $\Delta VD_{i_2} = \sigma_{j_1}^2 + \sigma_{k_1}^2$. Otherwise, $\Delta ED_{i_1} = \Delta ED_{i_2} = 0$ and $\Delta VD_{i_1} = \Delta VD_{i_2} = 0$.

(e)Insert (v_{j_2}, v_{k_2}) between (v_{h_1}, v_{j_1})

The two vertex sequences become $(v_{h_1}, v_{j_2}, v_{k_2}, v_{j_1}, v_{k_1}, v_{l_1})$ and (v_{h_2}, v_{l_2}). The changes in the round-trip lengths are $\Delta_{rl_1} = -c_{h_1,j_1} + c_{h_1,j_2} + c_{j_2,k_2} + c_{k_2,j_1} + \delta_{j_2} + \delta_{k_2}$ and $\Delta_{rl_2} = -c_{h_2,j_2} - c_{j_2,k_2} - c_{k_2,l_2} + c_{h_2,l_2} - \delta_{j_2} - \delta_{k_2}$. If v_{h_1} is the last customer visited on route r_1, $\Delta_{pl_1} = c_{h_1,j_2} + c_{j_2,k_2} + \delta_{j_2} + \delta_{k_2}$; otherwise, $\Delta_{pl_1} = \Delta_{rl_1}$. If v_{k_2} is the last customer visited on route r_2, $\Delta_{pl_2} = -c_{h_2,j_2} - c_{j_2,k_2} - \delta_{j_2} - \delta_{k_2}$; otherwise, $\Delta_{pl_2} = \Delta_{rl_2}$. The changes in the average delivery demands are $\Delta q_{r_1}^{i_1} = \mu_{j_2} + \mu_{k_2}$ and $\Delta q_{r_2}^{i_2} = -\mu_{j_2} - \mu_{k_2}$. If $i_1 \neq i_2$, $\Delta ED_{i_1} = \mu_{j_2} + \mu_{k_2}$, $\Delta ED_{i_2} = -\mu_{j_2} - \mu_{k_2}$, $\Delta VD_{i_1} = \sigma_{j_2}^2 + \sigma_{k_2}^2$ and $\Delta VD_{i_2} = -\sigma_{j_2}^2 - \sigma_{k_2}^2$. Otherwise, $\Delta ED_{i_1} = \Delta ED_{i_2} = 0$ and $\Delta VD_{i_1} = \Delta VD_{i_2} = 0$.

(f)Swap (v_{j_1}, v_{k_1}) and (v_{j_2}, v_{k_2})

The two vertex sequences become $(v_{h_1}, v_{j_2}, v_{k_2}, v_{l_1})$ and $(v_{h_2}, v_{j_1}, v_{k_1}, v_{l_2})$. The changes in the round-trip lengths are

$\Delta_{rl_1} = -c_{h_1,j_1} - c_{j_1,k_1} - c_{k_1,l_1} + c_{h_1,j_2} + c_{j_2,k_2} + c_{k_2,l_1} - \delta_{j_1} - \delta_{k_1} + \delta_{j_2} + \delta_{k_2}$ and

$\Delta_{rl_2} = -c_{h_2,j_2} - c_{j_2,k_2} - c_{k_2,l_2} + c_{h_2,j_1} + c_{j_1,k_1} + c_{k_1,l_2} - \delta_{j_2} - \delta_{k_2} + \delta_{j_1} + \delta_{k_1}$. If v_{k_1} is the last customer visited on route r_1, $\Delta_{pl_1} = -c_{h_1,j_1} - c_{j_1,k_1} + c_{h_1,j_2} + c_{j_2,k_2} - \delta_{j_1} - \delta_{k_1} + \delta_{j_2} + \delta_{k_2}$; otherwise, $\Delta_{pl_1} = \Delta_{rl_1}$. If v_{k_2} is the last customer visited on route r_2, $\Delta_{pl_2} = c_{h_2,j_1} + c_{j_1,k_1} - c_{h_2,j_2} - c_{j_2,k_2} + \delta_{j_1} + \delta_{k_1} - \delta_{j_2} - \delta_{k_2}$; otherwise, $\Delta_{pl_2} = \Delta_{rl_2}$. The changes in the average delivery demands are $\Delta q_{r_1}^{i_1} = -\mu_{j_1} - \mu_{k_1} + \mu_{j_2} + \mu_{k_2}$ and $\Delta q_{r_2}^{i_2} = \mu_{j_1} + \mu_{k_1} - \mu_{j_2} - \mu_{k_2}$. If $i_1 \neq i_2$, $\Delta ED_{i_1} = -\mu_{j_1} - \mu_{k_1} + \mu_{j_2} + \mu_{k_2}$,

$\Delta ED_{i_2} = \mu_{j_1} + \mu_{k_1} - \mu_{j_2} - \mu_{k_2}$, $\Delta VD_{i_1} = -\sigma_{j_1}^2 - \sigma_{k_1}^2 + \sigma_{j_2}^2 + \sigma_{k_2}^2$ and

$\Delta VD_{i_2} = \sigma_{j_1}^2 + \sigma_{k_1}^2 - \sigma_{j_2}^2 - \sigma_{k_2}^2$. Otherwise, $\Delta ED_{i_1} = \Delta ED_{i_2} = 0$ and $\Delta VD_{i_1} = \Delta VD_{i_2} = 0$. These six moves are a subset of the larger family of moves considered within the λ-interchange procedure by Osman (1993), and Renaud et al. (1996) indicated that very little quality is lost but much time is gained by concentrating on this restricted subset of six moves in multi-depot vehicle routing problem.

Three-Route Procedure

The three-route procedure is an exchange scheme involving three routes (Renaud et al., 1996). Let $(v_{h_1-1}, v_{h_1}, v_{h_1+1})$, $(v_{h_2-1}, v_{h_2}, v_{h_2+1}, ..., v_{j_2}, v_{k_2})$ and (v_{j_3}, v_{k_3}) be three sequences of consecutive vertices (possibly including a warehouse) from routes r_1, r_2 and r_3 with at least 3, 4 and 3 vertices respectively, based at warehouses i_1, i_2 and i_3. For routes r_2 and r_3, consider the sequences of two vertices (v_{j_2}, v_{k_2}) and (v_{j_3}, v_{k_3}) where $v_{j_2} \neq v_{h_2}$ and $v_{k_2} \neq v_{h_2}$. Then the following combination of moves is attempted as long as vehicle capacity feasibility and route duration feasibility are maintained, and a warehouse is not moved: insert v_{h_1} between v_{j_2} and v_{k_2} , and insert v_{h_2} between v_{j_3} and v_{k_3} . The move is described together with the calculation of changes in relevant q_r^i , rl_r^i , pl_r^i , ED_i , and VD_i . After three-route exchange, the three vertex sequences become (v_{h_1-1}, v_{h_1+1}) , $(v_{h_2-1}, v_{h_2+1}, ..., v_{j_2}, v_{h_1}, v_{k_2})$ and $(v_{j_3}, v_{h_2}, v_{k_3})$. The changes in the round-trip lengths are $\Delta_{rl_1} = -c_{h_1-1,h_1} - c_{h_1,h_1+1} + c_{h_1-1,h_1+1} - \delta_{h_1}$,

$\Delta_{rl_2} = -c_{h_2-1,h_2} - c_{h_2,h_2+1} + c_{h_2-1,h_2+1} - \delta_{h_2} - c_{j_2,k_2} + c_{j_2,h_1} + c_{h_1,k_2} + \delta_{h_1}$ and

$\Delta_{rl_3} = -c_{j_3,k_3} + c_{j_3,h_2} + c_{h_2,k_3} + \delta_{h_2}$. If v_{h_1} is the last customer visited on route r_1,

$\Delta_{pl_1} = -c_{h_1-1,h_1} - \delta_{h_1}$; otherwise, $\Delta_{pl_1} = \Delta_{rl_1}$. If v_{h_2} is the last customer visited on route r_2, $\Delta_{pl_2} = -c_{h_2-1,h_2} - \delta_{h_2} - c_{j_2,k_2} + c_{j_2,h_1} + c_{h_1,k_2} + \delta_{h_1}$. If v_{j_2} is the last customer visited on route r_2,

$\Delta_{pl_2} = -c_{h_2-1,h_2} - c_{h_2,h_2+1} + c_{h_2-1,h_2+1} - \delta_{h_2} + c_{j_2,h_1} + \delta_{h_1}$. Otherwise, $\Delta_{pl_2} = \Delta_{rl_2}$. If v_{j_3} is the last customer visited on route r_3, $\Delta_{pl_3} = c_{j_3,h_2} + \delta_{h_2}$. Otherwise, $\Delta_{pl_3} = \Delta_{rl_3}$. The changes in the average delivery demands are $\Delta q_{r_1}^{i_1} = -\mu_{h_1}$, $\Delta q_{r_2}^{i_2} = \mu_{h_1} - \mu_{h_2}$ and

$\Delta q_{r_3}^{i_3} = \mu_{h_2}$. If $i_1 \neq i_2 \neq i_3$, then

$\Delta ED_{i_1} = -\mu_{h_1}$, $\Delta ED_{i_2} = \mu_{h_1} - \mu_{h_2}$, $\Delta ED_{i_3} = \mu_{h_2}$, $\Delta VD_{i_1} = -\sigma^2_{h_1}$, $\Delta VD_{i_2} = \sigma^2_{h_1} - \sigma^2_{h_2}$ and $\Delta VD_{i_3} = \sigma^2_{h_2}$. If $i_1 = i_2 = i_3$, then $\Delta ED_{i_1} = \Delta ED_{i_2} = 0$ and $\Delta VD_{i_1} = \Delta VD_{i_2} = 0$. If $i_1 = i_2 \neq i_3$, $\Delta ED_{i_1} = \Delta ED_{i_2} = -\mu_{h_2}$, $\Delta ED_{i_3} = \mu_{h_2}$, $\Delta VD_{i_1} = \Delta VD_{i_2} = -\sigma^2_{h_2}$ and $\Delta VD_{i_3} = \sigma^2_{h_2}$.

If $i_1 = i_3 \neq i_2$, $\Delta ED_{i_1} = \Delta ED_{i_3} = -\mu_{h_1} + \mu_{h_2}$, $\Delta ED_{i_2} = \mu_{h_1} - \mu_{h_2}$, $\Delta VD_{i_1} = \Delta VD_{i_3} = -\sigma^2_{h_1} + \sigma^2_{h_2}$ and $\Delta VD_{i_2} = \sigma^2_{h_1} - \sigma^2_{h_2}$. If $i_1 \neq i_2 = i_3$, $\Delta ED_{i_1} = -\mu_{h_1}$, $\Delta ED_{i_2} = \Delta ED_{i_3} = \mu_{h_1}$, $\Delta VD_{i_1} = -\sigma^2_{h_1}$ and $\Delta VD_{i_2} = \Delta VD_{i_3} = \sigma^2_{h_1}$.

Sub-Phase II.1: Fast Improvement

The algorithm attempts to improve upon the incumbent by repeatedly applying the following three steps:

- Inter-warehouse: Apply two-route procedure between routes of two different warehouses.
- Intra-warehouse: Apply two-route procedure between routes of the same warehouse.
- Three-Route: Exchange vertices between three routes, using three-route procedure.

These steps are repeated until the incumbent does not improve for $n^{fast}_{\max}$ consecutive iterations. For each of the three steps, any move that yields an improvement is immediately implemented. Otherwise, the best non-tabu deteriorating move is implemented. Whenever a move is implemented, the one-route procedure is applied to all routes involved in the move.

Sub-Phase II.2: Intensification

This phase intensifies the search for better route, starting with the best known solution and working on one warehouse at the time. It applies the intra-warehouse step to each warehouse in turn until no improvement to the incumbent has been produced for $n^{\text{intens}}_{\max}$ consecutive iterations. Whenever a move is implemented, the one-route procedure is applied to all routes involved in the move.

Sub-Phase II.3: Diversification

The effect of the diversification phase is to perform a broader exploration of the solution space. The following two steps are repeated 20 times.

- First, we seek the best reinsertion of a vertex from its current route into a route belonging to a different warehouse; that is, apply the first move type of the two-route procedure limiting to only two routes associated with different warehouses. Choosing the same vertex for reinsertion is prohibited for the next 10 applications of this step. Whenever a move is implemented, the one-route procedure is applied to all routes involved in the move.
- Second, the inter-warehouse and intra-warehouse steps of the fast improvement sub-phase are applied for $n_{\max}^{FastDiver}$ consecutive iterations without improvement to the solution values obtained in the first step. Here the length of the interval during which a move is tabu is randomly chosen in [15,20] and no aspiration criterion is used. Whenever a move is implemented, the one-route procedure is applied to all routes involved in the move.

Selection of Routes for Two-Route and Three-Route Procedures in the Three Sub-Phases

The selection of routes to which two-route and three-route procedures are applied is described Renaud et al. (1996). To define the distance between a route and a warehouse or between two routes, each route is represented by its center of gravity. In inter-warehouse, we consider exchanges between each warehouse i and the $\left\lfloor \frac{n_{WH}}{2} \right\rfloor + 1$ warehouses closest to it. For each pair of warehouses i_1 and i_2, we consider exchanges between the $\left\lceil \frac{m_{i_1}}{2} \right\rceil$ routes of warehouse i_1 closest to warehouse i_2 and the $\left\lceil \frac{m_{i_2}}{2} \right\rceil$ routes of warehouse i_2 closest to warehouse i_1. In intra-warehouse, we consider all pairs of routes for each warehouse. In three-route procedure, the three routes r_1, r_2, and r_3 are selected as follows. All routes with at least 3 vertices are considered for route r_1. Route r_2 is the closest neighbor of route r_1 and has at least 4 vertices. Route r_3 is the closest neighbor of route r_2 with $r_3 \neq r_1$, and route r_3 has at least 3 vertices.

Throughout Phase II, the incumbent and its value are recorded. The current solution is not necessarily the best known because the deteriorations of the objective function are allowed. Whenever a customer is moved from its current route, moving this customer back into the same route is declared tabu for θ iterations, where θ is randomly chosen in $\left[\theta_{\min}^{FIND}, \theta_{\max}^{FIND}\right] = [4,10]$. Random tabu durations help avoid cycling. A tabu status may be overridden if implementing the corresponding move yields a better incumbent.

Chapter 4 Experimental Results

The tabu search heuristics are implemented in C++. These run on a computer with 1.73 GHz Intel Core i7 processor and 4 GB of RAM, running under Windows 7. The data are first described. Then, the computational results of two experiments are discussed.

4.1 Data

For IRP, there is not the standard set of instances for testing algorithms. We generated instances similar to the types used in VRP. The customer locations are generated from Solomon (1987)'s VRP with time windows instances. The Solomon instances are divided into six groups, denoted R1, R2, C1, C2, RC1 and RC2. In R1 and R2, the customer locations are randomly generated from a uniform distribution, and in C1 and C2, they are clustered. In RC1 and RC2, the customer locations are a combination of randomly generated and clustered points. Because the (x,y)-coordinates of the customer locations are the same for R1 and R2 and for RC1 and RC2, the Solomon instances yield only four sets of distinct customer locations: C1, C2, R1, RC1. In the same manner as Berger et al. (2007), we create five instances corresponding to each group of customer locations. The first instance includes the first 50 customers, the second instance the last 50 customers, the third instance the first 75 customers, the fourth instance the last 75 customers, and the fifth instance all 100 customers. These are denoted by 50a, 50b, 75a, 75b and 100, respectively. Thus, there are 20 instances of customer locations. The service times are set at 10 time units for all customers. The average demands of customers are equal to the demands used in Solomon (1987). The demand variances are based on the coefficients of variance randomly generated from the range [0.45, 0.55]. The customer data for C1-100, C2-100, R1-100 and RC1-100 are shown in Tables A1-A4, respectively, in Appendix A.

For the warehouse locations, we created two sets of 4 warehouse locations for each customer instance. The first and second sets of candidate warehouse locations are denoted by wh1 and wh2, respectively. We randomly generated the warehouse locations from a uniform distribution, so that two criteria are satisfied. First, each customer location could be reached by a singleton route with the associated route duration to the last customer of at most 80 time units (M=80) from at least one warehouse. Second, each warehouse location must be assigned at least 10, 15 and 20 customers for the 50, 75 and 100 customer instances, respectively, when assigning customers to their nearest warehouse. For all warehouse instances, homogeneous unit holding costs of the four warehouses are $0.3, $0.6, $0.9, $1.2 per product unit per day; the homogeneous ordering costs $450, $900, $1350 and $1800 per order. For all warehouses, the lead times are two days; inventory capacity 2000 product units; order quantity capacity 2000 product units; unit transport cost from the plant to warehouses is zero. The distance matrix is determined based on Euclidean distance between all vertex pair. The traveling speed is assumed 1 distance unit per time unit, and routing cost is assumed $1 per travel time unit to cover variable vehicle costs. Personnel costs and other vehicle related fixed costs are assumed to be considered outside the inventory-routing decision. The route duration limits are 100 time units. The number of available vehicles for each warehouse is unlimited with the homogeneous capacity of 100 product units, which are less constrained than the route duration limit constraints in all test problems. We identify each instance by an ID. The first part of the ID specifies the problem group (R1, C1, C2 or RC1). The second part specifies the customer subset (50a, 50b, 75a, 75b or 100). The third part specifies the set of warehouse locations (wh1 or wh2). Thus, there are 40 problem instances. The warehouse locations for the 40 problem instances are shown in Tables B1-B4 in Appendix B.

4.2 Computational Results

We calibrate the two tabu search algorithms by varying $n_{\max}^{fast}$, $n_{\max}^{\text{intens}}$ and $n_{\max}^{FastDiver}$ on a test problem, and found that the algorithm parameters suggested by Renaud et al. (1996) perform best ($n_{\max}^{fast}$=75, $n_{\max}^{\text{intens}}$ =300 and $n_{\max}^{FastDiver}$=50). We conduct two experiments. The first experiment compares the performances of the type-1 and type-2 tabu search heuristics in terms of computational time and solution quality against the sequential approach. The sequential approach first solves MDVRP with route duration limits, whose routing solutions are input to the continuous ICP with stochastic inventory capacity constraints and order quantity capacity constraints. In the second experiment, the sensitivity analysis is performed on problem instance RC1-100-wh1 by varying the route duration limit (M=80 and 100), order quantity capacity ($Q_{0i}^{\max}$=800, 1000 and 2000) and demand variance (-30%, 0% and +30% changes). The demand variances are the product of the original demand variance and demand variance factor (DVarF); thus, -30%, 0% and +30% changes in demand variances correspond to DVarF values of 0.7, 1.0 and 1.3, respectively.

Table 4.1 shows the best objective values found and total computational time by type-1 and type-2 tabu search algorithms and the sequential approach on the 40 problem instances.

Table 4.1. Computational Results of Sequential ICP and SDVRP, Sequential MDVRP and ICP, and Combined ICP and MDVRP

	Sequential MDVRP and ICP		Combined MDVRP and ICP					
			Init. Sol. Type 1			Init. Sol. Type 2		
	Best Obj. ($/day)	CPU Time (min)	Best Obj. ($/day)	% Improve.	CPU Time (min)	Best Obj. ($/day)	% Improve.	CPU Time (min)
50-Customers Problems								
C1-50a-wh1	3,727.02	1.36	**3,599.25**	**3.43%**	1.96	3,619.87	2.87%	1.67
C1-50a-wh2	3,821.86	1.26	**3,727.73**	**2.46%**	0.89	3,747.72	1.94%	0.79
C1-50b-wh1	4,083.87	0.66	**3,728.71**	**8.70%**	0.61	3,795.24	7.07%	0.78
C1-50b-wh2	3,999.61	0.74	**3,478.46**	**13.03%**	0.85	3,823.94	4.39%	0.71
C2-50a-wh1	3,840.98	0.98	3,550.59	7.56%	0.88	**3,528.95**	**8.12%**	1.02
C2-50a-wh2	3,514.08	0.94	**3,223.65**	**8.26%**	0.95	3,369.86	4.10%	0.96
C2-50b-wh1	4,211.79	0.79	3,956.97	6.05%	0.53	**3,905.52**	**7.27%**	0.62
C2-50b-wh2	4,108.18	0.58	3,627.97	11.69%	0.69	**3,588.85**	**12.64%**	0.64
R1-50a-wh1	3,570.89	0.76	3,367.28	5.70%	0.65	**3,235.19**	**9.40%**	0.83
R1-50a-wh2	3,693.61	0.74	3,584.06	2.97%	0.55	**3,485.92**	**5.62%**	0.71
R1-50b-wh1	3,715.19	0.84	3,284.29	11.60%	0.80	**3,190.68**	**14.12%**	0.98
R1-50b-wh2	3,927.35	1.21	3,723.42	5.19%	0.65	**3,588.96**	**8.62%**	0.76
RC1-50a-wh1	4,228.72	0.61	4,085.40	3.39%	0.53	**4,029.67**	**4.71%**	0.62
RC1-50a-wh2	4,414.56	1.12	3,844.28	12.92%	0.68	**3,814.29**	**13.60%**	0.53
RC1-50b-wh1	3,707.06	0.64	**3,270.56**	**11.77%**	0.74	3,284.03	11.41%	0.82
RC1-50b-wh2	3,623.14	0.74	3,320.46	8.35%	0.31	**3,203.86**	**11.57%**	0.63
75-Customers Problems								
C1-75a-wh1	4,966.38	1.80	**4,839.85**	**2.55%**	1.76	4,885.40	1.63%	1.49
C1-75a-wh2	5,152.99	1.35	5,152.99	0.00%	1.06	**5,014.38**	**2.69%**	0.99
C1-75b-wh1	5,545.76	1.09	5,393.16	2.75%	1.17	**5,366.82**	**3.23%**	1.23
C1-75b-wh2	5,375.33	1.03	5,249.51	2.34%	1.02	**5,100.32**	**5.12%**	0.78
C2-75a-wh1	5,306.20	1.18	**4,991.70**	**5.93%**	1.05	5,022.92	5.34%	1.22
C2-75a-wh2	4,985.98	1.26	**4,695.13**	**5.83%**	1.19	4,834.15	3.05%	1.25
C2-75b-wh1	5,528.38	1.22	**5,311.50**	**3.92%**	0.96	5,330.93	3.57%	1.17
C2-75b-wh2	5,338.21	1.28	5,210.52	2.39%	1.06	**5,138.19**	**3.75%**	1.01
R1-75a-wh1	4,703.12	0.95	4,632.99	1.49%	1.02	**4,535.88**	**3.56%**	1.29
R1-75a-wh2	4,593.40	0.93	**4,412.45**	**3.94%**	0.88	4,497.09	2.10%	1.14
R1-75b-wh1	4,788.27	1.34	**4,378.67**	**8.55%**	1.07	4,523.30	5.53%	1.11
R1-75b-wh2	4,763.60	1.81	4,536.31	4.77%	1.19	**4,490.63**	**5.73%**	1.35
RC1-75a-wh1	4,988.65	1.05	4,827.74	3.23%	0.61	**4,815.07**	**3.48%**	0.99
RC1-75a-wh2	5,236.32	1.07	**5,104.13**	**2.52%**	0.91	5,236.32	0.00%	0.88
RC1-75b-wh1	4,971.99	1.00	4,785.77	3.75%	0.68	**4,760.34**	**4.26%**	0.82
RC1-75b-wh2	4,987.55	1.03	**4,745.55**	**4.85%**	0.87	4,750.22	4.76%	1.02
100-Customers Problems								
C1-100-wh1	6,061.05	1.71	5,815.44	4.05%	1.90	**5,761.22**	**4.95%**	1.75
C1-100-wh2	6,214.90	1.64	6,110.84	1.67%	1.42	**6,005.50**	**3.37%**	1.62
C2-100-wh1	6,142.82	1.50	5,872.85	4.39%	1.33	**5,872.55**	**4.40%**	1.25
C2-100-wh2	6,590.24	1.41	6,507.64	1.25%	1.28	**6,380.97**	**3.18%**	1.47
R1-100-wh1	5,616.66	1.24	5,264.36	6.27%	1.23	**5,227.34**	**6.93%**	1.82
R1-100-wh2	5,574.31	1.97	**5,430.94**	**2.57%**	1.70	5,471.92	1.84%	1.49
RC1-100-wh1	6,112.11	1.43	6,016.59	1.56%	1.36	**5,984.98**	**2.08%**	1.47
RC1-100-wh2	6,324.67	1.32	6,258.55	1.05%	1.15	**6,127.76**	**3.11%**	1.39

As expected, the proposed tabu search algorithms outperform the sequential approach on all test problem instances. The computational times of all runs are less than two minutes. On 50-customer and 75-customer problem instances, the two tabu search algorithms perform approximately equally well, since the type-1 tabu search outperform the type-2 tabu search on about 50 percent of problem instances. On 100-customer instances, the type-2 tabu search outperforms the type-1 tabu search on almost all instances except an instance R1-100-wh2.

Next, the sensitivity analysis is performed to see how the solution changes with route duration limit, order quantity capacity and demand variance on problem instance RC1-100-wh1. We employ the type-2 tabu search in all runs as it performs best on this problem instance. Table 4.2 and Figure 4.1 show the best objective value found when varying route duration limits, order quantity capacity and demand variances.

Table 4.2. Best Objective Values Found by Type-2 Tabu Search with Varying Route Duration Limits, Order Quantity Capacity and Demand Variances

Route Duration Limit; Order Quantity Capacity	Demand Variance Factor		
	0.7	1	1.3
M100; Qmax2000	5902.68	5984.98	6030.5
M100; Qmax1000	5939.6	6003.1	6057.6
M100; Qmax800	6087.56	6194.37	6249.35
M80; Qmax2000	6328.16	6393.26	6449.12
M80; Qmax1000	6345.43	6411.14	6459.52
M80; Qmax800	6536.16	6601.42	6657.43

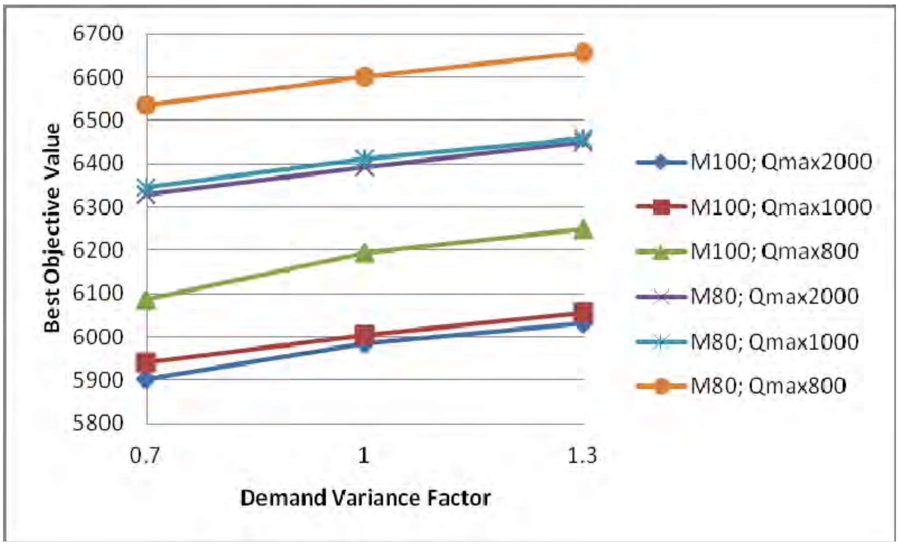


Figure 4.1. Best Objective Values Found by Type-2 Tabu Search with Varying Route Duration Limits, Order Quantity Capacity and Demand Variances

It can be seen that the best objective value increases with the increase of demand variance, but decreases with the increase of order quantity capacity and route duration limit. The best objective value is composed of three cost components: vehicle routing, holding and ordering. Table 4.3 and Figure 4.2 show the three cost components when varying route duration limit and order quantity capacity at DVarF=1.0. Table 4.4 and Figure 4.3 show the three cost components when varying route duration limit and demand variance at $Q_i^{\max}$=2000. As can be seen from Table 4.3 and Figure 4.2, the total ordering costs in the best solution increases with the decrease of the order quantity capacity, whereas the total holding costs decreases with the decrease of the order quantity capacity. Evidently, the order quantity capacity as well as inventory capacity plays a role in the trade-off between total holding costs and total ordering costs. Intuitively, when the order quantity is more constrained, the warehouse manager has to order more often and ordering costs are higher. Meanwhile, the peak inventory levels are lower and the total holding costs are less. Furthermore, Table 4.3 and Figure 4.2 show that the routing costs increases with the decrease of route duration limit. Once the longer route duration limit is allowed, each vehicle route may serve more customers, and the routing costs is less. Table 4.4 and Figure 4.3 show that the holding costs increase with the increase of demand variance, but it is unclear how the routing and ordering costs change with the demand variances.

This is as expected as the demand variance is only directly related to the holding costs as shown in Eq.(2.6.1). The demand variance can influence the customer assignments to different warehouses, resulting in different routing costs and ordering costs.

Table 4.3. Routing Costs, Holding Costs and Ordering Costs with Varying Route Duration Limits and Order Quantity Capacity (DVarF=1.0)

Route Duration Limit; Order Quantity Capacity	Demand Variance Factor = 1.0		
	Ordering Costs	Holding Costs	Routing Costs
M100; Qmax2000	1624.25	2013.05	2347.67
M100; Qmax1000	1817.97	1850.37	2334.77
M100; Qmax800	2266.31	1592.2	2335.85
M80; Qmax2000	1673.52	2072.03	2647.71
M80; Qmax1000	1907.1	1902.27	2601.77
M80; Qmax800	2365.88	1599.55	2636

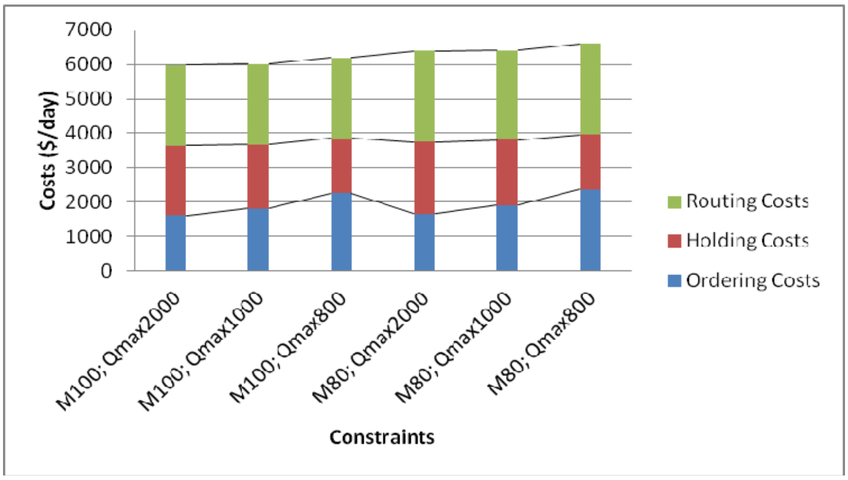


Figure 4.2. Routing Costs, Holding Costs and Ordering Costs with Varying Route Duration Limits and Order Quantity Capacity (DVarF=1.0)

Table 4.4. Routing Costs, Holding Costs and Ordering Costs with Varying Demand Variances

a) Order Quantity Capacity=2000 and Route Duration Limit = 100

Route Duration Limit; Order Quantity Capacity; Demand Variance Factor	Ordering Costs	Holding Costs	Routing Costs
M100; Qmax2000; DVarF=0.7	1619.26	1943.43	2339.99
M100; Qmax2000; DVarF=1.0	1624.25	2013.05	2347.67
M100; Qmax2000; DVarF=1.3	1619.26	2061.02	2350.21

b) Order Quantity Capacity=800 and Route Duration Limit = 100

Route Duration Limit; Order Quantity Capacity; Demand Variance Factor	Ordering Costs	Holding Costs	Routing Costs
M100; Qmax800; DVarF=0.7	2227.5	1525.29	2334.77
M100; Qmax800; DVarF=1.0	2266.31	1592.2	2335.85
M100; Qmax800; DVarF=1.3	2266.31	1647.18	2335.85

a)Order Quantity Capacity=2000 and Route Duration Limit = 100

b) Order Quantity Capacity=800 and Route Duration Limit = 100

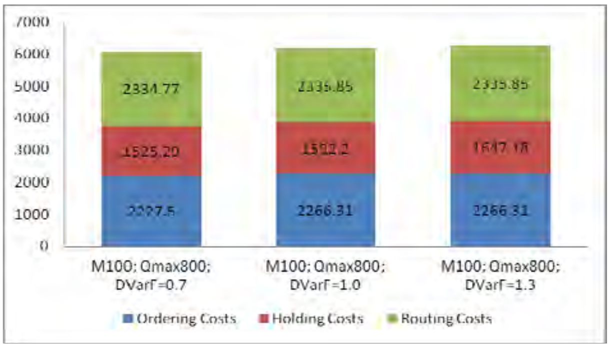
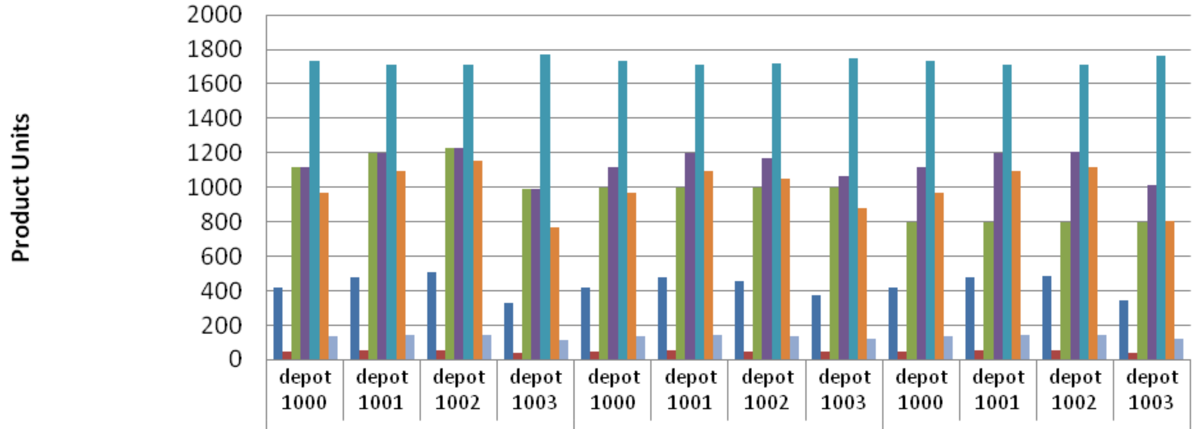


Figure 4.3. Routing Costs, Holding Costs and Ordering Costs with Varying Demand Variances

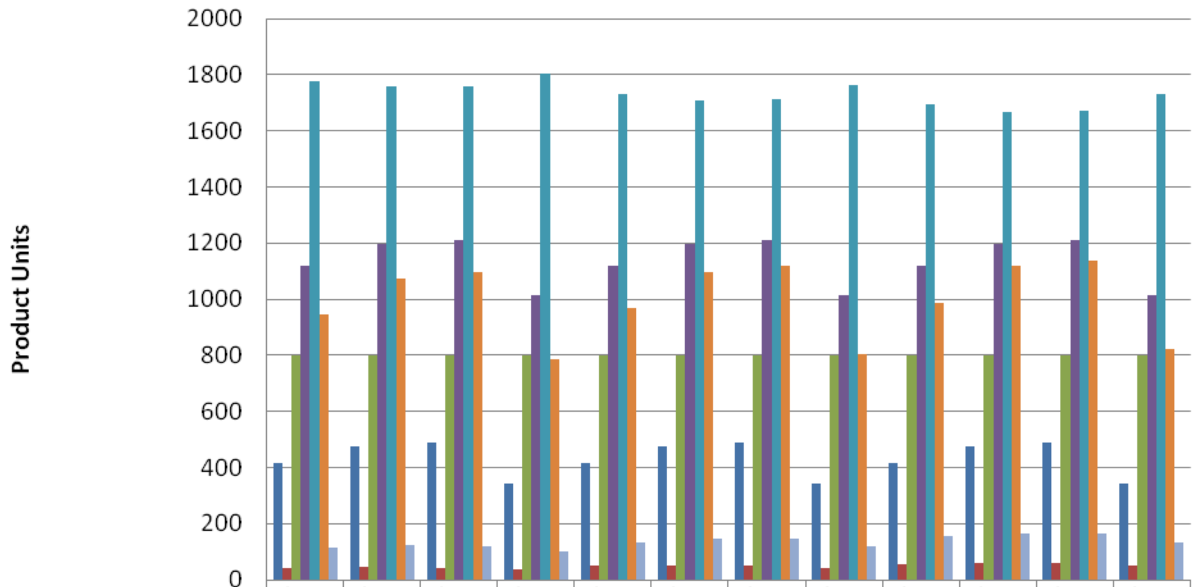
Figure 4.4 shows the continuous inventory control policies at four warehouses in the best solution when varying order quantity capacity at $L_{0i}^{\max}$=80 and DVarF=1.0. When the order quantity capacity ($Q_i^{\max}$=2000) is equal to the inventory capacity, the optimal order quantity is equal to the EOQ formula according to Eq.(3.3). When the order quantity capacity decreases to 1000 and 800, the customers as well as associated mean demands are reassigned between warehouses 1002 and 1003. As such the reorder points and safety stocks of warehouses 1000 and 1001 are unaffected with the change of order quantity capacity, but those of warehouses 1002 and 1003 are affected. The optimal order quantities for the case $Q_i^{\max} = 800$ and $Q_i^{\max} = 1000$ are equal to $Q_i^{\max}$ according to Eq.(3.3).



	Qmax2000				Qmax1000				Qmax800			
	depot 1000	depot 1001	depot 1002	depot 1003	depot 1000	depot 1001	depot 1002	depot 1003	depot 1000	depot 1001	depot 1002	depot 1003
Mean Demand (unit/day)	417	476	505	326	417	476	455	376	417	476	487	344
SD of Demand (unit/day)	49	53	53	42	49	53	50	45	49	53	52	43
Optimal Order Quantity	1118	1195	1231	989	1000	1000	1000	1000	800	800	800	800
EOQ	1118	1195	1231	989	1118	1195	1168	1062	1118	1195	1209	1016
Avail Icap	1731	1709	1708	1768	1731	1709	1722	1751	1731	1709	1711	1764
Reorder Point	969	1098	1156	768	969	1098	1049	877	969	1098	1118	806
Safety Stock	135	146	146	116	135	146	139	125	135	146	144	118

Figure 4.4. Continuous Inventory Control Policies at Four Warehouses with Varying Order Quantity Capacity (Route Duration Limit=80; DVarF=1.0)

Figure 4.5 shows the continuous inventory control policies at four warehouses in the best solution when varying demand variance at $L_{0i}^{\max}=80$ and $Q_i^{\max}=800$. The customers as well as associated mean demands assigned to the four warehouses are unaffected with the change of demand variance. The safety stock levels and reorder points at the four warehouses increase with the increase of demand variance, whereas the available inventory capacities at the four warehouses decrease with the increase of demand variance. This is intuitive as the safety stock is positively related to demand variance, and the reorder point includes the safety stock as shown in Eq.(2.2). The available inventory capacity is negatively related to demand variance (available inventory capacity = $I_i^{\max} - (Z_{1-\alpha} + Z_{1-\beta})\sqrt{LT_i}\sqrt{VD_i}$). The optimal order quantities are equal to the order quantity capacity according to Eq.(3.3). Table 4.5 shows the MDVRP policies for the four warehouses when $L_{0i}^{\max}$ =80 and $L_{0i}^{\max}$ =100. The number of routes is decreases with the increase of route duration limits. This is because the available vehicle capacity in each route is large enough to serve additional customers. As can be noticed in Table 4.5, each route has the travel time to last customer less than or equal to the route duration limit, and the mean demand of each route is less than the vehicle capacity.



	DVarF=0.7				DVarF=1.0				DVarF=1.3			
	depot 1000	depot 1001	depot 1002	depot 1003	depot 1000	depot 1001	depot 1002	depot 1003	depot 1000	depot 1001	depot 1002	depot 1003
Mean Demand (unit/day)	417	476	487	344	417	476	487	344	417	476	487	344
SD of Demand (unit/day)	41	44	44	36	49	53	52	43	55	60	59	49
Optimal Order Quantity	800	800	800	800	800	800	800	800	800	800	800	800
EOQ	1118	1195	1209	1016	1118	1195	1209	1016	1118	1195	1209	1016
Avail Icap	1775	1756	1758	1802	1731	1709	1711	1764	1693	1668	1671	1730
Reorder Point	947	1074	1095	787	969	1098	1118	806	987	1118	1139	823
Safety Stock	113	122	121	99	135	146	144	118	153	166	165	135

Figure 4.5. Continuous Inventory Control Policies at Four Warehouses with Varying Demand Variances (Route Duration Limit=80; Order Quantity Capacity=800)

Table 4.5. Multi-Depot Vehicle Routing Policies for Four Warehouses with Varying Route Duration Limits (Order Quantity Capacity =2000 and DVarF=1.0)

	M80			
	depot 1000	depot 1001	depot 1002	depot 1003
No. of Routes	6	8	8	8
Routes	1000-98-55-69-82	1001-6-7-79-8	1002-65-90-96-94	1003-67-93-71
	1000-88-60-78-73	1001-46-4-45-5-3	1002-95-92-91-80	1003-72-54-81
	1000-14-47-17-16-15	1001-42-44	1002-66-56-84-64	1003-62
	1000-59-97-75	1001-1-43-40	1002-83-57-24-22	1003-51-85-63
	1000-9-13-87	1001-36-35-37	1002-20-49-19-18	1003-76-89
	1000-10-11-12-53	1001-38-39-41	1002-48-21-23-25	1003-33-32-30-28-26
		1001-70-61-68	1002-77-58	1003-27-29-31-34
		1001-100-2	1002-74-86-52-99	1003-50
Mean Demands	61; 68; 80; 70; 43; 95	80; 90; 20; 70; 70; 60; 53; 33	71; 46; 70; 87; 80; 70; 27; 54	26; 34; 3; 27; 56; 70; 80; 30
Travel Times to Last Customer	75.46; 77.83; 78.00;	73.52; 64.66; 55.24; 74.49;	76.91; 77.24; 76.56; 69.41;	57.10; 79.49; 15.83; 61.36;
	76.68; 62.90; 72.08	79.54; 77.58; 62.61; 44.77	72.52; 78.78; 65.74; 78.36	52.26; 79.94; 75.18; 13.61
Travel Times (begin and end at depot)	81.78; 102.04; 97.65;	81.59; 73.66; 88.78; 113.54;	101.98; 93.52; 88.23; 90.99;	80.64; 112.91; 21.66; 74.36;
	119.76; 85.75; 77.18	122.95; 113.60; 86.69; 45.77	104.80; 109.78; 98.55; 87.42	79.43; 105.02; 90.48; 17.21

	M100			
	depot 1000	depot 1001	depot 1002	depot 1003
No. of Routes	6	6	6	6
Routes	1000-98-69-90-65-82	1001-2-6-7-8-46	1002-91-92-94-96-80	1003-67-93-71
	1000-53-88-60-79-78	1001-4-45-5-3-1	1002-64-84-95-56-66	1003-85-62
	1000-12-47-17-16-15-13	1001-42-44-43-40-39	1002-83-22-24-57	1003-51-76-89-63
	1000-97-75-59	1001-36-35-37-38	1002-20-49-19-18-48-21	1003-33-32-34
	1000-99-86-74-87-9	1001-81-54-72-41	1002-23-25-77-58	1003-31-29-27-26-28-30
	1000-73-14-11-10	1001-70-61-68-55-100	1002-52	1003-50
Mean Demands	67; 98; 100; 70; 84; 85	90; 100; 80; 100; 54; 72	89; 76; 87; 100; 77; 3	26; 5; 81; 50; 100; 30
Travel Times to last customer	87.24; 97.47; 93.83;	66.00; 64.47; 98.24;	92.71; 97.31; 76.47;	57.10; 43.57; 93.68;
	90.76; 97.51; 87.34	93.14; 99.39; 99.68	96.52; 96.94; 21.66	57.32; 97.25; 13.61
Travel Times (begin and end at depot)	93.56; 114.50; 112.70;	71.66; 73.96; 136.45;	108.99; 103.39; 88.55;	80.64; 49.40; 106.68;
	118.07; 112.38; 99.42	133.34; 135.40; 112.41	127.92; 129.74; 33.32	72.62; 115.36; 17.21

Chapter 5 Summary, Conclusions and Future Research

This research studies a two-level supply chain where a single plant supplies a single commodity to a set of warehouses which in turn serve a set of customers with stochastic demands. This research provides a combined stochastic chance constrained nonlinear integer programming formulation modeling the inventory management decisions at the warehouses and the routing of goods from the warehouses to the customers. The warehouses are assumed to manage the inventory using a continuous inventory policy. The model accounts for the service level at each warehouse which reflects the probability of available inventory meeting the demand during the lead time, probability of violation of inventory capacity, and restrictions on order quantity volume. The routing of goods from warehouse to customers is modeled as a route duration constrained capacitated multi-depot vehicle routing problem. Two tabu search heuristics – type 1 and type 2, differing primarily in the way initial solutions are generated are developed to solve the combined model. The optimal order quantity at each warehouse is approximated using the KKT conditions.

Computational runs are conducted on variations of the standard Solomon test instances available for vehicle routing problems with time windows. Type 2 tabu search was found to outperform type 1 tabu search for the 100 customer instance. For smaller customer instances, both the heuristics were found to perform equally well. Integrating the inventory management and routing decisions by solving the combined inventory management and routing problem was found to yield cost savings of up to 14% over the sequential approach where both problems are solved separately.

The best objective function value obtained by the tabu search heuristic was found to increase with increase in customer demand variance, decrease with increase in order quantity capacity and route duration limit. Variance of the customer demand was found to have significant impact on the solution quality. The safety stock levels, the reorder points and the total holding costs were found to increase with increase in customer demand variance. As expected, the available inventory capacity was found to decrease with increase in customer demand variance. It is unclear how the routing and ordering costs change with the demand variances. This is because the demand variance can influence the customer assignments to different warehouses, resulting in different routing costs and ordering costs. We found that the order quantity capacity and inventory capacity play a role in the trade-off between total holding costs and total ordering costs. The total ordering costs in the best solution increases with the decrease of the order quantity capacity, whereas the total holding costs decreases with the decrease of the order quantity capacity. The routing costs increase with the decrease of route duration limit. Thus, the combined inventory management and routing model can be used to study the tradeoffs between inventory holding costs, ordering costs, and routing costs.

This research can be extended in multiple directions. The immediate next step is to integrate warehouse facility location problem into the combined inventory management and routing model.

APPENDIX A: Customer Data

Table A1. Customer Data for Test Problem C1

No	x	y	Service Time	Mean Demand	Demand Variance	No	x	y	Service Time	Mean Demand	Demand Variance
1	45	68	10	10	23.89	26	25	55	10	10	29.28
2	45	70	10	30	208.21	27	23	52	10	10	27.51
3	42	66	10	10	25.79	28	23	55	10	20	113.51
4	42	68	10	10	26.12	29	20	50	10	10	22.02
5	42	65	10	10	25.40	30	20	55	10	10	27.36
6	40	69	10	20	116.36	31	10	35	10	20	98.89
7	40	66	10	20	119.04	32	10	40	10	30	205.09
8	38	68	10	20	89.28	33	8	40	10	40	389.72
9	38	70	10	10	21.88	34	8	45	10	20	114.96
10	35	66	10	10	20.76	35	5	35	10	10	21.47
11	35	69	10	10	27.29	36	5	45	10	10	23.09
12	25	85	10	20	105.87	37	2	40	10	20	91.40
13	22	75	10	30	271.78	38	0	40	10	30	236.99
14	22	85	10	10	26.84	39	0	45	10	20	91.27
15	20	80	10	40	473.91	40	35	30	10	10	27.85
16	20	85	10	40	480.16	41	35	32	10	10	21.22
17	18	75	10	20	85.74	42	33	32	10	20	85.88
18	15	75	10	20	96.22	43	33	35	10	10	27.76
19	15	80	10	10	21.45	44	32	30	10	10	27.21
20	30	50	10	10	22.60	45	30	30	10	10	20.87
21	30	52	10	20	120.47	46	30	32	10	30	183.66
22	28	52	10	20	102.68	47	30	35	10	10	24.88
23	28	55	10	10	21.77	48	28	30	10	10	28.14
24	25	50	10	10	28.53	49	28	35	10	10	22.43
25	25	52	10	40	468.31	50	26	32	10	10	20.38

Table A1. Customer Data for Test Problem C1 (Continued)

No	x	y	Service Time	Mean Demand	Demand Variance
51	25	30	10	10	22.76
52	25	35	10	10	30.19
53	44	5	10	20	93.77
54	42	10	10	40	467.78
55	42	15	10	10	22.75
56	40	5	10	30	199.65
57	40	15	10	40	389.95
58	38	5	10	30	219.18
59	38	15	10	10	28.52
60	35	5	10	20	86.43
61	50	30	10	10	23.27
62	50	35	10	20	94.10
63	50	40	10	50	520.50
64	48	30	10	10	23.64
65	48	40	10	10	24.62
66	47	35	10	10	20.40
67	47	40	10	10	21.07
68	45	30	10	10	25.61
69	45	35	10	10	21.01
70	95	30	10	30	230.62
71	95	35	10	20	98.61
72	53	30	10	10	28.84
73	92	30	10	10	25.69
74	53	35	10	50	534.26
75	45	65	10	20	102.65

No	x	y	Service Time	Mean Demand	Demand Variance
76	90	35	10	10	27.31
77	88	30	10	10	25.93
78	88	35	10	20	85.06
79	87	30	10	10	24.83
80	85	25	10	10	20.69
81	85	35	10	30	271.79
82	75	55	10	20	93.05
83	72	55	10	10	29.86
84	70	58	10	20	90.26
85	68	60	10	30	211.05
86	66	55	10	10	27.20
87	65	55	10	20	97.58
88	65	60	10	30	197.78
89	63	58	10	10	29.07
90	60	55	10	10	23.16
91	60	60	10	10	20.57
92	67	85	10	20	102.76
93	65	85	10	40	398.67
94	65	82	10	10	21.74
95	62	80	10	30	220.45
96	60	80	10	10	21.76
97	60	85	10	30	229.54
98	58	75	10	20	83.39
99	55	80	10	10	29.73
100	55	85	10	20	84.91

Table A2. Customer Data for Test Problem C2

No	x	y	Service Time	Mean Demand	Demand Variance	No	x	y	Service Time	Mean Demand	Demand Variance
1	52	75	10	10	21.09	26	8	62	10	10	20.38
2	45	70	10	30	237.69	27	23	52	10	10	29.42
3	62	69	10	10	27.18	28	4	55	10	20	94.88
4	60	66	10	10	24.63	29	20	50	10	10	24.64
5	42	65	10	10	25.56	30	20	55	10	10	21.55
6	16	42	10	20	85.37	31	10	35	10	20	103.10
7	58	70	10	20	94.75	32	10	40	10	30	268.15
8	34	60	10	20	92.42	33	8	40	10	40	333.34
9	28	70	10	10	28.19	34	8	45	10	20	110.83
10	35	66	10	10	21.73	35	5	35	10	10	24.31
11	35	69	10	10	25.45	36	5	45	10	10	20.32
12	25	85	10	20	93.10	37	2	40	10	20	110.00
13	22	75	10	30	187.07	38	0	40	10	30	184.39
14	22	85	10	10	23.68	39	0	45	10	20	98.24
15	20	80	10	40	387.23	40	36	18	10	10	28.34
16	20	85	10	40	335.03	41	35	32	10	10	24.98
17	18	75	10	20	83.61	42	33	32	10	20	115.04
18	15	75	10	20	86.02	43	33	35	10	10	24.34
19	15	80	10	10	25.98	44	32	20	10	10	20.74
20	30	50	10	10	23.51	45	30	30	10	10	20.47
21	30	56	10	20	106.98	46	34	25	10	30	240.94
22	28	52	10	20	120.07	47	30	35	10	10	21.22
23	14	66	10	10	26.43	48	36	40	10	10	27.16
24	25	50	10	10	22.43	49	48	20	10	10	29.41
25	22	66	10	40	463.58	50	26	32	10	10	20.94

Table A2. Customer Data for Test Problem C2 (Continued)

No	x	y	Service Time	Mean Demand	Demand Variance
51	25	30	10	10	27.00
52	25	35	10	10	25.13
53	44	5	10	20	93.80
54	42	10	10	40	438.08
55	42	15	10	10	20.96
56	40	5	10	30	238.24
57	38	15	10	40	471.20
58	38	5	10	30	259.84
59	38	10	10	10	29.01
60	35	5	10	20	108.91
61	50	30	10	10	21.30
62	50	35	10	20	81.04
63	50	40	10	50	555.76
64	48	30	10	10	23.85
65	44	25	10	10	21.69
66	47	35	10	10	29.84
67	47	40	10	10	24.79
68	42	30	10	10	22.68
69	45	35	10	10	25.21
70	95	30	10	30	236.35
71	95	35	10	20	94.66
72	53	30	10	10	26.86
73	92	30	10	10	20.61
74	53	35	10	50	539.30
75	45	65	10	20	117.35
76	90	35	10	10	20.47
77	72	45	10	10	20.39
78	78	40	10	20	103.12
79	87	30	10	10	28.01
80	85	25	10	10	26.53
81	85	35	10	30	185.14
82	75	55	10	20	120.81
83	72	55	10	10	23.47
84	70	58	10	20	96.64
85	86	46	10	30	233.96
86	66	55	10	10	25.41
87	64	46	10	20	90.23
88	65	60	10	30	232.82
89	56	64	10	10	28.91
90	60	55	10	10	24.60
91	60	60	10	10	21.88
92	67	85	10	20	85.50
93	42	58	10	40	454.05
94	65	82	10	10	28.60
95	62	80	10	30	209.66
96	62	40	10	10	27.69
97	60	85	10	30	219.15
98	58	75	10	20	96.59
99	55	80	10	10	23.91
100	55	85	10	20	96.42

Table A3. Customer Data for Test Problem R1

No	x	y	Service Time	Mean Demand	Demand Variance	No	x	y	Service Time	Mean Demand	Demand Variance
1	41	49	10	10	23.69	26	45	30	10	17	60.20
2	35	17	10	7	13.82	27	35	40	10	16	67.54
3	55	45	10	13	40.99	28	41	37	10	16	52.17
4	55	20	10	19	74.25	29	64	42	10	9	17.46
5	15	30	10	26	201.42	30	40	60	10	21	90.58
6	25	30	10	3	2.20	31	31	52	10	27	206.49
7	20	50	10	5	5.12	32	35	69	10	23	130.94
8	10	43	10	9	16.66	33	53	52	10	11	29.53
9	55	60	10	16	62.70	34	65	55	10	14	40.20
10	30	60	10	16	60.62	35	63	65	10	8	17.26
11	20	65	10	12	34.39	36	2	60	10	5	5.75
12	50	35	10	19	96.60	37	20	20	10	8	16.53
13	30	25	10	23	118.13	38	5	5	10	16	63.45
14	15	10	10	20	106.09	39	60	12	10	31	208.85
15	30	5	10	8	18.07	40	40	25	10	9	22.02
16	10	20	10	19	104.45	41	42	7	10	5	7.23
17	5	30	10	2	0.98	42	24	12	10	5	6.29
18	20	40	10	12	34.01	43	23	3	10	7	11.22
19	15	60	10	17	65.32	44	11	14	10	18	95.57
20	45	65	10	9	23.76	45	6	38	10	16	65.61
21	45	20	10	11	34.50	46	2	48	10	1	0.27
22	45	10	10	18	97.68	47	8	56	10	27	208.38
23	55	5	10	29	200.13	48	13	52	10	36	272.63
24	65	35	10	3	2.16	49	6	68	10	30	257.56
25	65	20	10	6	10.43	50	47	47	10	13	47.35

Table A3. Customer Data for Test Problem R1 (Continued)

No	x	y	Service Time	Mean Demand	Demand Variance
51	49	58	10	10	24.22
52	27	43	10	9	17.45
53	37	31	10	14	57.45
54	57	29	10	18	72.48
55	63	23	10	2	0.86
56	53	12	10	6	8.51
57	32	12	10	7	13.43
58	36	26	10	18	69.42
59	21	24	10	28	215.24
60	17	34	10	3	2.54
61	12	24	10	13	38.12
62	24	58	10	19	101.27
63	27	69	10	10	20.45
64	15	77	10	9	23.90
65	62	77	10	20	103.50
66	49	73	10	25	173.11
67	67	5	10	25	142.39
68	56	39	10	36	323.61
69	37	47	10	6	8.74
70	37	56	10	5	7.16
71	57	68	10	15	62.41
72	47	16	10	25	127.88
73	44	17	10	9	18.03
74	46	13	10	8	18.94
75	49	11	10	18	89.38

No	x	y	Service Time	Mean Demand	Demand Variance
76	49	42	10	13	39.75
77	53	43	10	14	43.11
78	61	52	10	3	2.49
79	57	48	10	23	158.76
80	56	37	10	6	9.95
81	55	54	10	26	182.44
82	15	47	10	16	57.02
83	14	37	10	11	32.81
84	11	31	10	7	11.91
85	16	22	10	41	347.89
86	4	18	10	35	327.76
87	28	18	10	26	163.37
88	26	52	10	9	21.71
89	26	35	10	15	59.41
90	31	67	10	3	1.92
91	15	19	10	1	0.26
92	22	22	10	2	1.19
93	18	24	10	22	129.86
94	26	27	10	27	190.10
95	25	24	10	20	102.09
96	22	27	10	11	34.27
97	25	21	10	12	35.56
98	19	21	10	10	22.27
99	20	26	10	9	19.34
100	18	18	10	17	72.62

Table A4. Customer Data for Test Problem RC1

No	x	y	Service Time	Mean Demand	Demand Variance
1	25	85	10	20	119.17
2	22	75	10	30	211.39
3	22	85	10	10	26.40
4	20	80	10	40	383.91
5	20	85	10	20	118.24
6	18	75	10	20	110.12
7	15	75	10	20	88.65
8	15	80	10	10	23.54
9	10	35	10	20	85.86
10	10	40	10	30	193.03
11	8	40	10	40	348.20
12	8	45	10	20	99.34
13	5	35	10	10	23.92
14	5	45	10	10	28.26
15	2	40	10	20	84.52
16	0	40	10	20	98.61
17	0	45	10	20	109.11
18	44	5	10	20	104.76
19	42	10	10	40	460.90
20	42	15	10	10	22.61
21	40	5	10	10	22.08
22	40	15	10	40	407.67
23	38	5	10	30	209.74
24	38	15	10	10	22.91
25	35	5	10	20	112.18
26	95	30	10	30	229.69
27	95	35	10	20	82.13
28	92	30	10	10	23.80
29	90	35	10	10	22.11
30	88	30	10	10	26.30
31	88	35	10	20	93.80
32	87	30	10	10	25.22
33	85	25	10	10	28.46
34	85	35	10	30	251.74
35	67	85	10	20	110.59
36	65	85	10	40	335.42
37	65	82	10	10	21.88
38	62	80	10	30	182.91
39	60	80	10	10	25.71
40	60	85	10	30	192.42
41	58	75	10	20	120.76
42	55	80	10	10	21.84
43	55	85	10	20	94.81
44	55	82	10	10	23.45
45	20	82	10	10	23.26
46	18	80	10	10	21.81
47	2	45	10	10	26.22
48	42	5	10	10	24.91
49	42	12	10	10	26.84
50	72	35	10	30	222.54

Table A4. Customer Data for Test Problem RC1 (Continued)

No	x	y	Service Time	Mean Demand	Demand Variance	No	x	y	Service Time	Mean Demand	Demand Variance
51	55	20	10	19	90.90	76	60	12	10	31	279.89
52	25	30	10	3	2.40	77	23	3	10	7	11.95
53	20	50	10	5	5.53	78	8	56	10	27	216.47
54	55	60	10	16	65.85	79	6	68	10	30	210.17
55	30	60	10	16	69.27	80	47	47	10	13	50.26
56	50	35	10	19	85.23	81	49	58	10	10	28.62
57	30	25	10	23	107.39	82	27	43	10	9	23.98
58	15	10	10	20	85.21	83	37	31	10	14	48.15
59	10	20	10	19	101.73	84	57	29	10	18	66.70
60	15	60	10	17	85.55	85	63	23	10	2	0.86
61	45	65	10	9	22.70	86	21	24	10	28	213.91
62	65	35	10	3	2.14	87	12	24	10	13	39.19
63	65	20	10	6	10.58	88	24	58	10	19	86.34
64	45	30	10	17	61.25	89	67	5	10	25	175.88
65	35	40	10	16	74.73	90	37	47	10	6	8.16
66	41	37	10	16	66.18	91	49	42	10	13	35.91
67	64	42	10	9	17.50	92	53	43	10	14	48.08
68	40	60	10	21	111.88	93	61	52	10	3	2.39
69	31	52	10	27	207.31	94	57	48	10	23	137.42
70	35	69	10	23	153.42	95	56	37	10	6	8.92
71	65	55	10	14	55.13	96	55	54	10	26	190.91
72	63	65	10	8	16.26	97	4	18	10	35	339.80
73	2	60	10	5	6.26	98	26	52	10	9	22.46
74	20	20	10	8	19.13	99	26	35	10	15	46.14
75	5	5	10	16	56.66	100	31	67	10	3	2.29

APPENDIX B: Warehouse Data

Table B1. Warehouse Data for Test Problem C1

a) C1-50a-wh1

Vertice	x	y
1000	41	59
1001	9	48
1002	41	51
1003	16	63

b)C1-50a-wh2

Vertice	x	y
1000	36	32
1001	36	43
1002	40	44
1003	12	41

c)C1-50b-wh1

Vertice	x	y
1000	50	63
1001	63	14
1002	48	52
1003	81	55

d)C1-50b-wh2

Vertice	x	y
1000	38	8
1001	73	53
1002	67	70
1003	64	48

e) C1-75a-wh1

Vertice	x	y
1000	13	56
1001	32	81
1002	38	44
1003	38	17

f)C1-75a-wh2

Vertice	x	y
1000	54	57
1001	38	6
1002	47	82
1003	58	45

g)C1-75b-wh1

Vertice	x	y
1000	88	24
1001	94	51
1002	10	57
1003	15	5

h)C1-75b-wh2

Vertice	x	y
1000	65	29
1001	40	10
1002	91	59
1003	14	75

i) C1-100-wh1

Vertice	x	y
1000	25	28
1001	34	72
1002	42	43
1003	54	22

j)C1-100-wh2

Vertice	x	y
1000	12	10
1001	74	46
1002	57	81
1003	6	75

Table B2. Warehouse Data for Test Problem C2

a) C2-50a-wh1

Vertice	x	y
1000	14	52
1001	14	78
1002	23	45
1003	46	41

b)C2-50a-wh2

Vertice	x	y
1000	27	41
1001	3	58
1002	48	71
1003	32	38

c)C2-50b-wh1

Vertice	x	y
1000	30	57
1001	84	10
1002	81	57
1003	73	69

d)C2-50b-wh2

Vertice	x	y
1000	74	10
1001	55	40
1002	48	51
1003	48	42

e) C2-75a-wh1

Vertice	x	y
1000	27	49
1001	37	69
1002	68	36
1003	9	33

f)C2-75a-wh2

Vertice	x	y
1000	36	5
1001	22	75
1002	56	17
1003	1	47

g)C2-75b-wh1

Vertice	x	y
1000	67	9
1001	17	7
1002	91	53
1003	32	71

h)C2-75b-wh2

Vertice	x	y
1000	65	29
1001	40	10
1002	91	59
1003	14	75

i) C2-100-wh1

Vertice	x	y
1000	48	53
1001	18	27
1002	62	58
1003	55	33

j)C2-100-wh2

Vertice	x	y
1000	43	73
1001	12	6
1002	20	70
1003	75	67

Table B3. Warehouse Data for Test Problem R1

a) R1-50a-wh1

Vertice	x	y
1000	43	19
1001	48	59
1002	12	52
1003	29	29

b)R1-50a-wh2

Vertice	x	y
1000	53	52
1001	12	59
1002	46	36
1003	21	3

c)R1-50b-wh1

Vertice	x	y
1000	38	37
1001	29	26
1002	33	41
1003	13	28

d)R1-50b-wh2

Vertice	x	y
1000	39	7
1001	59	9
1002	22	75
1003	8	6

e) R1-75a-wh1

Vertice	x	y
1000	54	10
1001	19	31
1002	24	38
1003	36	73

f)R1-75a-wh2

Vertice	x	y
1000	47	25
1001	29	8
1002	64	53
1003	11	67

g)R1-75b-wh1

Vertice	x	y
1000	8	33
1001	28	60
1002	37	30
1003	45	40

h)R1-75b-wh2

Vertice	x	y
1000	55	46
1001	14	16
1002	55	38
1003	12	40

i) R1-100-wh1

Vertice	x	y
1000	8	33
1001	28	60
1002	37	30
1003	45	40

j)R1-100-wh2

Vertice	x	y
1000	47	25
1001	29	8
1002	64	53
1003	11	67

Table B4. Warehouse Data for Test Problem RC1

a) RC1-50a-wh1

Vertice	x	y
1000	21	49
1001	12	17
1002	86	30
1003	4	55

b)RC1-50a-wh2

Vertice	x	y
1000	76	32
1001	21	56
1002	11	27
1003	14	57

c)RC1-50b-wh1

Vertice	x	y
1000	3	51
1001	63	28
1002	55	50
1003	27	62

d)RC1-50b-wh2

Vertice	x	y
1000	33	52
1001	38	25
1002	47	66
1003	28	20

e) RC1-75a-wh1

Vertice	x	y
1000	2	50
1001	49	56
1002	68	35
1003	36	7

f)RC1-75a-wh2

Vertice	x	y
1000	0	35
1001	50	50
1002	57	53
1003	15	58

g)RC1-75b-wh1

Vertice	x	y
1000	10	37
1001	39	67
1002	52	33
1003	63	44

h)RC1-75b-wh2

Vertice	x	y
1000	43	31
1001	7	52
1002	74	68
1003	50	27

i) RC1-100-wh1

Vertice	x	y
1000	21	45
1001	22	76
1002	35	36
1003	70	32

j)RC1-100-wh2

Vertice	x	y
1000	62	29
1001	9	44
1002	21	26
1003	87	48

REFERENCES

1. Anily, S., A. Federgruen. (1990) One warehouse multiple retailer systems with vehicle routing. Management Science, Vol.36, 92-114.
2. Anily, S., A. Federgruen. (1993) Two-echelon distribution systems with vehicle routing costs and central inventories. Operations Research, Vol. 41, 37–47.
3. Bard, J., Huang, L., Jaillet, P., M. Dror, M. (1998) A decomposition approach to the inventory routing problem with satellite facilities. Transportation Science, Vol.32, 189–203.
4. Beltrami, F., L. Bodin (1974). Networks and vehicle routing for municipal waste collection, Networks, Vol. 4, 65–94.
5. Berger, R.T., C.R. Coullard, and M.S. Daskin. (2007) Location-Routing Problems with Distance Constraints. Transportation Science, Vol.41, No.1, 29-43.
6. Bertazzi, L., Paletta, G., S. M. Grazia. (2002) Deterministic order-up-to level policies in an inventory routing problem. Transportation Science, Vol. 36 (11), 119–132.
7. Bertazzi, L., Speranza, M.G., M. W. P. Savelsbergh. (2008), Inventory routing, in:Vehicle routing: Latest advances and new challenges, Golden, B., Raghavan, R., E. Wasil (editors.), Springer, 49-72.
8. Bramel, J., D. Simchi-Levi. (1995) A location based heuristic for general routing problems. Operations Research, Vol.43, 649-660.
9. Campbell, A., M. Savelsbergh. (2004) A decomposition approach for the inventory–routing problem. Transportation Science, Vol.38 (4), 488–502.
10. Campbell, A.M. and M. Savelsbergh. (2004) Efficient Insertion Heuristics for Vehicle Routing and Scheduling Problems. Transportation Science, Vol.38, No.3, 369-378.
11. Campbell, A.M., Clarke, L., Kleywegt, A., M. W. P. Savelsbergh. (1998), The inventory routing problem, in: Fleet Management and Logistics, Crainic, T.G., G. Laporte. (editors), 95-113, Kluwer, Boston.
12. Cordeau, J-F., M. Gendreau, G. Laporte (1997) A tabu search heuristic for periodic and multi-depot vehicle routing problems, Networks, Vol.30, No.2, 105-119.

13. Daskin, M.S., C. Coullard and Z-J. M. Shen. (2002) An Inventory-Location Model: Formulation, Solution Algorithm and Computational Results, Annals of Operations Research, Vol.110, 83-106.
14. Dror, M., Ball, M., B. Golden. (1985) Computational comparison of algorithms for the inventory routing problem. Annals of Operations Reseach, Vol.4, 3–23.
15. Dror, M., M. O. Ball. (1987) Inventory/routing: Reduction from an annual to a short-period problem. Naval Research of Logistics Quarterly, Vol.34, 891–908.
16. Erlenkotter, D. (1990) Ford Whitman Harris and the Economic Order Quantity Model. Operations Research, Vol.38, 937-946.
17. Federgruen, A., D. Simchi–Levi. (1995), Analysis of vehicle routing and inventory-routing problems, in: Handbooks in Operations Research and Management Science, Ball, M.O., Magnanti, T.L., Monma, C.L., G .L. Nemhauser. (editors), Vol. 8, 297-373, North-Holland.
18. Federgruen, A., P. Zipkin. (1984) A combined vehicle routing and inventory allocation problem. Operations Research Vol. 32, 297–373.
19. Gallego, G., D. Simchi-Levi. (1990) On the effectiveness of direct shipping strategyf or the one-warehousem ulti-retailerR -systems. Management Science, Vol.36 240-243.
20. Gaur, V., M. L. Fisher. (2004) A periodic inventory routing problem at a supermarket chain. Operations Research, Vol.52, 813–822.
21. Gendreau, M., A. Hertz, and G. Laporte (1992) New Insertion and Postoptimization Procedures for the Traveling Salesman Problem. Operations Research, Vol.40, No.6.
22. Gendreau, M., A. Hertz, and G. Laporte (1994) A Tabu Search Heuristic for the Vehicle Routing Problem, *Management Science*, Vol.40, No.10.
23. Glover, F. (1989) “Tabu Search - Part I”, *ORSA Journal on Computing*, Vol. 1, No. 3, pp 190-206.
24. Glover, F. and M. Laguna. (1997) *Tabu Search*. Kluwer Academic Publishers, Norwell, MA, USA.
25. Gorr, W.P., M.P. Johnson, S. Roehrig (2001) Facility location model for home-delivered services: Application to the Meals on Wheels program, Journal of Geographic Systems, Vol.3, 181-197.

26. Jaillet, P.L., Huang, J.B., M. Dror. (2002) Delivery cost approximations for inventory routing problems in a rolling horizon framework. Transportation Science, Vol.36, 292–300.
27. Lenstra, J.K., A.H.G. Rinnooy Kan. (1981) Complexity of vehicle routing and scheduling problems, Networks, Vol.11, 221-227.
28. Luenberger, D. G. (1973) Introduction to Linear and Nonlinear Programming. Addison-Wesley, Reading, Mass.
29. Miranda, P.A. and R.A. Garrido. (2004) Incorporating inventory control decisions into a strategic distribution network design model with stochastic demand. Transportation Research Part E, Vol.40, 183-207.
30. Miranda, P.A. and R.A. Garrido. (2006) A Simultaneous Inventory Control and Facility Location Model with Stochastic Capacity Constraints. Networks and Spatial Economics. Vol.6, 39-53, DOI 10.1007/s11067-006-7684-5.
31. Osman, I.H. (1993) Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. Annals of Operations Research, Vol.41, 421-451.
32. Renaud, J., G. Laporte and F. F. Boctor. (1996) A Tabu Search Heuristic for the Multi-Depot Vehicle Routing Problem, Computers and Operations Research, Vol.23, No.3, 229-235.
33. Savelsbergh, M. V. P., J-H. Song. (2007) Inventory routing with continuous moves, Computers and Operations Research, Vol.34, 1744-1763.
34. Savelsbergh, M. V. P., J-H. Song. (2008) An optimization algorithm for the inventory routing problem with continuous moves, Computers and Operations Research, Vol.35, 2266-2282.
35. Solomon, M. M. (1987) Algorithm for the vehicle routing and scheduling problem with time window constraints. Operations Research, Vol.35, 254-265.

Output จากโครงการวิจัยที่ได้รับทุนจาก สกว.

ผลงานตีพิมพ์ในวารสารวิชาการนานาชาติ

- Karoonsoontawong and D-Y. Lin. Time-Varying Lane-based Capacity Reversibility for Traffic Management, *Computer-Aided Civil and Infrastructure Engineering*, 2011, Vol.26, Issue 8, 632-646.
- Karoonsoontawong. An Enhanced Combined Trip Distribution and Assignment Problem, *Journal of Society for Transportation and Traffic Studies*, December 2011, Vol.2, No.4, 27-38.
- P. Lertworawanich and A. Karoonsoontawong. Service-Life Analysis and Maintenance Program of Pavement Markings in Thailand. *Transportation Research Record: Journal of the Transportation Research Board*, No.2272, Transportation Research Board of the National Academies, Washington, D.C., 2012, pp.121-129.
- Karoonsoontawong and A. Unnikrishnan. Tabu Search Heuristics for Inventory Routing Problem with Route Duration Limits and Stochastic Inventory Capacity Constraints. *Transportation Research Record: Journal of the Transportation Research Board (Under Review)*.

การนำผลงานวิจัยไปใช้ประโยชน์

เชิงวิชาการ มีการพัฒนาการเรียนการสอนโดยใช้เนื้อหาบางส่วนจากงานวิจัยนี้ สำหรับวิชา CVE405 Introduction to Logistics สำหรับนักศึกษาระดับปริญญาตรีชั้นปีที่ 4 และวิชา CET666 Logistics Management สำหรับนักศึกษาระดับบัณฑิตศึกษา

การเสนอผลงานในที่ประชุมวิชาการ

- A. Karoonsoontawong and D-Y. Lin. Algorithm for Combined Trip Distribution and Paired Combinatorial Logit Stochastic User Equilibrium Assignment Problem. *91st Transportation Research Board Meeting*, January 2012.
- A. Karoonsoontawong and A. Unnikrishnan. Tabu search heuristics for inventory routing problem with route duration limits and stochastic inventory capacity constraints, *92nd Transportation Research Board Meeting*, January 13-17, 2013.
- A. Karoonsoontawong and K. Heebkhoksung. A modified wall building-based compound approach for container loading problem, *92nd Transportation Research Board Meeting*, January 13-17, 2013.

ภาคผนวก
ผลงานตีพิมพ์ในวารสารวิชาการนานาชาติ

Computer-Aided Civil and Infrastructure Engineering **26** (2011) 632–646

Time-Varying Lane-Based Capacity Reversibility for Traffic Management

Ampol Karoonsoontawong*

Department of Civil Engineering, King Mongkut's University of Technology Thonburi, Thung-Khru, Bangkok, Thailand

&

Dung-Ying Lin

Department of Transportation and Communication Management Science, National Cheng Kung University, Tainan City, Taiwan

Abstract: *This article presents a new bi-level formulation for time-varying lane-based capacity reversibility problem for traffic management. The problem is formulated as a bi-level program where the lower level is the cell-transmission-based user-optimal dynamic traffic assignment (UODTA). Due to its Non-deterministic Polynomial-time hard (NP-hard) complexity, the genetic algorithm (GA) with the simulation-based UODTA is adopted to solve multiorigin multidestination problems. Four GA variations are proposed. GA1 is a simple GA. GA2, GA3, and GA4 with a jam-density factor parameter (JDF) employ time-dependent congestion measures in their decoding procedures. The four algorithms are empirically tested on a grid network and compared based on solution quality, convergence speed, and central processing unit (CPU) time. GA3 with JDF of 0.6 appears best on the three criteria. On the Sioux Falls network, GA3 with JDF of 0.7 performs best. The GA with the appropriate inclusion of problem-specific knowledge and parameter calibration indeed provides excellent results when compared with the simple GA.*

1 INTRODUCTION

Nowadays, many metropolitan areas have adopted various traffic management techniques (Adeli and Samant, 2000; Samant and Adeli, 2000, 2001; Karim and Adeli, 2002a,b, 2003a,b,c; Ghosh-Dastidar and Adeli, 2003; Adeli and Jiang, 2003; Jiang and Adeli, 2004a,b, 2005; Liu and Danczyk, 2009; Hamad et al., 2009; Mirchandani et al., 2010; Ng et al., 2010; Sun and Kondyli, 2010; Ye and Zhang, 2010) to maintain an efficient flow of traffic. Capacity reversibility strategy (a.k.a. contraflow) is a traffic management method, which essentially accommodates the unbalanced traffic flows between two driving directions on a congested roadway section during daily peak periods (Tuydes, 2005). Over the past years, the literature on contraflow emergency evacuation has considerably increased, especially in the United States, due to natural and man-made disasters. Tuydes (2005), Shen et al. (2007), Kalafatas and Peeta (2009), and Xie et al. (2010) provided comprehensive reviews on this topic. Because the contraflow strategy becomes more widely accepted mainly for the emergency evacuation, the previously described implementation issues on costs, safety, and control should be resolved. Thus, the contraflow for daily traffic management may be reconsidered as a workable option, especially for the urban cities with available contraflow emergency evacuation plans. The practicality assumptions are made in this article, including acceptable cost, viable safety policy, sufficient manpower, signage, and potential of blockage actions, which are required to divide a multilane roadway segment.

This article proposes a time-varying lane-based capacity reversibility (TVLCR) model based on the

*To whom correspondence should be addressed. E-mail: *ampol.kar@kmutt.ac.th*.

DOI: 10.1111/j.1467-8667.2011.00722.x

user-optimal dynamic traffic assignment (UODTA) for peak-period traffic management on a daily basis. The model embeds a traffic flow theoretical model, namely, the cell transmission model (CTM, Daganzo, 1994) that can capture traffic realisms such as shockwaves and spillovers. It is noted that the CTM-based formulations for traffic network design problem can be found in Karoonsoontawong and Waller (2005, 2006, 2007, 2010), Kalafatas and Peeta (2009), and Ukkusuri and Waller (2007). Because the proposed model is Non-deterministic Polynomial-time hard (NP-hard), a global optimization method that employs the simulation-based UODTA is suitable for this problem because it can overcome the local-optimum issue. A simple genetic algorithm (GA) and three variations of problem-specific knowledge-based GAs are proposed, and the performances of these GA algorithms are compared on a test problem. Then, the best performing algorithm and the simple GA are applied to a larger size problem.

2 LITERATURE REVIEW

Due to the space limitation, only the most relevant literature review is described here. The proposed formulation is developed on the basis of the works by Tuydes (2005) and Tuydes and Ziliaskopoulos (2004, 2006). Tuydes and Ziliaskopoulos (2004) formulated the system-optimal dynamic traffic assignment (SODTA)-based capacity reversibility problem as a linear program (denoted by SODTA-CR), which propagates traffic based on the CTM to better represent vehicle-level movements, to capture spatiotemporal changes in disaster conditions, and to enable optimal capacity reversibility calculation. SODTA-CR has a major drawback on the continuous capacity redistribution variables that allow an unrealistic fraction-of-lane solution. Tuydes (2005) proposed three extensions of SODTA-CR: lane-based capacity reversibility (SODTA-LCR), total-or-nothing capacity reversibility (SODTA-TCR), and budgeted capacity reversibility (SODTA-BCR). SODTA-LCR addresses the drawback of SODTA-CR by using integer redistribution variables (i.e., lane-based reversibility). The deficiencies of lane-based capacity-reversibility models (including our proposed formulation) are on the cost of the street divisions and the risk in assigning contradicting flows on the same highway. The SODTA-TCR was developed to address these criticisms by allowing either whole road segment reversibility or none. SODTA-TCR is a restricted version of the SODTA-LCR, which in turn is a restricted version of SODTA-CR. SODTA-BCR accounts for the limited resources for the operation and construction of the contraflow interchange segments such as the required number of police patrol cars to block the intersections at the beginning, end, and along the reversed segments.

SODTA-CR, SODTA-LCR, SODTA-TCR, and SODTA-BCR have two major assumptions. First, these models assume that drivers fully follow the central instructions on the system-optimal evacuation paths assigned to different drivers. Thus, these models can be single-level because the drivers and the evacuation manager share the same objective in minimizing total system travel time (TSTT). Second, the capacity reversibility is unchanged (static) over the simulation period, so the models cannot determine the optimal starting time and duration of reversibility. Tuydes and Ziliaskopoulos (2006) pointed out that a deficiency of the SODTA-based formulations is high computational cost due to their analytical nature, and proposed a tabu-based heuristic to address this deficiency. Because problems of this type have complex solution space, a majority of the research efforts have focused on tackling this stream of problems using meta-heuristics (e.g., Sarma and Adeli, 2001; Fan and Machemehl, 2008; Ng et al., 2009; Unnikrishnan et al., 2009; Zeferino et al., 2009; Kang et al., 2009; Yang et al., 2007; Kaveh and Shojaee, 2007; Paya et al., 2008; Mohan Rao and Shyju, 2010). Among various streams of meta-heuristics, the GA has been widely used, and it is recognized as an effective search procedure for these types of difficult optimization problems. Since 1993, GAs have been used in various civil engineering fields, such as construction engineering (e.g., Al-Bazi and Dawood, 2010; Cheng and Yan, 2009), transportation engineering (e.g., Vlahogianni et al., 2007; Teklu et al., 2007; Lee and Wei, 2010), highway engineering (e.g., Kang et al., 2009), and design optimization (e.g., Adeli and Cheng, 1994a,b; Hung and Adeli, 1994; Adeli and Kumar, 1995a,b; Sarma and Adeli, 2000a,b, 2001, 2002; Kim and Adeli, 2001; Mathakari et al., 2007; Dridi et al., 2008), structural control (e.g., Jiang and Adeli, 2008), and environmental pollution (e.g., Martínez-Ballesteros et al., 2010).

Xie et al. (2010) proposed a bi-level model for the combined lane-based capacity reversibility and crossing elimination problem. The model is bi-level to capture different objectives between the roadway manager (minimize TSTT) and the drivers (minimize individual travel time). In other words, Xie et al. (2010) assume drivers do not receive instructions from the roadway manager, and behave in a user-optimal manner. Xie et al. (2010) also assume static reversibility, and developed a Lagrangian relaxation-based tabu search.

Our proposed model is bi-level and allows lane-based capacity reversibility, similar to Xie et al. (2010). However, our model allows time-varying reversibility with different reversibility durations for various candidate link pairs, so that the optimal starting times and the optimal reversibility durations for candidate link pairs can be determined for peak-period traffic management on a daily basis. In our solution method, we employ the dual analysis results from Tuydes and Ziliaskopoulos (2004) in developing decoding procedures for the proposed GAs.

3 PROPOSED FORMULATION

Tuydes (2005) formulated the SODTA-based static lane-based capacity reversibility problem as a mixed integer program (SODTA-LCR). We extend SODTA-LCR to become a mixed-zero-one continuous bi-level program (BLP) for the combined UODTA and TVLCR problem. The proposed model is denoted by BLP-TVLCR. The upper-level problem minimizes TSTT subject to the TVLCR constraints and the UODTA (the nested program). It is noted that different specifications of UODTA produce slightly different models; that is, using the linear programming formulation of Ukkusuri (2002) creates a mixed 0–1 continuous linear bi-level program, but is only suitable for a single destination. On the other hand, using the Visual Interactive System for Transport Algorithms (VISTA) simulator allows solution with multiple destinations, but destroys the linear structure of the lower level.

This formulation allows us to devise an approximation algorithm to estimate the dual variables of the lower-level linear constraints with respect to the upper-level linear objective function in our proposed solution method. It is assumed that possible lane reversal strategies for all candidate link pairs are limitedly enumerated, including starting times, reversibility durations, and numbers of reversed lanes. The lane-reversal starting time for each candidate link pair is within its allowable range, and the lane-reversal duration for each candidate link pair is within a common feasible range. We do not allow the total link reversibility by setting the minimum number of lanes for each direction to one, so that the network connectivity problem is not resulted. Over the simulation period, the lane reversal is allowed at most once for each candidate link pair. Two-way streets are considered in the formulation. However, for a one-way street, an artificial link with zero capacity can be added in the opposite direction to represent the reversibility potential (Tuydes, 2005). In this way, the total network capacity remains the same although the number of links in the augmented network may be increased. The proposed formulation (BLP-TVLCR) is shown below and the notations are given in Table 1.

$$\min_{z,p,v} \sum_{(i,j)\in E_S} \sum_{t\in T} \left(t \cdot y_{ij}^t\right) \tag{1}$$

subject to

$$\sum_{\phi\in\Phi_{\xi-\xi^*}} v_{\xi-\xi^*,\phi} = 1 \ \ \forall \xi-\xi^* \in \Xi \tag{2}$$

$$p_{i-i^*}^t = \sum_{\phi\in\Phi_{\xi-\xi^*}(t)} v_{\xi-\xi^*,\phi} \ \ \forall i-i^* \in \Psi(\xi-\xi^*), \ \forall t \in T_{\xi-\xi^*}, \forall \xi-\xi^* \in \Xi \tag{3}$$

$$\sum_{\phi\in\Phi_{\xi-\xi^*}(t)} v_{\xi-\xi^*,\phi} = \sum_{k\in\Lambda_{\xi-\xi^*}} z_{\xi-\xi^*,k}^t \forall t \in T_{\xi-\xi^*}, \forall \xi-\xi^* \in \Xi \tag{4}$$

$$v_{\xi-\xi^*,\phi} \in \{0,1\} \ \ \forall \phi \in \Phi_{\xi-\xi^*}, \forall \xi-\xi^* \in \Xi \tag{5}$$

$$z_{\xi-\xi^*,k}^t \in \{0,1\} \ \ \forall k \in \Lambda_{\xi-\xi^*}, \forall t \in T_{\xi-\xi^*}, \forall \xi-\xi^* \in \Xi \tag{6}$$

$$r_i^t = \frac{\sum_{k\in\Lambda_{\xi-\xi^*}} \left(k \cdot z_{\xi-\xi^*,k}^t\right)}{l_{\xi-\xi^*}}; \ r_{i^*}^t = 1 - r_i^t \ \ \forall i-i^* \in \Psi(\xi-\xi^*), \ \forall \xi-\xi^* \in \Xi, \forall t \in T, \tag{7}$$

$$p_{i-i^*}^t = 0 \ \forall i-i^* \in \Psi, \forall t \in T \backslash T_{i-i^*} \tag{8}$$

$$z_{\xi-\xi^*,k}^t = 0 \ \ \forall k \in \Lambda_{\xi-\xi^*}, \forall \xi-\xi^* \in \Xi, \forall t \in T \backslash T_{\xi-\xi^*} \tag{9}$$

$$(\boldsymbol{x}, \boldsymbol{y}) \in \text{UODTA}(\boldsymbol{p}, \boldsymbol{r}) \tag{10}$$

where

$$\boldsymbol{x} = \left[x_i^t \forall i \in C, t \in T\right]; \quad \boldsymbol{y} = \left[y_{ij}^t \forall (i,j) \in E, t \in T\right];$$
$$\boldsymbol{p} = \left[p_{i-i^*}^t \forall i-i^* \in \Psi, t \in T\right];$$
$$\boldsymbol{r} = \left[r_i^t; r_{i^*}^t \forall i-i^* \in \Psi, t \in T\right].$$

The candidate roadway sections for capacity reversibility are contained in the set Ξ. A candidate link pair $\xi-\xi^*$ has the total number of lanes of $l_{\xi-\xi^*}$, and the link pair $\xi-\xi^*$ is broken down into cell pairs contained in the set $\Psi(\xi-\xi^*)$ according to the CTM. The set of feasible numbers of lanes in the redesigned network is $\Lambda_{\xi-\xi^*} = \{1,\ldots,I_{\xi-\xi^*}-1\}$ with the assumption that total capacity reversibility is prohibited. The feasible capacity reversibility time periods for link pair $\xi-\xi^*$ are contained in the set $\Phi_{\xi-\xi^*}$. The set $\Phi_{\xi-\xi^*}(t)$ contains the time period $\phi \in \Phi_{\xi-\xi^*}$ that includes time interval t. The leader's objective function (Equation (1)) minimizes TSTT subject to a set of time-varying capacity reversibility constraints (Equations (2)–(9)) and the UODTA program (Equation (10)). TSTT basically equals the

Table 1
Notations for proposed mathematical formulation

Sets	
$\Xi =$	set of candidate link pairs ($\xi - \xi^* \in \Xi$)
$T =$	set of discrete time intervals
E; $E_s =$	set of cell connectors; set of sink cell connectors
$FS(i) =$	set of cell connectors emanating from cell i
$RS(i) =$	set of cell connectors emanating to cell i
$\Psi =$	set of candidate cell pairs for lane reversal implementation; $\Psi \subseteq C$
$\Psi(\xi - \xi^*) =$	set of candidate cell pairs corresponding to link pair $\xi - \xi^*$; (Based on CTM, a link pair $\xi - \xi^*$ is broken down into several cell pairs $i - i^* \in \Psi(\xi - \xi^*)$)
$\Lambda_{\xi-\xi^*} =$	set of feasible numbers of lanes corresponding to link pair $\xi - \xi^*$ in the redesigned network; $\Lambda_{\xi-\xi^*} = \{1, \ldots, l_{\xi-\xi^*} - 1\}$
$\Phi_{\xi-\xi^*} =$	set of feasible reversal time periods for link pair $\xi - \xi^*$ (Note: $\Phi_{\xi-\xi^*}$ must be enumerated based on allowable ranges of reversibility starting time and reversibility duration) (see Table 2 for an example)
$\Phi_{\xi-\xi^*}(t) =$	set of feasible reversal time periods covering time interval t for link pair $\xi - \xi^*$
$T_{\xi-\xi^*} =$	set of time intervals composing the time window for all feasible lane-reversal time periods for link pair $\xi - \xi^*$ (Note: $T_{\xi-\xi^*} \subset T$) (see Table 2 for an example)
$T_{i-i^*} =$	set of time intervals composing the time window for all feasible lane-reversal time periods for cell pair $i - i^*$ (Note : $\mathrm{T}_{\mathrm{i}-\mathrm{i}^*} = \mathrm{T}_{\xi-\xi^*}\ \forall \mathrm{i} - \mathrm{i}^* \in \Psi(\xi - \xi^*)$)
Parameters	
$\delta_i^t =$	ratio of link free flow speed and backward propagation speed for cell i and time interval t
$N_i^t =$	maximum number of vehicles in cell i at time interval t
$N_{i-i^*}^t =$	maximum number of vehicles in cell pair $i - i^*$ at time interval t ($N_{i-i^*}^t = N_i^t + N_{i^*}^t$)
$Q_i^t =$	maximum number of vehicles that can flow into or out of cell i during time interval t
$Q_{i-i^*}^t =$	max number of vehicles that can flow into or out of cell pair $i - i^*$ during time interval t; ($Q_{i-i^*}^t = Q_i^t + Q_{i^*}^t$)
$l_{\xi-\xi^*} =$	total number of lanes corresponding to link pair $\xi - \xi^*$
Variables	
$x_i^t =$	number of vehicles in cell i at time interval t
$y_{ij}^t =$	number of vehicles moving from cell i to cell j at time interval t
$z_{\xi-\xi^*,k}^t =$	1 if number of lanes corresponding to link ξ is equal to k and number of lanes corresponding to link ξ^* is equal to $l_{\xi-\xi^*} - k$ in time interval t in the redesigned network; equal to 0 otherwise. $\forall k \in \Lambda_{\xi-\xi^*} = \{1, \ldots, l_{\xi-\xi^*} - 1\}$
$p_{i-i^*}^t =$	1 if the lane reversal is implemented in time interval t on cell pair $i - i^*$; 0 otherwise
$v_{\xi-\xi^*,\phi} =$	1 if the lane-reversal time period ϕ for link pair $\xi - \xi^*$ is selected; 0 otherwise. $\forall \phi \in \Phi_{\xi-\xi^*}$
$r_i^t =$	ratio of redesigned capacity of cell i to the cell pair capacity in time interval t

difference between the summation of arrival times at the sink cell ($\sum_{(i,j)\in E_S}\sum_{t\in T}(t \cdot y_{ij}^t)$) and the summation of departure times from the source cells (a constant due to the fixed departure time OD demands assumption, so it can be dropped from the model without affecting the optimal solution). Equation (2) enforces that only one capacity reversibility time period is chosen for each candidate link pair; that is, $v_{\xi-\xi^*,\phi'} = 1$ and $v_{\xi-\xi^*,\phi} = 0\ \forall \phi \neq \phi', \phi \in \Phi_{\xi-\xi^*}$. Equation (3) determines the variable $p_{i-i^*}^t$, where $p_{i-i^*}^t = 1$ if the cell pair

Table 2
Illustration of sets $\Phi_{\xi-\xi^*}$, $\Phi_{\xi-\xi^*}(t)$, and $T_{\xi-\xi^*}$

*t*1	*t*2	*t*3	*t*4	*t*5	*Link reversal time period*
					$\phi 1$
					$\phi 2$
					$\phi 3$
					$\phi 4$

Assume allowable range of reversibility starting time is [t2,t3] and allowable range of reversibility duration is [1,2 time intervals].
$\Phi_{\xi-\xi^*} = \{\phi 1, \phi 2, \phi 3, \phi 4\}$; $T_{\xi-\xi^*} = \{t2, t3, t4\}$
$\Phi_{\xi-\xi^*}(t1) = \Phi_{\xi-\xi^*}(t5) = \{\}$; $\Phi_{\xi-\xi^*}(t2) = \{\phi 1, \phi 2\}$;
$\Phi_{\xi-\xi^*}(t3) = \{\phi 2, \phi 3, \phi 4\}$; $\Phi_{\xi-\xi^*}(t4) = \{\phi 4\}$

$i - i^*$ at time interval t adopts a capacity-reversibility strategy, and $p^t_{i-i^*} = 0$ otherwise. Equation (3) guarantees that the cell pairs, belonging to the same link pair, adopt the same capacity reversibility time period. Equation (4) enforces $z^t_{\xi-\xi^*,k}\ \forall k \in \Lambda_{\xi-\xi^*}$ to be 0 if no capacity reversibility strategy is adopted at time interval t on link pair $\xi - \xi^*$ (i.e., $\sum_{\phi\in\Phi_{\xi-\xi^*}(t)} v_{\xi-\xi^*,\phi} = 0$). If a capacity reversibility strategy is adopted at time interval t (i.e., $\sum_{\phi\in\Phi_{\xi-\xi^*}(t)} v_{\xi-\xi^*,\phi} = 1$), then Equation (4) redesigns the numbers of lanes for link pair $\xi - \xi^*$ at time interval t (e.g., $z^t_{\xi-\xi^*,k'} = 1$ and $z^t_{\xi-\xi^*,k} = 0\ \forall k \neq k', k \in \Lambda_{\xi-\xi^*}$). Apparently, the original numbers of lanes for each link pair can be chosen; then, it implies that the formulation allows a do-nothing option on each candidate link pair. Equations (5) and (6) enforce $v_{\xi-\xi^*,\phi}$ and $z^t_{\xi-\xi^*,k}$ to be binary variables. Equation (7) determines the ratio of the redesigned capacity of each cell pair $i - i^*$. Equations (8) and (9) enforce the variables $p^t_{i-i^*}$ and $z^t_{\xi-\xi^*,k}$ to be 0 for the time intervals outside the capacity-reversal time window $T_{\xi-\xi^*}$, where $T_{\xi-\xi^*}$ contains possible time intervals to adopt capacity reversibility.

The nested UODTA linear program (10) makes all vehicles behave in the user-optimal manner; that is, UODTA returns user-optimal flows ($\boldsymbol{x}$ and $\boldsymbol{y}$) given input parameters $\boldsymbol{p}$ and $\boldsymbol{r}$. The constraints include the cell mass conservation. The traffic flow between two cells is constrained by the number of vehicles occupying the upstream cell, the remaining capacity of the downstream cell, and the maximum flow that can get out of the upstream cell and into the downstream cell. The specific modifications to UODTA are on the following three constraint sets:

$$\sum_{(j,i)\in RS(i)} y^t_{ji} \le \delta^t_i\left(r^t_i N^t_{i-i^*} + \left(1-p^t_{i-i^*}\right)N^t_i - x^t_i\right) \quad \forall i \in \Psi, t \in T \tag{11}$$

$$\sum_{(j,i)\in RS(i)} y^t_{ji} \le r^t_i Q^t_{i-i^*} + (1-p^t_{i-i^*})Q^t_i \quad \forall i \in \Psi, t \in T \tag{12}$$

$$\sum_{(i,j)\in FS(i)} y^t_{ij} \le r^t_i Q^t_{i-i^*} + (1-p^t_{i-i^*})Q^t_i \quad \forall i \in \Psi, t \in T \tag{13}$$

Equations (11–13) concern candidate cell pairs. For candidate cell pair $i - i^*$, if the time interval t is contained in $T\backslash T_{i-i^*}$, then $p^t_{i-i^*} = 0$ (see Equation (8)), $r^t_i = 0$ (see Equations (7) and (9)), and Equations (11–13) result in the constraints with original values of N^t_i and Q^t_i. If the time interval t is contained in T_{i-i^*}, then the upper-level problem determines whether $p^t_{i-i^*} = 1$ or $p^t_{i-i^*} = 0$. If $p^t_{i-i^*} = 1$, then Equations (3), (4), and (7) ensure that $r^t_i > 0$, and Equations (11–13) result in the constraints with the associated redesigned values of N^t_i and Q^t_i. If $p^t_{i-i^*} = 0$, then Equations (3), (4), and (7) ensure that $r^t_i = 0$, and Equations (11–13) result in the constraints with original values of N^t_i and Q^t_i.

4 SOLUTION METHOD

A simulation-based heuristic approach is proposed in this article. The lower-level program is replaced by the simulation-based UODTA (Ziliaskopoulos and Waller, 2000), which uses a mesoscopic simulator based on an extension of CTM, to propagate traffic and satisfy capacity constraints as well as the first-in first-out traffic property. The upper-level program (Equations (1–9)) can be substituted by a metaheuristic algorithm. In this article, the GA is adopted due to its evident efficiency and effectiveness in literature (e.g., Lee and Wei, 2010; Adeli and Cheng, 1994a,b; Sarma and Adeli, 2001; Mathakari et al., 2007; Teklu et al., 2007; Ng et al., 2009; Unnikrishnan et al., 2009; Kang et al., 2009). Because the proposed formulation is linear bi-level, the dual variables of the lower-level constraints (Equations (11–13)) with respect to the upper-level objective may be approximated in a similar manner as Lin et al. (2008). This inspires us to incorporate the approximated dual variables in the decoding procedure of GA. Three decoding procedures are developed with the increasing degree of randomness, yielding three problem-specific GAs. These are tested against each other and the simple GA in the next section. This study employs a C source code named GENESIS Version 5.0 (Grefenstette, 1990) for the GA implementation with major modifications that will be described in the next subsections.

4.1 Decision variables and solution representations

There are three sets of decision variables for each candidate link pair $\xi - \xi^*$ in the proposed GA: reversibility starting times ($ST_{\xi-\xi^*}$), reversibility durations ($RD_{\xi-\xi^*}$), and the modified numbers of lanes in both driving directions after reversibility (MNL_ξ and MNL_{ξ^*}). Note that the reversibility ending times ($ET_{\xi-\xi^*}$) is the summation of $ST_{\xi-\xi^*}$ and $RD_{\xi-\xi^*}$. Based on our initial experimental results, the direct use of these variables regarding the capacity reversibility time period and the redesigned numbers of lanes for all candidate link pairs indeed did not yield a significant improvement in TSTT. This caused us to introduce randomness to the incorporation of problem-specific knowledge. An additional decision variable for each candidate link pair $\xi - \xi^*$ is created: the reversibility indicator variable ($RIV_{\xi-\xi^*}$). The indicator variable is equal to 1 if the reversibility is allowed for the candidate link pair, and equal to 0 if the reversibility is prohibited. The reversibility indicator variable allows randomness to play a role on the decision of capacity reversibility adoption for each candidate link pair. Without this variable, it is likely that the population would be composed of similar chromosomes, and thus premature convergence of GA would result. The reversibility indicator variable is employed to prevent the premature convergence of GA.

The appropriate representation for $RIV_{\xi-\xi^*}$ is binary; those for $ST_{\xi-\xi^*}$ and $RD_{\xi-\xi^*}$ are real-valued; and those for MNL_{ξ} and MNL_{ξ^*} are integers. That is, the genetic structures are the vectors of mixed integers. Because $ST_{\xi-\xi^*}$ and $RD_{\xi-\xi^*}$ have to be within their allowable ranges $[lb_{ST}, ub_{ST}]$ and $[lb_{RD}, ub_{RD}]$, respectively; thus, the fractional variables (ranged between 0 and 1) can be employed instead of the real-valued variables. Also, $RIV_{\xi-\xi^*}$, MNL_{ξ}, and MNL_{ξ^*} can be determined by fractional variables. The fractional variables can be translated into a binary string. Thus, the vectors of mixed integers can be encoded into binary string structures.

4.2 Computer programming implementation for the TVLCR

In our implementation, duplicated links corresponding to candidate links are added to the network, and time-based capacity factors associated with these candidate and duplicated links are employed to represent various capacity reversibility strategies. For example, a candidate link pair is link (A,B) and link (B,A) (i.e., from nodes A to B and from nodes B to A, respectively). The duplicated links (A′,B′) and (B′,A′) are added to the network such that (A′,B′) is a copy of (A,B) and (B′,A′) is a copy of (B,A). To incorporate different TVLCR strategies, a set of time-based capacity factors associated with each candidate link is employed. A capacity factor, which is ranged between 0 and 1, indicates the proportion of the original link capacity for a candidate link during a time period. To illustrate, links (A,B) and (B,A) represent two-lane roadways, and a TVLCR strategy states that the link (A,B) should have three lanes during the second hour of the three-hour simulation period (in other words, a lane should be reversed from the second direction to the first direction during the second hour). Then, from 0 to 3,600 seconds and from 7,201 to 10,800 seconds (i.e., during the first hour and the third hour), the capacity factors of the original links (A,B) and (B,A) are equal to 1, and the capacity factors of the duplicated links (A′,B′) and (B′,A′) are equal to 0. This implies no capacity reversibility in the first hour and the third hour. From 3,601 to 7,200 seconds (i.e., during the second hour), the capacity factors of (A,B) and (A′,B′) are equal to respective 1 and 0.5, and those of (B,A) and (B′,A′) are equal to respective 0.5 and 0. This implies three lanes in the first direction (i.e., an original two-lane roadway and a lane reversed from the coupled link) and one lane in the second direction (i.e., an original two-lane roadway becomes a one-lane road). Formally, the capacity factors of link pair $\xi - \xi^*$ can be determined from the subprocedure below.

Subprocedure $DetermineCF\ (\xi - \xi^*)$
  $CF_{\xi}(t) = 1, CF_{\xi^*}(t) = 1, CF_{\xi'}(t) = 0, CF_{\xi^{*\prime}}(t) = 0;$
    $\forall t \in [0, ST) \cup (ET, SimDuration]$
  For $t \in [ST, ET]$
    if $MNL_{\xi} < ONL_{\xi}$
      $CF_{\xi}(t) = MNL_{\xi}/ONL_{\xi}; CF_{\xi'}(t) = 0; CF_{\xi^*}(t) = 1;$
      $CF_{\xi^{*\prime}} = (MNL_{\xi^*} - ONL_{\xi^*})/ONL_{\xi^*}$
    else $CF_{\xi}(t) = 1; CF_{\xi'}(t) = (MNL_{\xi} - ONL_{\xi})/ONL_{\xi};$
      $CF_{\xi^*}(t) = MNL_{\xi^*}/ONL_{\xi^*}; CF_{\xi^{*\prime}}(t) = 0$

where

$CF_{\xi}(t)$ and $CF_{\xi^*}(t)$ = capacity factor for the original links ξ and ξ^* at time t;
$CF_{\xi'}(t)$ and $CF_{\xi^{*\prime}}(t)$ = capacity factor for the duplicated links ξ' and $\xi^{*\prime}$ at time t;
MNL_{ξ} and MNL_{ξ^*} = modified number of lanes after reversibility for links ξ and ξ^*
ONL_{ξ} and ONL_{ξ^*} = original number of lanes of links ξ and ξ^*

4.3 Problem-specific knowledge for development of GA2, GA3, and GA4

In the development of GA2, GA3, and GA4, the problem-specific knowledge of BLP-TVLCR is heuristically taken from the dual variable analysis of the analytical SODTA-CR in Tuydes (2005), which was employed in the tabu-based heuristic approach for the evacuation contraflow problem (Tuydes and Ziliaskopoulos, 2006). The relationship between the dual variable of the lower-level program and the upper-level objective in BLP-TVLCR is not known, but it may be approximated by the analysis of the single-level system-optimal counterpart. This approximation approach is in a similar manner to Lin et al. (2008). The dual variable analysis of the SODTA-CR is briefly described here. Over the analysis period, the total marginal cost of reversing one more unit of capacity in the direction of one cell will have the same marginal cost as that of reversing in the direction of the coupled cell (Tuydes and Ziliaskopoulos, 2004). The total marginal costs are the dual variables associated with the constraints Equations (11–13) in the proposed model. These dual variables are nonzero only when the corresponding constraints are binding (i.e., the storage or flow capacities are fully used). The marginal costs of reversing a unit capacity in a link can be approximated by a congestion measure: the total number of times over the analysis period that link capacities are used at the maximum levels.

In our work, the modified congestion measure is determined as follows. To account for time-varying reversibility, the simulation period (*SimDuration*) is divided into many time slices where the duration of time slice is called *SecPerSlice*. The congestion measure of link ξ in time slice j is a binary variable $RC_{\xi j}$, which equals to 1 if it is congested, and 0 otherwise. In time slice j, the traffic count per lane on candidate link ξ is calculated from the simulation-based UODTA flows: $link\_count\_per\_lane_{\xi j} = \frac{link\_count_{\xi j}+link\_count_{\xi' j}}{(CF_{\xi}(j)+CF_{\xi'}(j))\cdot ONL_{\xi}}$, where $link\_count_{\xi j}$ and $link\_count_{\xi' j}$ = traffic counts on original link ξ and duplicated link ξ' in time slice j; $CF_{\xi}(j)$ and $CF_{\xi'}(j)$ = link capacity factors of link ξ and ξ' during time interval j.

If the traffic count per lane on link ξ exceeds the factored jam density per lane during a time slice j, $RC_{\xi j}$ is set to 1, indicating the congestion on this link at this time slice j. The jam density per lane is calculated from the link length divided by the vehicle length, and the factored jam density per lane is equal to the product of common jam-density factor (*JDF*, an algorithm parameter) and the jam density per lane. Note that the factored jam density per lane is employed as a congestion criterion on links, and this is not used in the proposed math formulation. In our experiment, we use the vehicle length of 20 feet, which is determined from the sum of the default vehicle length and distance headway. The *JDF* of 0.5 is initially employed in the experiment as we assume that at 50% of the jam density, the traffic flow reaches capacity.

For each candidate link pair, the difference in the congestion measures over each time slice can be calculated, and this difference indicates the driving direction that should be assigned more lanes reversed from its coupled link during this time slice. For candidate link pair (ξ, ξ^*), if the direction of link ξ is the first direction and that of link ξ^* is the second direction, then we define $\Delta RC_{\xi-\xi^*,j} = RC_{\xi j} - RC_{\xi^* j}$. Then, $\Delta RC_{\xi-\xi^*,j}$ equals to 1 (i.e., $RC_{\xi j} = 1$ and $RC_{\xi^* j} = 0$), implying the link ξ should receive more lane(s) reversed from the link ξ^* during time slice j. $\Delta RC_{\xi-\xi^*,j}$ equals to –1 (i.e., $RC_{\xi j} = 0$ and $RC_{\xi^* j} = 1$) implies the opposite. $\Delta RC_{\xi-\xi^*,j}$ equals to 0 (i.e., $RC_{\xi j} = 0$ and $RC_{\xi^* j} = 0$; or $RC_{\xi j} = 1$ and $RC_{\xi^* j} = 1$), implying no reversibility should be adopted in time slice j. $FTS_{\xi-\xi^*}$ is the first time slice j with the nonzero value of $\Delta RC_{\xi-\xi^*,j}$ and within the range of allowable starting time $[lb_{ST}, ub_{ST}]$. $FTS_{\xi-\xi^*}$ is set to –1 if there is not such time slice. If $FTS_{\xi-\xi^*}$ is not equal to –1, $\Delta RC_{\xi-\xi^*}$ stores the nonzero value of $\Delta RC_{\xi-\xi^*,j}$ $LTS_{\xi-\xi^*}$ is the last successive time slice that has this same congestion-measure difference value; that is, the time slices $j+1, j+2, \ldots, LTS_{\xi-\xi^*}$ have the same congestion-measure difference value ($\Delta RC_{\xi-\xi^*}$) value as the time slice j ($\Delta RC_{\xi-\xi^*} = \Delta RC_{\xi-\xi^*,j}$).

4.4 Encoding procedure and decoding procedure

The lower bound, upper bound, and required precision of each decision variable must be specified. A decision variable i (gene i) can be replaced by a fractional variable f_i. The decision variables share the same lower bound ($f_i^{\min} = 0$) and upper bound ($f_i^{\max} = 1$). That is, the TVLCR constraints are replaced by boundary constraints (because a decision variable is determined by the corresponding fractional variable and feasible range). The fractional variables have the same required precision (*prec* after decimal point). The required bits (m) for each decision variable is determined from (Goldberg, 1989). $2^{m-1} < ((f_i^{\max} - f_i^{\min}) \cdot 10^{prec} + 1) \leq 2^m$. We consider the required precision of 2 is sufficient ($m = 7$). The total bits required to represent a solution (i.e., the length of a chromosome) are $Length = m \cdot N_{\text{var}} \cdot |\Xi|$, where N_{var} is the number of variables per candidate link pair; $|\Xi|$ the number of candidate link pairs. N_{var} for GA1, GA2, GA3, and GA4 are respective 4, 2, 2, and 3. $|\Xi|$ for the grid and Sioux Falls networks are 12 and 14. Further, the GA implementation translates the binary structures into the packed bit arrays based on the octal number representation to maximize both space and time efficiency in manipulating structures (Grefenstette, 1990). Figure 1 shows binary string structures as well as associated decoded variables employed in four GA variations. In GA1, GA2, GA3, and GA4, we employ respective 4, 2, 2, and 3 fractional variables per candidate link pair; for example, the fractional variables $f_{\xi-\xi^*,1}$, $f_{\xi-\xi^*,2}$, $f_{\xi-\xi^*,3}$, $f_{\xi-\xi^*,4}$ are used to determine the decision variables $RIV_{\xi-\xi^*}$; $ST_{\xi-\xi^*}$; $RD_{\xi-\xi^*}$; MNL_{ξ}; and MNL_{ξ^*} in GA1.

The decoding procedure translates a binary string into a set of real numbers ($real_i$) corresponding to a set of fractional variables (f_i); then, f_i is determined from $f_i = \frac{real_i}{2^m-1}$ $\forall i = 1, 2, \ldots, N_{\text{var}} \cdot |\Xi|$; where $real_i$ is a real number corresponding to the binary string associated with the ith decision variable. For BLP-TVLCR, there are only two constraint sets: TVLCR constraint and UODTA conditions. Recall that the former constraint set is replaced by the boundary constraints. Because $f_i^{\min} = 0$ and $f_i^{\max} = 1$, this decoding procedure always satisfies the boundary constraints. Also, because the simulation-based UODTA is employed for functional evaluation, the UODTA conditions are always

GA	Binary String for Link Pair $\xi - \xi^*$/ Fractional Variables /Decoded Decision Variables
GA1	_______;_______;_______;_______/ $f_{\xi-\xi^*,1}$; $f_{\xi-\xi^*,2}$; $f_{\xi-\xi^*,3}$; $f_{\xi-\xi^*,4}$ /$RIV_{\xi-\xi^*}$; $ST_{\xi-\xi^*}$; $RD_{\xi-\xi^*}$; MNL_{ξ} and MNL_{ξ^*}
GA2	________;________ / $f_{\xi-\xi^*,1}$; $f_{\xi-\xi^*,2}$ / $RIV_{\xi-\xi^*}$; $RD_{\xi-\xi^*}$
GA3	________;________ / $f_{\xi-\xi^*,1}$; $f_{\xi-\xi^*,2}$ / $RIV_{\xi-\xi^*}$; MNL_{ξ} and MNL_{ξ^*}
GA4	________;________;________/ $f_{\xi-\xi^*,1}$; $f_{\xi-\xi^*,2}$; $f_{\xi-\xi^*,3}$ / $RIV_{\xi-\xi^*}$; MNL_{ξ} and MNL_{ξ^*}; $RD_{\xi-\xi^*}$

Fig. 1. Illustration of binary string structures in proposed GA variations.

satisfied. Thus, the constraint handling mostly based on the concept of penalty functions that penalize infeasible solutions, is not required. The decoding procedures for GA1 are shown below.

Decoding Procedure for Candidate Link Pair $\xi - \xi^*$ in GA1

$RIV_{\xi-\xi^*} = 0$ if $f_{\xi-\xi^*,1} \in [0, 0.5)$ and 1 if $f_{\xi-\xi^*,1} \in [0.5, 1]$

If $RIV_{\xi-\xi^*} = 1$

$ST_{\xi-\xi^*} = lb_{ST} + (ub_{ST} - lb_{ST}) \cdot f_{\xi-\xi^*,2}; RD_{\xi-\xi^*} = lb_{RD} + (ub_{RD} - lb_{RD}) \cdot f_{\xi-\xi^*,3}$

$ET_{\xi-\xi^*} = ST_{\xi-\xi^*} + RD_{\xi-\xi^*}; MNL_{\xi} = 1 + \lfloor (ONL_{\xi} + ONL_{\xi^*} - 2 + 0.999999) \cdot f_{\xi-\xi^*,4} \rfloor$

$MNL_{\xi^*} = ONL_{\xi} + ONL_{\xi^*} - MNL_{\xi}$

Call subprocedure $DetermineCF(\xi - \xi^*)$

Else if $RIV_{\xi-\xi^*} = 0$

$RIV_{\xi-\xi^*} = -1; ST_{\xi-\xi^*} = 0; RD_{\xi-\xi^*} = SimDuration;$

$ET_{\xi-\xi^*} = ST_{\xi-\xi^*} + RD_{\xi-\xi^*}$

$MNL_{\xi} = ONL_{\xi}; MNL_{\xi^*} = ONL_{\xi^*}$

$CF_{\xi}(t) = 1; CF_{\xi^*}(t) = 1; CF_{\xi'}(t) = 0; CF_{\xi^{*\prime}}(t) = 0 \quad \forall t \in [0, SimDuration]$

The decoding procedures in GA2, GA3, and GA4 are developed based on the described problem-specific knowledge (specifically, $FTS_{\xi-\xi^*}$, $\Delta RC_{\xi-\xi^*}$, and $LTS_{\xi-\xi^*}$). Apparently, these variables indicate the time period that the unbalanced traffic densities takes place and the driving direction with higher traffic density. Based on our initial experimental results, the direct use of these variables regarding the capacity reversibility time period and the redesigned numbers of lanes did not yield a significant improvement in TSTT. Thus, we devise three decoding algorithms for GA that employ some variables from $FTS_{\xi-\xi^*}$, $\Delta RC_{\xi-\xi^*}$, and $LTS_{\xi-\xi^*}$ together with the genes to determine the reversibility time period and the redesigned numbers of lanes. These are named GA2, GA3, and GA4 with the increasing degree of randomness.

GA2, GA3, and GA4 determines the variable $RIV_{\xi-\xi^*}$ from the corresponding gene in the chromosome for link pair $\xi - \xi^*$. If $FTS_{\xi-\xi^*}$ is not equal to –1 (i.e., this link pair have unbalanced traffic densities, and may have capacity reversibility), then the variable $RIV_{\xi-\xi^*}$ is unchanged; otherwise, $RIV_{\xi-\xi^*}$ is set to 0. GA2 determines the starting reversal time from $FTS_{\xi-\xi^*}$ GA2 employs $\Delta RC_{\xi-\xi^*}$ to indicate the driving direction to be improved; then, it deterministically adds a lane in that direction and decreases a lane in the opposite direction. GA2 allows the corresponding gene in the chromosome to determine the variable $RD_{\xi-\xi^*}$ (i.e., randomness plays a role here). GA3 is similar to GA2 except two points. First, the variable $RD_{\xi-\xi^*}$ is determined from $LTS_{\xi-\xi^*}$, $FTS_{\xi-\xi^*}$, and its lower and upper limits. Second, the redesigned numbers of lanes are determined from the corresponding genes in the chromosome for link pair $\xi - \xi^*$. GA4 is the combination of GA2 and GA3. GA4 determines the variable $RD_{\xi-\xi^*}$ in the same way as GA2, but determines the redesigned numbers of lanes in the same way as GA3. Formally, the decoding procedures of GA2, GA3, and GA4 are shown below.

Decoding Procedure for Candidate Link Pair $\xi - \xi^*$ in GA2, GA3 and GA4

$RIV_{\xi-\xi^*} = 0$ if $f_{\xi-\xi^*,1} \in [0, 0.5)$ and 1 if $f_{\xi-\xi^*,1} \in [0.5, 1]$

If $FTS_{\xi-\xi^*} = -1$ (i.e., the evidence shows that reversibility is not necessary), set $RIV_{\xi-\xi^*} = 0$.

If $RIV_{\xi-\xi^*} = 1$

$ST_{\xi-\xi^*} = FTS_{\xi-\xi^*} \cdot SecPerSlice;$

$$RD_{\xi-\xi^*} = \begin{cases} lb_{RD} + (ub_{RD} - lb_{RD}) \cdot f_{\xi-\xi^*,2} \text{ for GA2} \\ max(lb_{RD}, \min(ub_{RD}, (LTS_{\xi-\xi^*} - FTS_{\xi-\xi^*} \\ \quad +1) \cdot SecPerSlice)) \text{ for GA3} \\ lb_{RD} + (ub_{RD} - lb_{RD}) \cdot f_{\xi-\xi^*,3} \text{ for GA4} \end{cases}$$

$ET_{\xi-\xi^*} = ST_{\xi-\xi^*} + RD_{\xi-\xi^*}$

If $\Delta RC_{\xi-\xi^*} = 1$,

$$MNL_{\xi} = \begin{cases} \min(MNL_{\xi} + 1, ONL_{\xi} + ONL_{\xi^*} - 1) \\ \quad \text{for GA2} \\ \lfloor (f_{\xi-\xi^*,2} \cdot (ONL_{\xi} + ONL_{\xi^*} - 1 - MNL_{\xi} + \\ \quad 0.999999) \rfloor + MNL_{\xi} \text{ for GA3 and GA4} \end{cases}$$

$MNL_{\xi^*} = ONL_{\xi} + ONL_{\xi^*} - MNL_{\xi}$

Else if $\Delta RC_{\xi-\xi^*} = -1$,

$$MNL_{\xi^*} = \begin{cases} \min(MNL_{\xi^*} + 1, ONL_{\xi} + ONL_{\xi^*} - 1) \\ \quad \text{for GA2} \\ \lfloor f_{\xi-\xi^*,2} \cdot (ONL_{\xi^*} + ONL_{\xi^*} - 1 - MNL_{\xi^*} + \\ \quad 0.999999) \rfloor + MNL_{\xi^*} \text{ for GA3 and GA4} \end{cases}$$

$MNL_{\xi} = ONL_{\xi} + ONL_{\xi^*} - MNL_{\xi^*}$
Else if $RIV_{\xi-\xi^*} = 0$, set $RIV_{\xi-\xi^*}$, $ST_{\xi-\xi^*}$, $RD_{\xi-\xi^*}$, $ET_{\xi-\xi^*}$, MNL_{ξ}, MNL_{ξ^*} to the incumbent.
Call subprocedure $DetermineCF(\xi - \xi^*)$

4.5 Fitness evaluation

After converting the chromosomes to the vectors of decision variables, the UODTA with the decoded TVLCR strategy is solved by the simulation-based UODTA. TSTT for each solution is used to calculate a fitness measure. Because the objective of the problem minimizes TSTT, the functional form shown in Step 2 is adopted to ensure that the less TSTT corresponds to the greater fitness value. TSTT is determined from the UODTA flows. For GA2, GA3, and GA4, the congestion measures ($FTS_{\xi-\xi^*}$, $\Delta RC_{\xi-\xi^*}$, and $LTS_{\xi-\xi^*}$) are calculated from the UODTA flows. This study employs the UODTA module in the VISTA (Ziliaskopoulos and Waller, 2000) to evaluate different TVLCR strategies for larger-size problems. The UODTA module in VISTA is a departure-time-based version of the simulation-based UODTA approach using RouteSim (Ziliaskopoulos and Lee, 1996), which is a mesoscopic simulator based on an extension of CTM, to propagate traffic and satisfy capacity constraints. The DTA module iteratively employs the time-dependent shortest path algorithm (Ziliaskopoulos and Mahmassani, 1994) to generate vehicle paths, and the inner approximation dynamic user equilibrium (IADUE) algorithm (Chang, 2004) for equilibration.

5 COMPUTATIONAL EXPERIENCE

We consider a grid network and the Sioux Falls network. These test problems are first described. Then, the performance comparisons of GA1, GA2, GA3, and GA4 on the grid network are discussed. The sensitivity analysis of *JDF* is performed. Subsequently, the identified best GA variation and the simple GA are applied to a Sioux Falls network. All experiments are performed on a Linux machine with an Intel(R) 3.00 GHz Xeon(TM) CPU and 32 GB memory, running under Fedora Core 10.

5.1 Test problems

Figure 2 shows the grid network composed of 9 nodes, 24 links. All links are three-lane and 2-miles long with the free flow speed of 49.5 miles per hour (mph) and the capacity of 1,000 vehicles per hour per lane (vphpl). The simulation period is 3 hours (6:00–9:00 AM). All 12 link pairs are candidates for TVLCR. We consider 20 O-D pairs (nodes 1, 3, 5, 7, and 9 are both sources

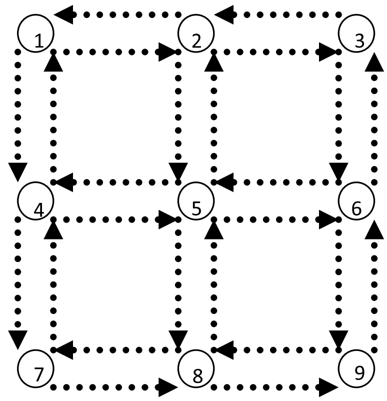


Fig. 2. Grid network (12 candidate link pairs).

and sinks). The O-D demands from node 5 to each of the other four sinks are 500 vehicle trips. The O-D demands from each of the other four sources to node 5 are 3,250 vehicle trips. The other O-D demands are 1,250 vehicle trips. Then, the total vehicle trips are 30,000. The static demands are distributed over the first 12 10-minute time slices by the weights: 0.05, 0.05, 0.05, 0.1, 0.1, 0.15, 0.15, 0.1, 0.1, 0.05, 0.05, and 0.05, respectively. Within each time slice, the demands are assumed uniformly distributed. The allowable ranges of reversal starting time and reversal duration are, respectively [6:20 AM, 7:30 AM] and [30 minutes, 90 minutes]. The TSTT of the original network is 2,035.12 hours.

Figure 3 shows a Sioux Falls network composed of 24 nodes and 76 arcs. Fourteen link pairs with dashed

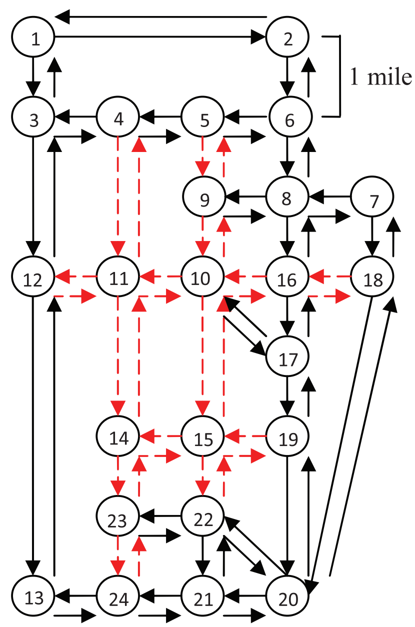


Fig. 3. Sioux Falls (14 candidate link pairs).

arrows are candidates for TVLCR. The network is the aggregated network of the city of Sioux Falls, South Dakota, used by many researchers in the literature. All links are two-lane with the capacity of 1,200 vphpl and the free flow speed of 49.5 mph. The link lengths can be determined from the distance scale in Figure 3. We consider 33 O-D pairs (see Karoonsoontawong and Waller, 2009) with 65,443 vehicle trips. The static demands are distributed over the first 12 15-minute time slices (i.e., first 3 hours) by the following weights: 0.05, 0.05, 0.05, 0.1, 0.1, 0.15, 0.15, 0.1, 0.1, 0.05, 0.05, and 0.05, respectively. The simulation period is 4 hours (6:00–10:00 AM). The allowable ranges of reversal starting time and reversal duration are respective [6:20 AM, 8:30 AM] and [30 minutes, 120 minutes]. The TSTT of the original network is 20,313.30 hours.

5.2 Performance comparison of proposed GA variations on grid test problem

The following GA parameters obtained from the GA parameter calibration for dynamic network design problem in Karoonsoontawong and Waller (2006) are employed for all GA runs: population size of 50, crossover rate of 0.6, and mutation rate of 0.001. Given the same parameter sets, the performance of the four GA variations can fairly be compared in terms of solution quality, convergence speed and computational time. In the comparison of GA variations, *JDF* of 0.5 is employed. After identifying the best GA variation, the sensitivity analysis of *JDF* will be performed. GA2, GA3, and GA4 determine time-dependent congestion measures, so GA2, GA3, and GA4 spend more CPU time per functional evaluation (trial) than GA1. A generation of GA1, GA2, GA3, and GA4 may have different numbers of trials, so we choose to compare the algorithm performance by the number of trials. We run the four GA variations for 1,000 trials. Figure 4 shows the convergence characteristics of the four algorithms. Table 3 presents the results of the best solutions obtained from the four algorithms. Apparently, GA3 outperforms GA1, GA2, and GA4 in terms of solution quality, CPU time found best and convergence speed; and GA1 appears second best. Interestingly, GA2 and GA4 yield worse results. Thus, the problem-specific knowledge has to be properly included to achieve excellent results.

Because GA3 is identified as the best GA variation, we then perform sensitivity analysis of *JDF* on GA3. Each run is for 1,000 trials. Figure 5 shows the convergence characteristics of GA3 with different *JDF* values, and Table 4 shows best solutions found from different *JDF* values. The *JDF* of 0.6 yields the much improved best solution found with the percentage improvement of 45.22 (i.e., the best solution found yields the improved TSTT that is 45.22% better than the initial TSTT), CPU time found best of 2.36 hours, and the trial found of 91. This reiterates that GA with the appropriate incorporation of problem-specific knowledge and

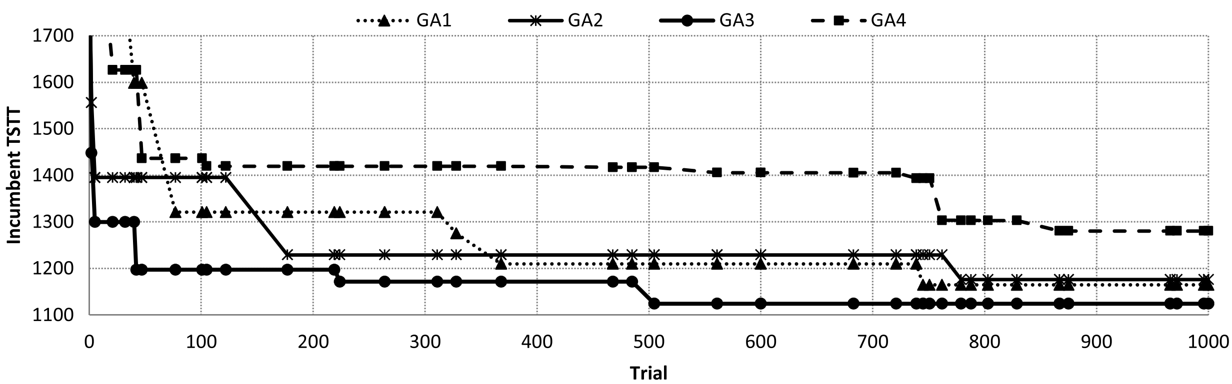


Fig. 4. Convergence characteristics of four GA variations on grid network.

Table 3
Best solutions from GA1 to GA4 on grid network

Algorithm	*Best obj. value (TSTT)*	*Percentage improvement from original TSTT*	*Trial found*	*CPU time found best (hours)*	*Total CPU time (hours)*
GA1	1164.77	42.77	745	18.34	24.50
GA2	1176.18	42.21	779	24.32	31.56
GA3	1124.35	44.75	505	13.95	29.41
GA4	1280.47	37.08	867	24.01	27.48

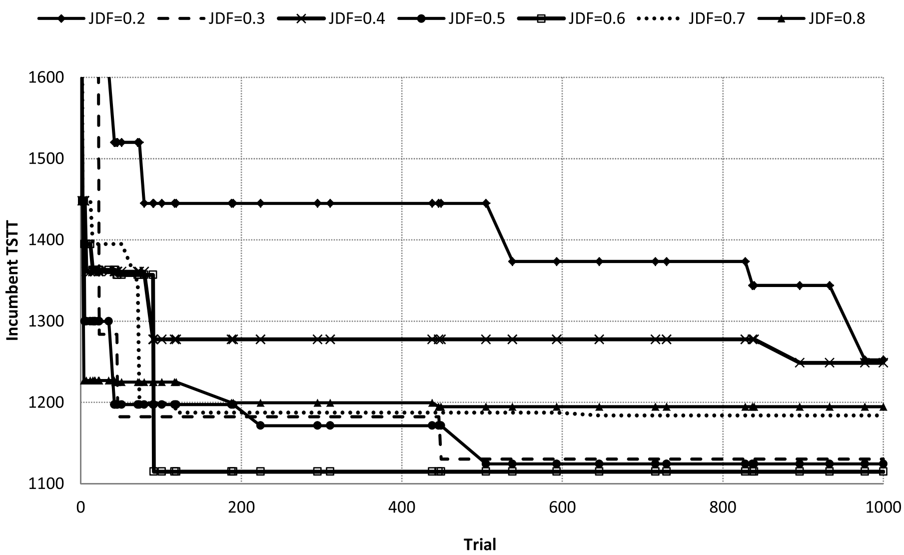


Fig. 5. Convergence characteristics of GA3 with different *JDF* values on grid network.

Table 4
Best solutions from GA3 with different *JDF* values on grid network

JDF	*Best obj. value (TSTT)*	*Percentage improvement from original TSTT*	*Trial found*	*CPU time found best (hours)*	*Total CPU time (hours)*
0.2	1252.36	38.46	977	28.95	29.60
0.3	1130.32	44.46	449	12.62	28.03
0.4	1248.78	38.64	896	21.21	23.52
0.5	1124.35	44.75	505	13.95	29.41
0.6	1114.79	45.22	91	2.42	25.97
0.7	1183.89	41.83	646	19.64	28.86
0.8	1194.70	41.30	446	13.20	29.23

Table 5
Best solution found on grid network

Link pair	*Capacity reversibility time period*	*Duration (minutes)*	*Number of lanes*
(1,4) & (4,1)	6:00–7:00; 7:00–7:30; 7:30–9:00	60; 30; 90	3 & 3; 4 & 2; 3 & 3
(4,7) & (7,4)	6:00–6:50; 6:50–7:30; 7:30–9:00	50; 40; 90	3 & 3; 2 & 4; 3 & 3
(3,6) & (6,3)	6:00–7:20; 7:20–7:50; 7:50–9:00	80; 30; 70	3 & 3; 5 & 1; 3 & 3

with a parameter calibration (GA3 with $JDF = 0.6$) indeed provides much better results when compared with simple GA (GA1). The best solution found is shown in Table 5.

In the initial traffic condition, among the 12 candidate link pairs, there are four candidate link pairs with unbalanced traffic densities; namely, link pairs (<u>14</u>, 41), (<u>12</u>,21), (23,<u>32</u>), and (<u>36</u>,63). The traffic on underlined links is congested (here defined as traffic density greater than 0.5 $\times$ jam density) and the other link in the pair is not. The four link pairs have unbalanced traffic densities during the respective time periods 6:30–7:50, 6:50–7:50, 7:20–7:50, and 6:30–7:00. After implementing the time-varying capacity reversibility in Table 5, the number of link pairs with unbalanced traffic densities is reduced to three link pairs, namely (<u>14</u>,41), (<u>36</u>,63), and (<u>78</u>,87) during respective time periods 6:50–7:40, 6:50–7:20, and 7:00–7:10. The link pairs with unbalanced traffic densities are more spread out over the network and the durations of unbalanced traffic densities are shorter than the initial traffic flow condition. Although the initial traffic density condition indicates that the four link pairs in the northern part of the network have unbalanced traffic densities, the best

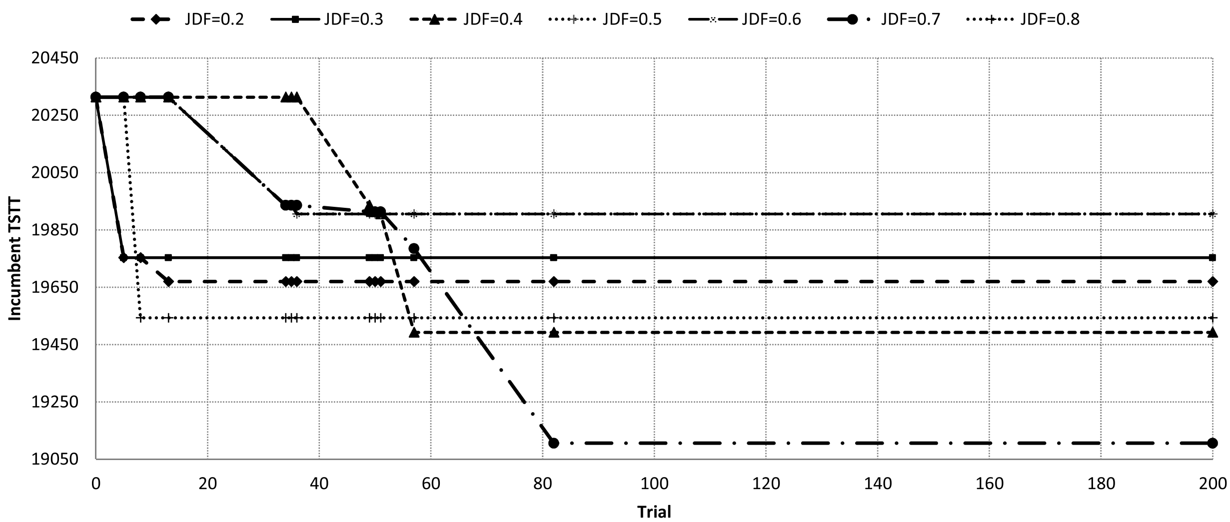


Fig. 6. Convergence characteristics of GA3 with different *JDF* values on the Sioux Falls network.

Table 6
Results of sensitivity analysis of *JDF* for GA3 on the Sioux Falls network

JDF	*Best obj. value (TSTT)*	*Percentage improvement from original TSTT*	*Trial found*	*CPU time found best (hours)*	*Total CPU time (hours)*
0.2	19669.73	3.17	13	5.86	100.08
0.3	19752.94	2.76	5	2.76	95.02
0.4	19492.35	4.04	57	22.55	73.06
0.5	19905.42	2.01	36	14.89	74.06
0.6	19905.42	2.01	36	14.89	65.54
0.7	19106.19	5.94	82	31.45	78.16
0.8	19543.05	3.79	8	3.98	79.62

solution found does not simply adopt the capacity reversibility on these four link pairs. Only two of the four link pairs and another link pair in the southern part of the network implement the capacity reversibility strategies. This is the nature of bi-level solution; the proposed bi-level formulation accounts for the dynamic user equilibrium behavior (the lower level) while the objective minimizes TSTT (the upper level).

5.3 Application of GA3 on the Sioux Falls problem

The simple GA (GA1) and the best GA variation identified from the previous section (GA3) are applied to the Sioux Falls network with the same set of GA parameters as that employed on the grid network. The sensitivity analysis of the parameter *JDF* is conducted for GA3 as shown in Figure 6 and Table 6. The stopping criterion of 200 total trials is employed. GA1 finds its best objective value of 19,776.38 at trial 13, whereas GA3 with the *JDF* of 0.6 does not yield a better solution (its best objective value of 19,905.42 at trial 36). Apparently, the optimal *JDF* value of 0.6 found on the grid network no longer yields a best result on the Sioux Falls network, and GA3 with *JDF* of 0.7 yields a best solution (objective value of 19,106.19 at trial 87). This reiterates the fact that the parameter *JDF* should be calibrated on new problems to achieve satisfactory results. Table 7 shows the best

Table 7
Best solution found on the Sioux Falls network

Link pair	*Capacity reversibility time periods*	*Duration (minutes)*	*Number of lanes*
(10,11) & (11,10)	6:00–8:10; 8:10–9:20; 9:20–10:00	130; 70; 40	2 & 2; 3 & 1; 2 & 2
(15,19) & (19,15)	6:00–6:40; 6:40–8:40; 8:40–10:00	40; 120; 80	2 & 2; 3 & 1; 2 & 2

solution found by GA3 on the Sioux Falls problem. The best solution found yields the improved TSTT that is 5.94% better than the initial TSTT. In the initial traffic condition, among the 14 candidate link pairs, there are 5 candidate link pairs with unbalanced traffic densities; namely, link pairs (1112,<u>1211</u>), (<u>1011</u>,1110), (1016, <u>1610</u>), (1423,<u>2314</u>), and (1522,<u>2215</u>). The traffic on underlined links is congested (here defined as traffic density greater than 0.5 × jam density) and the other link in the pair is not. The five link pairs have unbalanced traffic densities during the respective time periods 7:10–7:50, 8:10–8:40, 7:20–7:30, 7:10–8:00, and 7:30–7:50. After implementing the time-varying capacity reversibility in Table 7, the number of link pairs with unbalanced traffic densities is reduced to 3 link pairs, namely (1112,<u>1211</u>), (<u>1011</u>,1110), and (1423,<u>2314</u>) with respective time periods 7:10–7:50, 7:30–8:00, and 7:10–8:20.

6 SUMMARY AND CONCLUSIONS

A new formulation for TVLCR for daily traffic management application is proposed. Due to the NP-hard complexity of the formulation, the GAs with simulation-based UODTA are developed to solve multiorigin multidestination problems. The decision variables are starting times, reversal durations, and redesigned numbers of lanes for candidate link pairs (instead of cell pairs in the analytical model). An additional decision variable for a candidate link pair is the capacity reversibility indicator variable, which is added for the GA to prevent premature convergence. Four GA variations are proposed. GA1 is a simple GA. GA2, GA3, and GA4 are developed (with the *JDF*) based on problem-specific knowledge with increasing degrees of randomness. The problem-specific knowledge is adapted from the dual variable analysis of the analytical model, and involves the time-varying congestion measures. The experiment is conducted to compare the performances of the four GA variations on a grid network. The performance comparison is considered on three criteria: solution quality, convergence speed, and CPU time found best. We find that GA3 performs best on the three criteria on a grid test problem, whereas the simple GA appears second. A sensitivity analysis of *JDF* on GA3 shows that the best solution found can much further be improved when using the optimal *JDF* of 0.6. Furthermore, the identified best GA variation (GA3) and GA1 are performed on the Sioux Falls network where we found that GA3 with *JDF* of 0.7 outperforms GA1 and GA3 with other *JDF* values on the three criteria. In the best solution found on both networks, there is less number of link pairs with unbalanced traffic densities and the durations of unbalanced densities are also shorter when compared with the initial condition. Based on our computational experience, the GA with the appropriate inclusion of problem-specific knowledge and with a parameter (i.e., *JDF*) calibration indeed provides excellent results when compared with simple GA. The future research may extend the proposed formulation to account for the limited resources (or budgets) for the operation and construction of the lane-based capacity reversibility.

ACKNOWLEDGMENTS

The first author gratefully acknowledges the support from The Thailand Research Fund under contract number MRG5380066 and King Mongkut's University of Technology Thonburi Research Fund. The second author would like to acknowledge the National Science Council, Taiwan, R.O.C., for the funding support under project no. NSC 98-2410-H-006-119.

REFERENCES

Adeli, H. & Cheng, N.-T. (1994a), Augmented Lagrangian genetic algorithm for structural optimization, *Journal of Aerospace Engineering*, ASCE, **7**(1), 104–18.

Adeli, H. & Cheng, N.-T. (1994b), Concurrent genetic algorithms for optimization of large structures, *Journal of Aerospace Engineering*, ASCE, **7**(3), 276–96.

Adeli, H. & Jiang, X. (2003), Neuro-fuzzy logic model for freeway work zone capacity estimation, *Journal of Transportation Engineering*, ASCE, **129**(5), 484–93.

Adeli, H. & Kumar, S. (1995a), Distributed genetic algorithms for structural optimization, *Journal of Aerospace Engineering*, **8**(3), 156–63.

Adeli, H. & Kumar, S. (1995b), Concurrent structural optimization on a massively parallel supercomputer, *Journal of Structural Engineering*, ASCE, **121**(11), 1588–97.

Adeli, H. & Samant, A. (2000), An adaptive conjugate gradient neural network: wavelet model for traffic incident detection, *Computer-Aided Civil and Infrastructure Engineering*, **15**(4), 251–60.

Al-Bazi, A. & Dawood, N. (2010), Developing crew allocation system for precast industry using genetic algorithms, *Computer-Aided Civil and Infrastructure Engineering*, **25**(8), 581–95.

Chang, E. J. (2004). Time-varying intermodal person trip assignment, Ph.D. Dissertation, Northwestern University.

Cheng, T. M. & Yan, R. Z. (2009), Integrating messy genetic algorithms and simulation to optimize resource utilization, *Computer-Aided Civil and Infrastructure Engineering*, **24**(6), 401–15.

Daganzo, C. F. (1994), The cell transmission model: a simple dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation Research B*, **28**(4), 269–87.

Dridi, L., Parizeau, M., Mailhot, A. & Villeneuve, J. P. (2008), Using evolutionary optimisation techniques for scheduling water pipe renewal considering a short planning horizon, *Computer-Aided Civil and Infrastructure Engineering*, **23**(8), 625–35.

Fan, W. & Machemehl, R. B. (2008), Tabu search strategies for the public transportation network optimizations with variable transit demand, *Computer-Aided Civil and Infrastructure Engineering*, **23**(7), 502–20.

Ghosh-Dastidar, S. & Adeli, H. (2003), Wavelet-clustering-neural network model for freeway incident detection, *Computer-Aided Civil and Infrastructure Engineering*, **18**(5), 325–38.

Goldberg, D. E. (1989), *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, MA.

Grefenstette, J. J. (1990), *A User's Guide to GENESIS Version 5.0*. http://www.aic.nrl.navy.mil/galist/src/, accessed August 1, 2008.

Hamad, K., Shourijeh, M. T., Lee, E., Faghri, A. (2009), Near-term travel speed prediction utilizing Hilbert-Huang transform, *Computer-Aided Civil and Infrastructure Engineering* **24**(8), 551–76.

Hung S. L. & Adeli, H. (1994), A parallel genetic/neural network learning algorithm for MIMD shared memory machines, *IEEE Transactions on Neural Networks*, **5**(6), 900–9.

Jiang, X. & Adeli, H. (2004a), Object-oriented model for freeway work zone capacity and queue delay estimation, *Computer-Aided Civil and Infrastructure Engineering*, **19**(2), 144–56.

Jiang, X. & Adeli, H. (2004b), Wavelet packet-autocorrelation function method for traffic flow pattern analysis, *Computer-Aided Civil and Infrastructure Engineering*, **19**(6), 324–37.

Jiang, X. & Adeli, H. (2005), Dynamic wavelet neural network model for traffic flow forecasting, *Journal of Transportation Engineering*, ASCE, **131**(10), 771–9.

Jiang, X. & Adeli, H. (2008), Neuro-genetic algorithm for nonlinear active control of high rise buildings, *International Journal for Numerical Methods in Engineering*, **75**(8), 770–86.

Kalafatas, G. & Peeta, S. (2009), Planning for evacuation: insights from an efficient network design model, *Journal of Infrastructure Systems*, **15**(1), 21–30.

Kang, M. W., Schonfeld, P. & Yang, N. (2009), Prescreening and repairing in a genetic algorithm for highway alignment optimization, *Computer-Aided Civil and Infrastructure Engineering*, **24**(2), 109–19.

Karim, A. & Adeli, H. (2002a), Comparison of the fuzzy: wavelet RBFNN freeway incident detection model with the California algorithm, *Journal of Transportation Engineering*, ASCE, **128**(1), 21–30.

Karim, A. & Adeli, H. (2002b), Incident detection algorithm using wavelet energy representation of traffic patterns, *Journal of Transportation Engineering*, ASCE, **128**(3), 232–42.

Karim, A. & Adeli, H. (2003a), Fast automatic incident detection on urban and rural freeways using wavelet energy algorithm, *Journal of Transportation Engineering*, ASCE, **129**(1), 57–68.

Karim, A. & Adeli, H. (2003b), CBR model for freeway work zone traffic management, *Journal of Transportation Engineering*, ASCE, **129**(2), 134–45.

Karim, A. & Adeli, H. (2003c), Radial basis function neural network for work zone capacity and queue estimation, *Journal of Transportation Engineering*, ASCE, **129**(5), 494–503.

Karoonsoontawong, A. & Waller, S. T. (2005), A comparison of system- and user-optimal stochastic dynamic network design models using Monte Carlo bounding techniques, *Journal of the Transportation Research Board*, **1923**, 91–102.

Karoonsoontawong, A. & Waller, S. T. (2006), Dynamic continuous network design problem: linear bilevel programming and metaheuristic approaches. *Journal of the Transportation Research Board*, **1964**, 104–17.

Karoonsoontawong, A. & Waller, S. T. (2007), Robust dynamic continuous network design problem, *Journal of the Transportation Research Board*, **2029**, 58–71.

Karoonsoontawong, A. & Waller, S. T. (2009), Reactive Tabu search approach for the combined dynamic user equilibrium and traffic signal optimization problem, *Journal of the Transportation Research Board*, **2090**, 29–41.

Karoonsoontawong, A. & Waller, S. T. (2010), Integrated network capacity expansion and traffic signal optimization problem: robust bi-level dynamic formulation, *Networks and Spatial Economics*, **10**(4), DOI:10.1007/s11067-008-9071-x.

Kaveh, A. & Shojaee, S. (2007), Optimal design of scissor-link foldable structures using ant colony optimization algorithm, *Computer-Aided Civil and Infrastructure Engineering*, **22**(1), 56–64.

Kim, H. & Adeli, H. (2001), Discrete cost optimization of composite floors using a floating point genetic algorithm, *Engineering Optimization*, **33**(4), 485–501.

Lee, Y. & Wei, C.-H. (2010), A computerized feature selection method using genetic algorithms to forecast freeway accident duration times, *Computer-Aided Civil and Infrastructure Engineering*, **25**(2), 132–48.

Lin, D.-Y., Karoonsoontawong, A. & Waller, S. T. (2008), A Dantzig-Wolfe decomposition based heuristic scheme for bi-level dynamic network design problem, *Networks and Spatial Economics*, **11**(1), 101–26.

Liu, H. X. & Danczyk, A. (2009), Optimal sensor locations for freeway bottleneck identification, *Computer-Aided Civil and Infrastructure Engineering*, **24**(8), 535–50.

Martínez-Ballesteros, M., Troncoso, A., Martínez-Álvarez, F. & Riquelme, J. C. (2010), Mining quantitative association rules based on evolutionary computation and its application to atmospheric pollution, *Integrated Computer-Aided Engineering*, **17**(3), 227–42.

Mathakari, S., Gardoni, P., Agarwal, P., Raich, A. & Haukaas, T. (2007), Reliability-based optimal design of electrical transmission towers using multi-objective genetic algorithms, *Computer-Aided Civil and Infrastructure Engineering*, **22**(4), 282–92.

Mirchandani, P. B., Li, J.-Q. & Long, Y. (2010), Locating a surveillance infrastructure in and near ports or on other planar surfaces to monitor flows, *Computer-Aided Civil and Infrastructure Engineering*, **25**(2), 89–100.

Mohan Rao, A. R. & Shyju, P. P. (2010), A meta-heuristic algorithm for multi-objective optimal design of Hybrid Laminate Composite Structures, *Computer-Aided Civil and Infrastructure Engineering*, **25**(3), 149–70.

Ng, M. W., Lin, D.-Y. & Waller, S. T. (2009), Optimal long-term infrastructure maintenance planning accounting for traffic dynamics, *Computer-Aided Civil and Infrastructure Engineering*, **24**(7), 459–69.

Ng, M. W., Park, J. & Waller, S. T. (2010), A hybrid bi-level model for the optimal shelter assignment in emergency evacuations, *Computer-Aided Civil and Infrastructure Engineering*, **25**(8), 547–56.

Paya, I., Yepes, V., González-Vidosa, F. & Hospitaler, A. (2008), Multiobjective optimization of concrete frames by simulated annealing, *Computer-Aided Civil and Infrastructure Engineering*, **23**(8), 96–610.

Samant, A. & Adeli, H. (2000), Feature extraction for traffic incident detection using wavelet transform and linear discriminant analysis, *Computer-Aided Civil and Infrastructure Engineering*, **15**(4), 241–50.

Samant, A. & Adeli, H. (2001), Enhancing neural network incident detection algorithms using wavelets, *Computer-Aided Civil and Infrastructure Engineering*, **16**(4), 239–45.

Sarma, K. & Adeli, H. (2000a), Fuzzy genetic algorithm for optimization of steel structures, *Journal of Structural Engineering*, ASCE, **126**(5), 596–604.

Sarma, K. & Adeli, H. (2000b), Fuzzy discrete multicriteria cost optimization of steel structures, *Journal of Structural Engineering*, ASCE, **126**(11), 1339–47.

Sarma, K. C. & Adeli, H. (2001), Bi-level parallel genetic algorithms for optimization of large steel structures, *Computer-Aided Civil and Infrastructure Engineering*, **16**(5), 295–304.

Sarma, K. C. & Adeli, H. (2002), Life-cycle cost optimization of steel structures, *International Journal for Numerical Methods in Engineering*, **55**(12), 1451–62.

Shen, W., Nie, Y. & Zhang, H. M. (2007), Dynamic network simplex method for designing emergency evacuation plans, transportation research record, *Transportation Research Record: Journal of the Transportation Research Board, No. 2022*, Transportation Research Board of the National Academies, Washington, D.C., pp. 83–93.

Sun, D. J. & Kondyli, A. (2010), Modeling vehicle interactions during lane-changing behavior on arterial streets, *Computer-Aided Civil and Infrastructure Engineering*, **25**(8), 557–71.

Teklu, F., Sumalee, A. & Watling, D. (2007), A genetic algorithm for optimizing traffic control signals considering routing, *Computer-Aided Civil and Infrastructure Engineering*, **22**(1), 31–43.

Tuydes, H. (2005), Network traffic management under disaster conditions. Ph.D. Dissertation, Civil and Environmental Engineering, Northwestern University.

Tuydes, H. & Ziliaskopoulos, A. (2004), Network redesign to optimize evacuation contraflow. *Presented at 83rd Annual Meeting of the Transportation Research Board*, Washington, D.C.

Tuydes, H. & Ziliaskopoulos, A. (2006), Tabu-based heuristic approach for optimization of network evacuation contraflow. *Transportation Research Record: Journal of the Transportation Research Board, No. 1964*, Transportation Research Board of the National Academies, Washington, D.C., pp. 157–68.

Ukkusuri S. (2002), Linear programs for user optimal dynamic traffic assignment problem, Master's thesis, University of Illinois at Urbana Champaign.

Ukkusuri, S. V. & Waller, S. T. (2007), Linear programming models for the user and system optimal dynamic network design problem: formulations, comparisons and extensions, *Networks and Spatial Economics*, **8**(4), 383–406.

Unnikrishnan, A., Valsaraj, V., Damnjanovic, I. & Waller, S. T. (2009), Design and management strategies for mixed public private transportation networks: a meta-heuristic approach, *Computer-Aided Civil and Infrastructure Engineering*, **24**(4), 266–79.

Vlahogianni, E. I., Karlaftis, M. G. & Golias, J. C. (2007), Spatio-temporal short-term urban traffic flow forecasting using genetically-optimized modular networks, *Computer-Aided Civil and Infrastructure Engineering*, **22**(5), 317–25.

Xie, C., Lin, D-Y & Waller, S. T. (2010), A dynamic evacuation network optimization problem with lane reversal and crossing elimination strategies, *Transportation Research: Part E.*, **46**(3), 295–316.

Yang, Z., Yu, B. & Cheng, C. (2007), A parallel ant colony algorithm for bus network optimization, *Computer-Aided Civil and Infrastructure Engineering*, **22**(1), 44–55.

Ye, Z. & Zhang, Y. (2010), Speed estimation from single loop data using an unscented particle filter, *Computer-Aided Civil and Infrastructure Engineering*, **25**(7), 494–503.

Zeferino, J. A., Antunes, A. P. & Cunha, M. C. (2009), An efficient simulated annealing algorithm for regional wastewater systems planning, *Computer-Aided Civil and Infrastructure Engineering*, **24**(5), 359–70.

Ziliaskopoulos, A. K. & Lee, S. (1996), A cell transmission based assignment-simulation model for integrated freeway/surface street systems, *Transportation Research Record*, **1701**, 12–23.

Ziliaskopoulos, A. K. & Mahmassani, H. S. (1994), Time-dependent, shortest-path algorithm for real-time intelligent vehicle highway system applications, *Transportation Research Record* **1408**, 94–100.

Ziliaskopoulos, A. K. & Waller, S. T. (2000), An Internet-based geographic information system that integrates data, models and users for transportation applications, *Transportation Research Part C*, **8**, 427–44.

AN ENHANCED COMBINED TRIP DISTRIBUTION AND ASSIGNMENT PROBLEM

Ampol KAROONSOONTAWONG
Assistant Professor
Department of Civil Engineering
King Mongkut's University of Technology
126 Pracha Uthit Rd, Bang Mod,
Thung Khru, Bangkok 10140
Fax: +662 470 9145
E-mail: ampol.kar@kmutt.ac.th

ABSTRACT:

This paper proposes a mathematical formulation and a solution method to the enhanced combined trip distribution and traffic assignment. The trip distribution is a doubly-constrained gravity model. The traffic assignment is the paired-combinatorial-logit stochastic user equilibrium accounting for effects of congestion, stochastic perception error and path similarity. This is an enhancement to existing multinomial-logit (MNL)-based model. The proposed solution method is a disaggregate simplicial decomposition algorithm. I find that the relationship of O-D flow difference and dispersion factor is unclear, whereas link flow patterns from the two models are more identical at higher dispersion factors. The enhanced model assigns less flow to a path with higher average similarity index and higher path cost than MNL model. The enhanced model generally assigns less flow to links with more paths passing through than MNL model. The relationship between O-D flow allocation and the average similarity indices for O-D pairs is not obvious.

Key Words: stochastic user equilibrium, gravity model, combined travel demand model

1. INTRODUCTION

The combined distribution and assignment (CDA) problem is an instance of combined travel demand models. CDA simultaneously determines the distribution of trips between origins and destinations in a transportation network and the assignment of trips to routes in each origin-destination pair. The trip distribution is mostly assumed to be a gravity model with a negative exponential deterrence function. The static trip assignment is either user equilibrium model (UE) or stochastic user equilibrium model (SUE). UE assumes that drivers have complete and accurate information on the state of the network when they make their route choices, and drivers select optimal routes to benefit themselves the most. SUE assumes that trip assignment follows a probabilistic route choice model. The multinomial logit-based SUE model (MNL-SUE) is widely adopted in the literature. Evans (1976) formulated the CDA problem that integrates the gravity-model trip distribution and user-equilibrium trip assignment (CDA-UE). Erlander (1990) formulated the CDA that integrates the gravity-model trip distribution and multinomial-logit stochastic-user-equilibrium assignment (CDA-MNL-SUE). Lundgren and Patriksson (1998) outlined the solution algorithms for CDA-UE and CDA-MNL-SUE.

With the property of independence of irrelevant alternatives (IIA) in the MNL model, the MNL-SUE has an infamous deficiency in the incapability to account for similarities between different routes. Although the multinomial probit-based SUE model by Daganzo and Sheffi (1977) can account for similarity between different routes, it is not attractive due to the lack of closed form of probability function. Over the past years, researchers adopted other discrete choice model structures to SUE in order to capture the similarity between routes on the perceptions and decisions of drivers while keeping the analytical tractability of the logit choice probability function. The SUE models based on the modifications of MNL are C-logit model and path-size logit model. The SUE models based on the generalized extreme value theory are paired combinatorial logit model (Prashker and Bekhor, 2000), cross-nested logit model, logit kernel model, link-nested logit model, and generalized nested logit model. Chen et al. (2003) pointed out that among these extended logit models, the paired combinatorial logit model (PCL) is considered the most suitable for adaptation to the route choice problem due to two features that can be employed to address the IIA property in the MNL model.

In this paper, I propose a combined gravity-model distribution and pairedcombinatorial -logit stochastic-user-equilibrium assignment formulation (CDA-PCL-SUE) and develop a disaggregate simplicial decomposition algorithm. The trip distribution model is doubly constrained such that both the total flow generated at each origin node and the total flow attracted to each destination node are fixed and known.

2. EQUIVALENT MATHEMATICAL FORMULATION

Denote by CDA-PCL-SUE the proposed combined distribution-assignment (CDA) equivalent mathematical model. The underlying route choice in CDA-PCL-SUE is a hierarchical route choice model that decomposes the choice probability into two levels. The upper level computes the marginal probabilities $P(kj)$ of choosing an unordered route pair k and j, based on the similarity index and the systematic utility.
The lower level is a binary logit model that computes the conditional probabilities of choosing a route given the chosen route pair: $P(k|kj)$ and $P(j|kj)$. The underlying trip distribution in CDA-PCL-SUE is a doubly constrained model that requires the O-D flows out of an origin node and into a destination node to be equal to the known origin demands and destination demands, respectively.

The definitions of sets, parameters, decision variables and mathematical formulation are given below, followed by the first-order conditions that are shown to be identical to the PCL-SUE equations and gravity-model based trip distribution equations.

Set

K_{rs} = set of routes between origin r and destination s

L_{rs} = set of unordered route pairs between origin r and destination s

R = set of origins

S = set of destinations

RS = set of origin-destination (O-D) pairs

A = set of arcs

Parameters

O_r = total trips originated from origin *r*

D_s = total trips destined to destination *s*

θ = dispersion coefficient

β_{kj}^{rs} = measure of dissimilarity index between routes *k* and *j* connecting O-D *r-s* ($\beta_{kj}^{rs} = 1 - \sigma_{kj}^{rs}$)

σ_{kj}^{rs} = measure of similarity index between routes *k* and *j* connecting O-D *r-s*

δ_{ka}^{rs} =1 if arc *a* is on route *k* connecting origin *r* to destination *s*; 0 otherwise

Decision Variables

x_a = flow on link *a*

t_a = travel time on link *a*

q_{rs} = demand between origin *r* and destination *s*

$f_{k(kj)}^{rs}$ = flow on route *k* of route pair *kj* between origin *r* and destination *s*

Mathematical Formulation

$$\min z = z_1 + z_2 + z_3 \quad \text{(Eq.1.1)}$$

$$z_1 = \sum_{a \in A} \int_0^{x_a} t_a(w)dw \quad \text{(Eq.1.1a)}$$

$$z_2 = \frac{1}{\theta} \sum_{r \in R} \sum_{s \in S} \sum_{k \in K_{rs}} \sum_{\substack{j \in K_{rs} \\ j \neq k}} \beta_{kj}^{rs} f_{k(kj)}^{rs} \ln \frac{f_{k(kj)}^{rs}}{\beta_{kj}^{rs}} \quad \text{(Eq.1.1b)}$$

$$z_3 = \frac{1}{\theta} \sum_{r \in R} \sum_{s \in S} \sum_{k=1}^{|K_{rs}|-1} \sum_{j=k+1}^{|K_{rs}|} (1 - \beta_{kj}^{rs})(f_{k(kj)}^{rs} + f_{j(kj)}^{rs}) \ln \frac{f_{k(kj)}^{rs} + f_{j(kj)}^{rs}}{\beta_{kj}^{rs}} \quad \text{(Eq.1.1c)}$$

Subject to

$$\sum_{k \in K_{rs}} \sum_{\substack{j \in K_{rs} \\ j \neq k}} f_{k(kj)}^{rs} = q_{rs} \quad \forall r \in R, s \in S \quad \text{(Eq.1.2)}$$

$$\sum_{s \in S} q_{rs} = O_r \quad \forall r \in R \quad \text{(Eq.1.3)}$$

$$\sum_{r \in R} q_{rs} = D_s \quad \forall s \in S \quad \text{(Eq.1.4)}$$

$$f_{k(kj)}^{rs} \geq 0 \quad \forall k \in K_{rs}, kj \in L_{rs}, r \in R, s \in S \quad \text{(Eq. 1.5)}$$

$$x_a = \sum_{r \in R} \sum_{s \in S} \sum_{k \in K_{rs}} \sum_{\substack{j \in K_{rs} \\ j \neq k}} \delta_{ak}^{rs} \cdot f_{k(kj)}^{rs} \quad \forall a \in A \quad \text{(Eq. 1.6)}$$

The objective function (Eq.1.1) is composed of three components, similar to the objective of the PCL-SUE model. Eq.1.1a accounts for the congestion effects. Eq.1.1b and 1.1c are two entropy terms that represent the marginal and conditional probabilities in a hierarchical route choice model. Dissimilarity indices are incorporated into the objective function (Eq.1.1b and Eq.1.1c), allowing the model to capture the similarity effect and stochastic perception error effect in addition to the congestion effect (Eq.1.1a). Eq.1.2 enforces the summation of all path flows connecting an O-D pair to be equal to the O-D flows (q_{rs}) of this O-D pair. Eq.1.3 and Eq.1.4 are the O-D flow balance constraints for the origin nodes and destination nodes, respectively. Eq.1.5 are the non-negativity constraints for all path flow variables. Eq.1.6 determines the link flow variable from the summation of all path flows passing through this link. It is easy to show that the optimality conditions of the proposed formulation equal to the PCL formula in Eq.5.7-5.8 and the gravity-model based trip distribution equation:

$$q_{rs} = A_r B_s O_r D_s g(c^{rs}) \quad \forall r \in R, \ \forall s \in S \quad \text{(Eq.2)}$$

Where

$$A_r = \frac{\exp(\theta u_r - 1)}{O_r}; \ B_s = \frac{\exp(\theta v_s)}{D_s};$$

c^{rs} = vector of route travel times of O-D pair *r,s*;

$g(c^{rs}) =$

$$\sum_{k\in K_{rs}}\sum_{\substack{j\in K_{rs}\\ j\neq k}} \frac{\beta_{kj}^{rs}\cdot\exp\left(-\frac{\theta c_k^{rs}}{\beta_{kj}^{rs}}\right)}{\left(\exp\left(-\frac{\theta c_k^{rs}}{\beta_{kj}^{rs}}\right)+\exp\left(-\frac{\theta c_j^{rs}}{\beta_{kj}^{rs}}\right)\right)^{1-\beta_{kj}^{rs}}}$$

3. DISAGGREGATE SIMPLICIAL DECOMPOSITION ALGORITHM

The proposed algorithm for CDA-PCL-SUE is based on the disaggregate simplicial decomposition algorithm by Lundgren and Patriksson (1998) and Larsson and Patriksson (1992). The proposed algorithm alternates between two phases. In phase I (the restricted master problem), given known subsets of routes between O-D pairs $\hat{K}_{rs} \subseteq K_{rs}$ $\forall r \in R, s \in S$, of the total sets of routes in the network, the corresponding restriction of CDA-PCL-SUE (denoted by CDA-PCL-SUE-R) is solved approximately using a partial linearization descent algorithm, which is a descent algorithm for continuous optimization problems. In phase 2 (the column generation problem), at the approximate solution to CDA-PCL-SUE-R, the subsets $\hat{K}_{rs}$ are augmented by the generation of new routes, through the solution of a set of shortest path problems, given appropriately chosen link costs.

3.1. Phase I: Restricted Master Problem

The problem CDA-PCL-SUE-R is solved by a partial linearization descent algorithm (Patriksson, 1993). The projection of CDA-PCL-SUE-R onto the set of feasible route flows is employed. Given a feasible route flow vector $f^n = \{f_{k(kj)}^{rs^n}\}$ at some iteration n, an approximation of CDA-PCL-SUE-R is roughly solved in order to define an auxiliary feasible solution and a search direction. The approximate problem is constructed by linearizing the first term (z_1) of the objective function of CDA-PCL-SUE-R. The effect of this linearization is that the link costs are fixed at their levels given the current flow f^n; i.e. $\frac{\partial z_1(f^n)}{\partial f_{k(kj)}^{rs}} = c_k^{rs^n}$. The corresponding route costs are calculated as: $c_k^{rs^n} = \sum_{a\in A}\delta_{ka}^{rs}\cdot t_a(x_a^n)$ $\forall k \in \hat{K}_{rs}, r\in R, s\in S$, where x_a^n is the flow on arc a corresponding to the route flow f^n. The partially linearized problem (denoted by CDA-PCL-SUE-R-PL) becomes:

<u>Formulation of CDA-PCL-SUE-R-PL</u>

(the solution is denoted by $\underline{f}_{k(kj)}^{rs}$)

$$\min \tilde{z} = \tilde{z}_1 + \tilde{z}_2 + \tilde{z}_3 \quad \text{(Eq.3.1)}$$

$$\tilde{z}_1 = \sum_{r\in R}\sum_{s\in S}\sum_{k\in\hat{K}_{rs}}\sum_{\substack{j\in\hat{K}_{rs}\\ j\neq k}} c_k^{rs^n}\cdot f_{k(kj)}^{rs} \quad \text{(Eq.3.1a)}$$

$$\tilde{z}_2 = \frac{1}{\theta}\sum_{r\in R}\sum_{s\in S}\sum_{k\in\hat{K}_{rs}}\sum_{\substack{j\in\hat{K}_{rs}\\ j\neq k}} \beta_{kj}^{rs} f_{k(kj)}^{rs}\ln\frac{f_{k(kj)}^{rs}}{\beta_{kj}^{rs}} \quad \text{(Eq.3.1b)}$$

$$\tilde{z}_3 = \frac{1}{\theta}\sum_{r\in R}\sum_{s\in S}\sum_{k=1}^{|\hat{K}_{rs}|-1}\sum_{j=k+1}^{|\hat{K}_{rs}|} (1-\beta_{kj}^{rs})(f_{k(kj)}^{rs}+f_{j(kj)}^{rs})\ln\frac{f_{k(kj)}^{rs}+f_{j(kj)}^{rs}}{\beta_{kj}^{rs}} \quad \text{(Eq.3.1c)}$$

Subject to

$$\sum_{s\in S}\sum_{k\in\hat{K}_{rs}}\sum_{\substack{j\in\hat{K}_{rs}\\ j\neq k}} f_{k(kj)}^{rs} = O_r \quad \forall r\in R \quad \text{(Eq.3.2)}$$

$$\sum_{r\in R}\sum_{k\in\hat{K}_{rs}}\sum_{\substack{j\in\hat{K}_{rs}\\ j\neq k}} f_{k(kj)}^{rs} = D_s \quad \forall s\in S \quad \text{(Eq.3.3)}$$

$$f_{k(kj)}^{rs} \geq 0 \ \forall k\in\hat{K}_{rs}, kj\in\hat{L}_{rs}, r\in R, s\in S \quad \text{(Eq.3.4)}$$

It is noted that in CDA-PCL-SUE-R-PL only $f_{k(kj)}^{rs}$ are decision variables, since q_{rs} are substituted by $q_{rs} = \sum_{k\in\hat{K}_{rs}}\sum_{\substack{j\in\hat{K}_{rs}\\ j\neq k}} f_{k(kj)}^{rs}$. I next consider the following equivalent formulation to CDA-PCL-SUE-R-PL, which is the projection of CDA-PCL-SUE-R-PL onto the demand space, in order to solve the problem CDA-PCL-SUE-R-PL.

Equivalent Formulation to CDA-PCL-SUE-R-PL

$$\min U(q) \quad \text{(Eq.4.1)}$$

Subject to

$$\sum_{s\in S} q_{rs} = O_r \quad \forall r \in R \quad \text{(Eq.4.2)}$$

$$\sum_{r\in R} q_{rs} = D_s \quad \forall s \in S \quad \text{(Eq.4.3)}$$

$$q_{rs} \geq 0 \ \forall r \in R, s \in S \quad \text{(Eq.4.4)}$$

Where $U(q) = \tilde{z}_1 + \tilde{z}_2 + \tilde{z}_3$ (Eq.4.5)

Subject to

$$\sum_{k\in \hat{K}_{rs}} \sum_{\substack{j\in \hat{K}_{rs} \\ j\neq k}} f^{rs}_{k(kj)} = q_{rs} \ \forall r \in R, s \in S \quad \text{(Eq.4.6)}$$

$$f^{rs}_{k(kj)} \geq 0 \ \forall k \in \hat{K}_{rs}, kj \in \hat{L}_{rs}, r \in R, s \in S \quad \text{(Eq.4.7)}$$

This equivalent formulation utilizes the fact that the solution to Eq.4.5-4.7 (i.e. the restricted PCL-SUE) is easily obtained by the use of the PCL formula: $f^{rs}_{k(kj)} = P(kj)\cdot P(k\,|\,kj)\cdot q_{rs}$. By performing the substitution of the PCL formula in Eq.4.5 (i.e. $f^{rs}_{k(kj)} = P(kj)_n \cdot P(k\,|\,kj)_n \cdot \ q_{rs}$ and $f^{rs}_{j(kj)} = P(kj)_n \cdot P(j\,|\,kj)_n \cdot q_{rs}$ are substituted in $U(q) = \tilde{z}_1 + \tilde{z}_2 + \tilde{z}_3$), it can be proved that the implicit function $U(q)$ actually has the explicit form of the entropy maximization problem (problem Eq.5.1-5.8 in Phase I.1). Hence, it is clear that CDA-PCL-SUE-R-PL is solved through the solution of the entropy maximization problem followed by the application of the PCL formula. Specifically, an optimal solution to the equivalent formulation of CDA-PCL-SUE-R-PL is obtained by Phases I.1-I.3.

Phase I.1 Entropy Maximization Problem (the solution is denoted by $\underline{q}^n_{rs} \ \forall r \in R, s \in S$)

$$\min \sum_{r\in R}\sum_{s\in S} \gamma^n_{rs} \cdot \underline{q_{rs}} + \nu^n_{rs} \cdot \underline{q_{rs}} \cdot \ln \underline{q_{rs}} \quad \text{(Eq.5.1)}$$

Subject to

$$\sum_{s\in S} \underline{q_{rs}} = O_r \quad \forall r \in R \quad : \alpha_r \quad \text{(Eq.5.2)}$$

$$\sum_{r\in R} \underline{q_{rs}} = D_s \quad \forall s \in S \quad : \lambda_s \quad \text{(Eq.5.3)}$$

$$\underline{q_{rs}} \geq 0 \ \forall r \in R, s \in S \quad \text{(Eq.5.4)}$$

Where

$$\gamma^n_{rs} = \left\{ \begin{array}{l} \displaystyle\sum_{k\in \hat{K}_{rs}} \sum_{\substack{j\in \hat{K}_{rs} \\ j\neq k}} \left[\begin{array}{l} P(kj)_n \cdot P(k\,|\,kj)_n \cdot \\ \left(\begin{array}{l} c^{rs^n}_k + \dfrac{1}{\theta}\beta^{rs}_{kj} \cdot \\ \ln\left(\dfrac{P(kj)_n \cdot P(k\,|\,kj)_n}{\beta^{rs}_{kj}} \right) \end{array} \right) \end{array} \right] \\ + \displaystyle\sum_{k=1}^{|\hat{K}_{rs}|-1} \sum_{j=k+1}^{|\hat{K}_{rs}|} \frac{1}{\theta}(1-\beta^{rs}_{kj}) \cdot P(kj)_n \cdot \ln \frac{P(kj)_n}{\beta^{rs}_{kj}} \end{array} \right\}$$

(Eq.5.5)

$$\nu^n_{rs} = \left\{ \begin{array}{l} \displaystyle\sum_{k\in \hat{K}_{rs}} \sum_{\substack{j\in \hat{K}_{rs} \\ j\neq k}} \frac{1}{\theta}\beta^{rs}_{kj} \cdot P(kj)_n \cdot P(k\,|\,kj)_n \\ + \displaystyle\sum_{k=1}^{|\hat{K}_{rs}|-1} \sum_{j=k+1}^{|\hat{K}_{rs}|} \frac{1}{\theta}(1-\beta^{rs}_{kj}) \cdot P(kj)_n \end{array} \right\}$$

(Eq.5.6)

$$P(kj)_n = \frac{\beta^{rs}_{kj}\left(\exp(-\theta c^{rs^n}_k/\beta^{rs}_{kj}) + \exp(-\theta c^{rs^n}_j/\beta^{rs}_{kj})\right)^{\beta^{rs}_{kj}}}{\displaystyle\sum_{m=1}^{|\hat{K}_{rs}|-1}\sum_{l=m+1}^{|\hat{K}_{rs}|} \beta^{rs}_{ml}\left(\exp(-\theta c^{rs^n}_m/\beta^{rs}_{ml}) + \exp(-\theta c^{rs^n}_l/\beta^{rs}_{ml})\right)^{\beta^{rs}_{ml}}}$$

(Eq.5.7)

$$P(k\,|\,kj)_n = \frac{\exp(-\theta c^{rs^n}_k / \beta^{rs}_{kj})}{\exp(-\theta c^{rs^n}_k / \beta^{rs}_{kj}) + \exp(-\theta c^{rs^n}_j / \beta^{rs}_{kj})}$$

(Eq.5.8)

The entropy maximization problem Eq.5.1-5.8 can be solved by Bregman's balancing method (Lamond and Stewart, 1981; Bregman, 1967), and the result is an auxiliary demand $\underline{q}^n = \{\underline{q}^n_{rs}\}$. The balancing method is briefly described below.

Bregman's Balancing Method

Initialization of Balancing Method:

$\underline{q^0_{rs}} = \exp(-1-\gamma^n_{rs}/\nu^n_{rs}) \quad \forall r \in R, s \in S$

(see *(21)* for the derivation of initial auxiliary O-D flows)

$t \leftarrow 0$ (t is iteration counter for the balancing method)

$i \leftarrow 1$ (i is the constraint counter of the entropy maximization problem)

General Step of Balancing Method (Balancing Constraint i):

Find the unique solution $\underline{q^{t+1}_{rs}}$ and ξ of:

$$\nu^n_{rs} \ln \underline{q^{t+1}_{rs}} - \nu^n_{rs} \ln \underline{q^t_{rs}} - \xi a_{i,rs} = 0 \ \forall r \in R, s \in S \quad \text{(Eq.6)}$$

and $\sum_{rs \in RS} a_{i,rs} \underline{q^{t+1}_{rs}} = b_i$. (Eq.7)

The derivation of Eq.6 and Eq.7 is referred to Bregman (1967). Then, Eq.6 can be written as Eq.8.

$$\underline{q^{t+1}_{rs}} = \underline{q^t_{rs}} \exp\left(\frac{\xi a_{i,rs}}{\nu^n_{rs}}\right) \quad \forall r \in R, s \in S \quad \text{(Eq.8)}$$

which is then substitute into Eq.(7), yielding:

$$\sum_{rs} a_{i,rs} \cdot \underline{q^t_{rs}} \exp\left(\frac{\xi a_{i,rs}}{\nu^n_{rs}}\right) = b_i \quad \text{(Eq.9)}$$

where if $b_i = O_i$ and $\xi = \alpha_i$ if $1 \le i \le |R|$

$b_i = D_i$ and $\xi = \lambda_i$ if $|R|+1 \le i \le |R|+|S|$

ξ is determined by Newton's method, since it cannot be solved analytically.

Then, determine $\underline{q^{t+1}_{rs}}$ from Eq.8:

$i \leftarrow (i \text{ modulo } (|R|+|S|)) + 1$

$t \leftarrow t+1$

For each pass of the algorithm (when all origins and destinations are balanced once), if $\underline{q^t_{rs}}$ is converged, terminate the algorithm.

<u>Phase I.2</u> The solution ($\underline{f}^{rs^n}_{k(kj)}$) of CDA-PCL-SUE-R-PL is obtained by applying the PCL formula:

$$\underline{f}^{rs^n}_k = P(k)_n \cdot \underline{q}^n_{rs} \ \forall k \in K_{rs}, r \in R, s \in S$$

Where

$$P(k)_n = \sum_{\substack{j \in \hat{K}_{rs} \\ j \neq k}} P(kj)_n \cdot P(k \mid kj)_n$$

<u>Phase I.3 (Line Search)</u>

An approximate line search is then made with respect to $z = z_1 + z_2 + z_3$ (the objective function of CDA-PCL-SUE-R) in the (feasible) direction of $\underline{f}^n - f^n$ and $\underline{q}^n - q^n$, resulting in the new solution f^{n+1} and q^{n+1}. Note that $\underline{f}^n$ and $\underline{q}^n$ are the auxiliary solutions to the auxiliary (partially linearized) problem CDA-PCL-SUE-R-PL; whereas f^n and q^n are the current solution to CDA-PCL-SUE-R at iteration *n*. The process is repeated with *n*=*n*+1 until a convergence criterion terminates the solution of CDA-PCL-SUE-R.

3.2. Phase II: Column Generation Problem

The partial linearization algorithm in Phase I solves the restricted master problem, given the subsets of routes between O-D pairs $\hat{K}_{rs} \subseteq K_{rs} \ \forall r \in R, s \in S$. The quality of travel pattern solution obtained from Phase I depends on the quality of $\hat{K}_{rs}$ in approximating K_{rs}. Damberg et al. (1996) suggested and evaluated two route generation strategies based on the calculation of shortest paths given the solution of the restricted master problem. I adopt Damberg et al.'s first route generation strategy for Phase II. Routes are generated from the solution of shortest path problems based on the deterministic travel times; i.e. random components of travel times are temporarily ignored. At the solution to this restricted master problem, the link travel times are updated accordingly, and the subsets $\hat{K}_{rs} \subseteq K_{rs}\ \forall r \in R, s \in S$ are augmented by the generation of new routes using the shortest path algorithm.

It is worth noting that the algorithm is not guaranteed to converge to the unique optimal solution of CDA-PCL-SUE. However, it is guaranteed to solve the restriction of CDA-PCL-SUE to any set of routes generated. In the proposed algorithm, it terminates when the root mean square error of link flows and O-D flows from two successive iterations are within a user-specified tolerance.

4. ILLUSTRATIVE EXAMPLES

The test network is a simple network with five nodes, eight links and four O-D pairs as shown in Figure 1. The Bureau Public Road link cost function is employed:

$$t_a(x_a) = t_a^0\left[1 + \alpha_a\left(\frac{x_a}{s_a}\right)^{\beta_a}\right]$$

The parameters t_a^0, s_a, α_a and β_a are also given in Table 1, and the length of link a is set to t_a^0. Two congestion levels are considered as follows. For higher-congestion level (lower-congestion level), origin demands of origin nodes 1 and 2 are 45 and 50 trips (22 and 25 trips), respectively; destination demands of destination nodes 4 and 5 are 35 and 60 trips (17 and 30 trips), respectively. The employed tolerances are $\varepsilon_{Simplicial} = \varepsilon_{Bregman} = \varepsilon_{Newton} = \varepsilon_{LineSearch} = 0.001$.

The CDA-MNL-SUE results are obtained from the algorithm in Lundgren and Patriksson (1998). The algorithms for both CDA-PCL-SUE and CDA-MNL-SUE are implemented in C. These run on a computer with 1.73 GHz Intel Core i7 processor and 4 GB of RAM, running under Windows 7. The CPU times of all runs on the test network are within 1 minute. I compare the results from CDA-PCL-SUE and CDA-PCL-MNL to examine the effects of congestion, travelers' stochastic perception error and path similarity to simultaneously solve doubly-constrained trip distribution problem and stochastic user equilibrium problem.

The dispersion parameters are set at various values for two congestion levels. The differences in O-D flows and link flows from the two combined distribution and assignment solutions is measured by the root mean square errors:

$$RMSE_L = \sqrt{\frac{1}{|A|}\sum_{a\in A}(x^*_{a,PCL} - x^*_{a,MNL})^2}$$

Where $x^*_{a,PCL}$ and $x^*_{a,MNL}$ are the converged link flows in CDA-PCL-SUE and CDA-MNL-SUE, respectively.

$$RMSE_{OD} = \sqrt{\frac{1}{|RS|}\sum_{rs\in RS}(q^*_{rs,PCL} - q^*_{rs,MNL})^2}$$

Where $q^*_{rs,PCL}$ and $q^*_{rs,MNL}$ are the converged O-D flows in CDA-PCL-SUE and CDA-MNL-SUE, respectively. Figure 2 shows the values of $RMSE_L$ and $RMSE_{OD}$ with various dispersion factors at two congestion levels. $RMSE_{OD}$ appears fluctuated at the higher-congestion level, whereas at the lower-congestion level $RMSE_{OD}$ appears smooth over the dispersion factors. At both congestion levels, $RMSE_L$ decreases with the increase of the dispersion factor. The decrease rate of $RMSE_L$ is greater when the dispersion factor is close to 0, and the decrease rate at the higher-congestion level is greater than that of the lower-congestion level on both networks. Based on the empirical results, the link flow patterns from CDA-PCL-SUE and CDA-MNL-SUE are closer as the dispersion factor increases on both congestion levels. The O-D flow patterns from both models differ in different degree over various dispersion factors.

Since the proposed algorithm employs the column generation phase to generate paths, it is possible that the generated paths from CDA-PCL-SUE are not the same as those from CDA-MNL-SUE. Then, it may not be comparable in terms of route flows. However, I found that the dispersion factor of 0.125 yields the same path set in both models. Thus, this is employed for path flow comparison. Table 2 shows the path flow results obtained from CDA-PCL-SUE and CDA-MNL-SUE at the higher-congestion level. As can be observed in Table 2, the path costs for each O-D pair in both CDA-PCL-SUE and CDA-MNL-SUE are not equal, and both models disperse travel demands to many paths for each O-D pair. These are the effects of travelers' stochastic perception error captured by both models. For each O-D pair, the similarity index is calculated for each route pair connecting this O-D pair. The similarity index of each route pair is completely independent of that of other route pairs. Prashker and Bekhor (2000) indicated that this property is highly desirable for route choice models. Table 2 shows the average similarity index for each route, which is the mean value of all similarity indices involving this route.

CDA-PCL-SUE generally considers a route with a high value of similarity as less attractive in route flow allocation. CDA-PCL-SUE accounts for the overlapping paths in route choice such that a path with a higher value of average similarity index and higher path cost will be assigned less flows. As can be seen in Table 2, in the CDA-MNL-SUE model, the cost of path 3 is 7.76% and 5.83% higher than paths 1 and 2, and assigns less flows to paths 3 (85.05% and 88.34% of flows assigned to paths 1 and 2, respectively). In contrast, CDA-PCL-SUE accounts for the path overlapping effect. The average similarity index of path 3 of O-D 1-4 is 101.09% higher than paths 1 and 2 connecting this O-D pair, and in the CDA-PCL-SUE model the cost of path 3 is 4.09% and 2.68% higher than paths 1 and 2. Then, CDA-PCL-SUE assigns much less flows to path 3 (51.82% and 53.24% of flows assigned to paths 1 and 2, respectively) than CDA-MNL-SUE does.

Table 3 shows the O-D flow results of the two models. Apparently, the O-D flows are distributed differently in the two models. As can be seen in Table 3, the total O-D flows out of each origin in both models are the same, and the total O-D flows into each destination in both models are equal. These are due to the doubly constrained trip distribution embedded in the two models. Table 3 also shows the average similarity index for each O-D pair, which is the mean value of the average similarity indices for all paths connecting this O-D pair. The weighted average path cost for each O-D pair is calculated by the summation of the products of path costs and route choice probabilities. I will explore the results to check whether I can relate the attractiveness of an O-D pair in doubly-constrained O-D trip distribution in CDA-PCL-SUE to the average similarity index for each O-D pair and the weighted average path cost of each O-D pair. I consider the O-D flow distribution for origin node 1. From Table 3, the weighted average path cost of O-D 1-4 in CDA-PCL-SUE is 19.78% higher than that of O-D 1-5, whereas in CDA-MNL-SUE it is 23.09% higher. The average similarity index of O-D 1-4 is 31.26% higher than O-D 1-5. The O-D flows allocated to O-D 1-5 is 14.00% higher than O-D 1-4 in CDA-PCL-SUE, whereas in CDA-MNL-SUE, it is 41.34% higher. It seems that CDA-PCL-SUE may assign more flows to O-D 1-4 with higher similarity index than CDA-MNL-SUE does. Next, I consider the O-D flow distribution for destination node 5. The weighted average path cost of O-D 1-5 in CDA-PCL-SUE is 45.99% higher than that of O-D 2-5, whereas in CDA-MNL-SUE it is 50.06% higher. The average similarity index of O-D 1-5 is 284.71% higher than O-D 2-5. The O-D flows allocated to O-D 2-5 is

50.29% higher than O-D 1-5 in CDA-PCL-SUE, whereas in CDA-MNL-SUE, it is 27.67% higher. In this case, CDA-PCL-SUE assigns less flow to O-D 1-5 with higher similarity index than CDA-MNL-SUE does. Apparently, it cannot be concluded how CDA-PCL-SUE distributes O-D flows among different O-D pairs, given weighted average path cost and average similarity index. This is because CDA-PCL-SUE also has the origin flow balance constraints and destination flow balance constraints that must be satisfied. In fact, the trip distribution in CDA-PCL-SUE can be determined by Eq.(2); i.e. it is based on the path costs, dispersion factor, dual variables of origin and destination flow balance constraints, and similarity indices. The average similarity indices and weighted average path costs are not directly employed in determining the trip distribution.

Table 4 shows the link flow results. The traffic flow patterns are different as the two models have different objective functions used in the trip distribution and route choice to capture the effects of congestion, stochastic perception error and path overlapping. Links with more paths passing through mostly have smaller flows assigned by CDA-PCL-SUE when compared with CDA-MNL-SUE such as links 1, 3, 4, 6 and 8. CDA-PCL-SUE assigns less number of flows to these links than CDA-MNL-SUE does.

5. SUMMARY AND CONCLUSIONS

The enhanced combined trip distribution and traffic assignment formulation is proposed. It combines the doubly-constrained gravity-model based trip distribution and the paired-combinatorial-logit stochastic user equilibrium assignment. The proposed solution method for CDA-PCL-SUE is a disaggregate simplicial decomposition algorithm. A test network with two congestion levels are employed. The results from CDA-PCL-SUE are compared to those from CDA-MNL-SUE in order to illustrate how CDA-PCL-SUE distributes O-D flows and route flows when accounting for similarity effects in addition to the congestion effect and stochastic-perception-error effect. I found that the relationship of O-D flow difference and dispersion factor is unclear, whereas link flow patterns from the two models are more identical at higher dispersion factors. CDA-PCL-SUE assigns less flow to a path with higher average similarity index and higher path cost than CDA-MNL-SUE. CDA-PCL-SUE generally assigns less flow to links with more paths passing through than CDA-MNL-SUE. The relationship between O-D flow allocation and the average similarity indices for O-D pairs is not obvious.

The future research is to include the singly-constrained gravity-based trip distribution version and to incorporate trip generation and modal split.

ACKNOWLEDGMENTS

The author gratefully acknowledges the support from The Thailand Research Fund under contract number MRG5380066 and King Mongkut’s University of Technology Thonburi Research Fund.

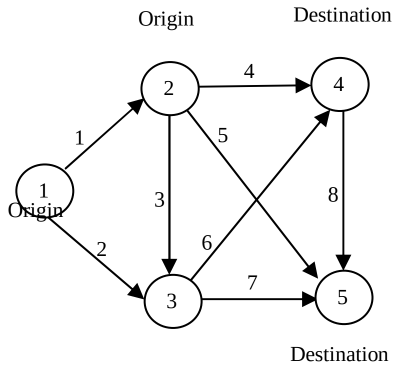


Figure 1 Test Network

Table 1 Parameters of Test Network

Link	s_a	t_a^0	α_a	β_a
(1,2)	25	4.0	0.15	4.0
(1,3)	25	5.2	0.15	4.0
(2,3)	30	1.0	0.15	4.0
(2,4)	15	5.0	0.15	4.0
(2,5)	15	5.0	0.15	4.0
(3,4)	15	4.0	0.15	4.0
(3,5)	15	4.0	0.15	4.0
(4,5)	30	1.0	0.15	4.0

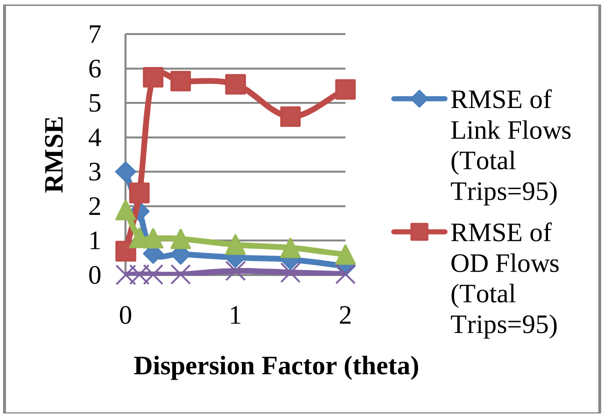


Figure 2 Root Mean Square Errors

Table 2 Path Flow Results of CDA-PCL-SUE and CDA-MNL-SUE (Total O-D Demand = 90 Trips, Dispersion Factor = 0.125)

O-D	Link Seq	Average Similarity Index	Route Choice Probability	
			PCL	MNL
1-4	1-4	0.2198	0.4014	0.3555
	2-6	0.2198	0.3907	0.3422
	1-3-6	0.4420	0.2079	0.3023
1-5	1-5	0.2811	0.2984	0.3250
	2-7	0.0000	0.4415	0.3632
	1-3-6-8	0.3072	0.1231	0.1433
	1-4-8	0.3072	0.1370	0.1685
2-4	4	0.0000	0.5204	0.5404
	3-6	0.0000	0.4796	0.4596
2-5	5	0.0000	0.3600	0.3393
	3-7	0.0609	0.3136	0.3350
	3-6-8	0.1164	0.1455	0.1497
	4-8	0.0556	0.1809	0.1760

O-D	Link Seq	Path Flow		Path Cost	
		PCL	MNL	PCL	MNL
1-4	1-4	8.43	6.62	15.95	16.68
	2-6	8.21	6.38	16.17	16.98
	1-3-6	4.37	5.63	16.61	17.97
1-5	1-5	7.17	8.56	12.17	12.45
	2-7	10.59	9.57	12.18	11.56
	1-3-6-8	2.93	3.77	17.63	19.00
	1-4-8	3.26	4.44	16.97	17.70
2-4	4	7.27	8.83	11.23	11.58
	3-6	6.70	7.51	11.88	12.88
2-5	5	12.99	11.41	7.45	7.36
	3-7	11.31	11.27	7.89	7.46
	3-6-8	5.22	5.03	12.90	13.90
	4-8	6.49	5.92	12.25	12.61

Table 3 O-D Flow Results of CDA-PCL-SUE and CDA-MNL-SUE (Total O-D Demand = 90 Trips, Dispersion Factor = 0.125)

O-D	Average Similarity Index	Weighted Average Path Cost		O-D Flow	
		PCL	MNL	PCL	MNL
1-4	0.2939	16.18	17.17	21.02	18.64
1-5	0.2239	13.50	13.95	23.97	26.35
2-4	0.0000	11.55	12.18	13.97	16.35
2-5	0.0582	9.25	9.30	36.02	33.64

Table 4 Link Flow Results of CDA-PCL-SUE and CDA-MNL-SUE
(Total O-D Demand = 90 Trips,
Dispersion Factor = 0.125)

Link	Number of Paths Passing Through*			
	O-D 1	O-D 2	O-D 3	O-D 4
1	2	3	0	0
2	1	1	0	0
3	1	1	1	2
4	1	1	1	1
5	0	1	0	1
6	2	1	1	1
7	0	1	0	1
8	0	2	0	2

Link	Link Flow		Link Cost	
	PCL	MNL	PCL	MNL
1	26.19	29.04	4.72	5.09
2	18.80	15.95	5.45	5.32
3	30.55	33.24	1.16	1.22
4	25.47	25.82	11.23	11.58
5	20.16	19.98	7.45	7.36
6	27.44	28.35	10.72	11.65
7	21.91	20.84	6.73	6.23
8	17.92	19.17	1.01	1.02

REFERENCES

Bregman, L.M. (1967) *The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming*, U.S.S.R. Computational Mathematics and Mathematical Physics, Vol.7, 200-217.

Chen, A., P. Kasikitwiwat, and Z. Ji (2003) *Solving the overlapping problem in route choice with paired combinatorial logit model*, *Transportation* Research Record 1857, 65-73.

Daganzo, C.F. and Y. Sheffi (1977) *On Stochastic Models of Traffic Assignment*, Transportation Science, Vol.11, 253-274.

Damberg, O., J.T. Lundgren and M. Patriksson (1996) *An algorithm for the stochastic user equilibrium problem*, Transportation Research Part B, Vol.30, 115-131.

Erlander, S. (1990) *Efficient behavior and the simultaneous choices of origins, destinations and routes*, Transportation Research Part B, Vol.24, 363-373.

Evans, S.P. (1976) *Derivation and analysis of some models for combining trip distribution and assignment*, Transportation Research, Vol.10, 37-57.

Lamond, B. and N.F. Stewart (1981) *Bregman's Balancing Method*, Transportation Research Part B, Vol.15, No.4, 239-248.

Larsson, T. and M. Patriksson (1992) *Simplicial decomposition with disaggregated representation for the traffic assignment problem*, Transportation Science, Vol.26, 4-17.

Lundgren, J.T. and M. Patriksson (1998) *The combined distribution and stochastic assignment problem*, Annals of Operations Research, Vol.82, 309-329.

Prashker, J.N. and S. Bekhor (2000) *Congestion, Stochastic, and Similarity Effects in Stochastic User-Equilibrium Models*, Transportation Research Record 1733, 80-87.

Patriksson, M. (1993) *Partial linearization methods in nonlinear programming*, Journal of Optimization Theory and Applications, Vol.78, 227-246.

Service Life Analysis and Maintenance Program of Pavement Markings in Thailand

Ponlathep Lertworawanich and Ampol Karoonsoontawong

Pavement markings are one of the most important highway assets. Good pavement markings provide good visibility for traffic, whereas poor pavement markings can adversely affect traffic operations on highways. However, a limited amount of research has been conducted on the effects of working conditions on the service life of pavement markings. This paper presents duration models for retroreflectivity of thermoplastic pavement markings in Thailand. This approach allows the service life of pavement markings to follow probability distributions in which model parameters are assumed as a function of relevant independent variables such as traffic volumes. The maximum likelihood estimation technique was used to estimate means and standard errors of the model parameters. Retroreflectivity data of thermoplastic pavement markings were collected from the eastern highway network of Thailand, which consists of more than 5,000 km of highways in various traffic conditions. The analysis results showed that traffic volumes had negative effects on the service life of the pavement markings. This paper proposes a preemptive goal program for approximating required budgets to ensure the maintenance of the percentages of good condition pavement markings over the planning horizon. The first-priority goal is to maintain the percentage of control sections that are in good condition, and the second-priority goal is to minimize total maintenance costs. In the illustrative example, the inconsistency between the Thailand Department of Highways' specifications and field practices caused estimated required annual budgets and the deterioration of pavement markings to greatly fluctuate over the planning period. For more consistency between specification and field practice, the proposed models will be applied in a pavement marking management system.

Driving safely at night requires highway markings to be reflective to help drivers navigate in a low-visibility condition. Retroreflectivity is the property that measures the ability of the marking to reflect the light from the headlamp back toward the driver's eyes. Good visibility of pavement markings is needed because it helps reduce the likelihood of traffic accidents, as discussed in Smadi et al. (*1*). Retroreflectivity is obtained by dropping glass beads on the top of marking materials. Good quality control in painting operations with the right proportion of painting materials and glass beads results in high retroreflective and long-lasting pavement markings. The ability to predict the service life of pavement markings is crucial for maintenance management and operations. Many researchers have studied the retroreflectivity properties to explain the causes of deterioration and suggest service life. Lee et al. (*2*) used the 15-m geometry device to collect retroreflectivity data on 50 test sites and evaluated the performance of marking materials with linear regression analysis. The accuracy of the prediction models was relatively low with the R^2 varying between .14 and .18. Snowplow activity was the most significant factor while annual average daily traffic (AADT), speed limit, and percentage of heavy vehicle traffic did not correlate with the retroreflectivity degradation. Abboud and Bowman (*3*) studied the service life, cost of application, and cost associated with crashes related to marking retroreflectivity and proposed logarithmic regression models to predict the degradation curves. The unique aspect of the model was the use of vehicle exposure variables, defined by AADT and age of markings. This model did not depend on types of road surfaces, which have been used as an independent variable by many researchers. In Sarasua et al. (*4*), the South Carolina Department of Transportation sponsored the joint research between Clemson University and the Citadel to study the effective life cycle of pavement marking retroreflectivity. The data were collected with the 30-m geometry on 150 sites throughout the South Carolina interstate system. Traffic wearing was initially thought to cause degradation in retroreflectivity; however, the statistical test showed nonsignificance, so this variable was dropped from the models. Their final degradation models were presented by the difference in retroreflectivity over time. In Kopf (*5*), Washington State Transportation Center used the mobile retroreflectometer with the 30-m geometry to collect field data of waterborne and solvent-based paints on 80 sites throughout the state. Data were categorized by marking materials, colors, levels of traffic, and geography into several groups and analyzed with linear or logarithm models. Some models showed poor prediction because of a large variation in data, whereas some models showed a high value of R^2, but the data points were either too few or not well spread throughout the service lifetime. The recent study by Sitzabee (*6*) and Sitzabee et al. (*7*) provided a comprehensive degradation model for a variety of pavement markings. The project included data from 30,000 mi (48,000 km) of road throughout the state of North Carolina. A large number of data were categorized into several groups and the group that had complete information was analyzed by linear regression analysis. The degradation models for the paints depended on the initial value of retroreflectivity ($R_{L,\text{int}}$) and time, whereas the models for thermoplastics depended on $R_{L,\text{int}}$, time, AADT, lateral locations of markings, and colors. In addition to the pavement marking literature, a review was conducted on a related subject, state transition probabilities of bridge decks. Mishalani and Madanat (*8*) studied transition probabilities of infrastructures using stochastic

P. Lertworawanich, Department of Highways, Bureau of Road Research and Development, Sri-Ayudhaya Road, Ratchatevi, Bangkok 10400, Thailand. A. Karoonsoontawong, Department of Civil Engineering, King Mongkut's University of Technology Thonburi, 126 Pracha-Utit Road, Bangmod, Thung-Khru, Bangkok, Thailand. Corresponding author: P. Lertworawanich, ponla.le@doh.go.th or ponlathep@psualum.com.

Transportation Research Record: Journal of the Transportation Research Board, No. 2272, Transportation Research Board of the National Academies, Washington, D.C., 2012, pp. 121–129.
DOI: 10.3141/2272-14

duration models. In their study, the distributions of time intervals between state transitions were computed. Assuming a Weibull distribution, relevant model parameters were estimated with maximum likelihood estimation (MLE) where the hazard rates were found to be an increasing function. This indicates that the aging process reduces the service life of bridge decks. Expectation of state transition time intervals can also be predicted from their model.

In summary, most studies use ordinary linear regression models to fit the relation between the retroreflectivity of pavement markings and the relevant independent variables. These models cannot provide good fits with data. The service life should itself be a prime factor of interest. If an ordinary regression model is fitted to the measured retroreflectivity of markings, pavement markings with the retroreflectivity below the minimum requirement are normally omitted from consideration. This omission leads to a truncation bias. Therefore, a new modeling approach rather than ordinary regression models should be investigated. In recent years, the Department of Highways (DOH) in Thailand has increased awareness of pavement marking performance with the aim of improving road safety. The DOH has a specification on thermoplastic pavement markings that all yellow markings should have a minimum retroreflectivity (R_L) of 100 millicandelas per lux per square meter (mcd/lx/m^2) and all white markings should have a R_L of 150 mcd/lx/m^2 with the 15-m geometry measurement, a former ASTM E1710 standard. The DOH also requires that all thermoplastic markings have at least 2 years of service life. In 2008, the DOH launched a pilot project to conduct a field survey of pavement markings. The survey aimed at assessing marking conditions in the eastern part of Thailand and establishing a pavement marking database for future research. In this paper, the database from this pilot project is analyzed to examine the service life of thermoplastic pavement markings and to study the effects of traffic conditions on thermoplastic pavement markings. In addition, a pavement markings budgeting module is also developed to optimally approximate the required budget to maintain pavement markings in a good condition. As a result, the objectives of this study are

- To collect the retroreflectivity of thermoplastic pavement markings in field conditions,
- To develop a service life model of thermoplastic pavement markings based on the duration modeling, and
- To develop a pavement markings budgeting module based on an optimization formulation.

In the remainder of this paper, the data collection scheme is presented followed by the proposed methodology to predict the service life of the thermoplastic pavement markings based on duration models in which the MLE technique is utilized to estimate parameter values. These models can be used to predict the service life of the thermoplastic pavement markings. Next, an optimization formulation is proposed to approximate the required budget to optimally maintain pavement markings during the planning horizon and to search for an optimal maintenance pattern during the planning horizon. The paper closes with conclusions and recommendations for future research.

FIELD DATA COLLECTION

In this section, an outline of the field data collection scheme is briefly provided. The eastern highway network of Thailand consists of 5,000 km of highways, most of which are multilane highways.

FIGURE 1 Location of study area.

The study area consists of five provinces located east of Bangkok. Figure 1 shows the location of the study area in Thailand.

To collect data, this study employs a retroreflectometer, ZRM6013, capable of measuring R_L and Q_d (day visibility) with the 30-m geometry according to the ASTM E1710-05 standard. However, for a comparison to be made with the DOH standard, which is based on the 15-m geometry of measurement for R_L, an experiment to find the relationship between the 15-m geometry and the 30-m geometry measurements was set up. In this study, the DOH uses two devices: (*a*) the ZRM6013 for the 30-m geometry and (*b*) the Mirolux for the 15-m geometry, to measure several markings on the same locations. Then, a simple regression is conducted to find a relation as shown in Figure 2.

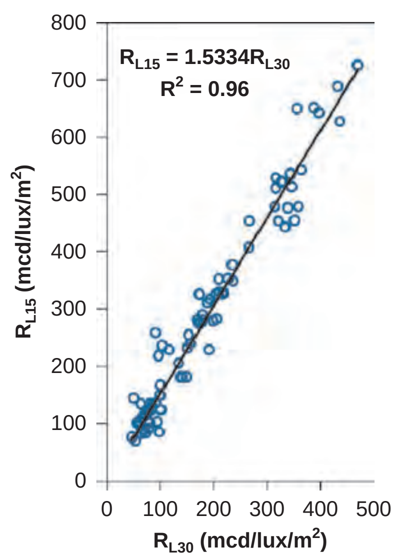
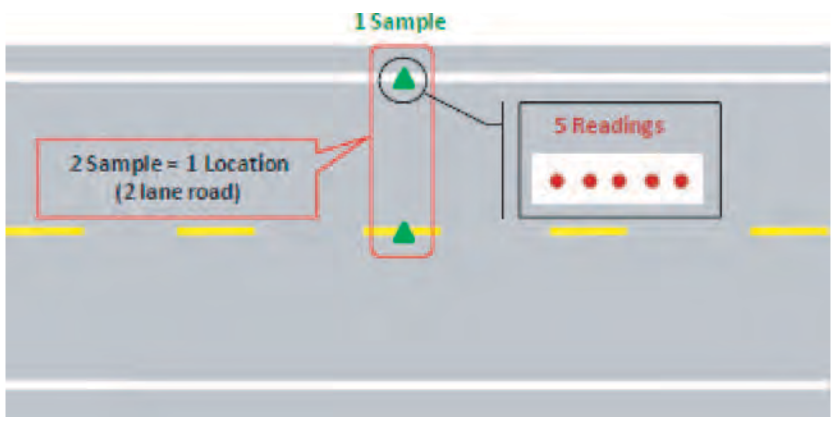


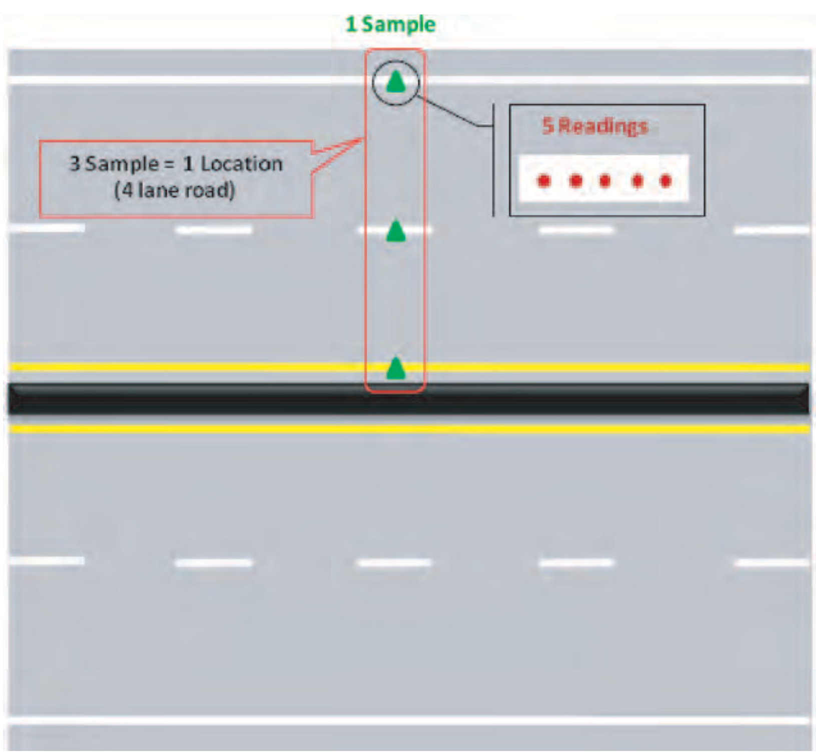
FIGURE 2 Relationship between 15- and 30-m geometry measurements.

From Figure 2, it is found that the 15-m geometry measurement is approximately 1.53 times the 30-m geometry measurement. This number is used to convert the values of the 30-m geometry measurements to the equivalent 15-m geometry ones so that the comparison to the DOH can be made. The ZRM6013 device can provide both 15-m and 30-m geometry values of retroreflectivity. In the data collection scheme, there must be at least one sampling for every 100 m^2 of the markings. In the case of two-lane highways, samples are collected at an interval of 800 m, with R_L of markings measured at two locations, the shoulder line (white line) and the direction-separating line (yellow lines), as shown in Figure 3*a*. For highways of four or more lanes, samples are collected at an interval of 400 m and R_L of markings is measured at three locations, the shoulder line (white line), the lane-separating line (white dotted line), and the median line (yellow line), as shown in Figure 3*b*.

One measurement is the average of the five readings from the same location. In total, the number of measurements comes up to 5,000 samples through the course of the data collection scheme. The geographic information system (GIS) coordinates of each location are also collected to represent the GIS map of the study area. In addition, other relevant prevailing information is collected, such as AADT, percentage of heavy vehicles, and number of lanes at the measurement sites. This information will be used as explanatory variables in the service life models of the pavement markings. The data set consists of several variables as summarized in Table 1.



(a)



(b)

FIGURE 3 Data collection locations for (*a*) two-lane highways and (*b*) highways of four or more lanes.

TABLE 1 Analysis Variables

Variable	Description
AADT_LANE	Annual average daily traffic per lane (vehicles/day/lane)
PHV	Percentage of heavy vehicles (percent)
R_{L15}	Marking retroreflectivity measured with 15-m instrument (mcd/lx/m^2)
Age	Age of pavement marking at the time of observation (days)
Surf	Wearing surface type: 1 = concrete, 0 = asphalt

PAVEMENT MARKING SERVICE LIFE MAINTENANCE MODELS

In this section, the methodology to represent the service life of the thermoplastic pavement markings is presented based on the duration modeling analysis. The service life of the thermoplastic pavement markings is assumed to follow a certain distribution in which the distribution parameters are a function of relevant field variables such as traffic volumes. The optimal parameters are estimated with the MLE. This section consists of two parts. The first part presents details of the MLE of the service life distribution parameters and the second part presents an application of the model to estimate the field data parameters.

MLE of Service Life Model

In this research, the service life of the pavement markings is assumed to follow a Weibull distribution in which the distribution parameters are a function of traffic volume per lane and other relevant variables at each site. The Weibull distribution is selected because it has widely been used to model service life distribution of pavements and bridges as mentioned in Mishalani and Madanat (*8*). The service life probability density function (pdf) can be expressed as

$$f_T(t) = p\lambda^{-p}t^{p-1}e^{-\left(\frac{t}{\lambda}\right)^p} \qquad t \geq 0 \tag{1}$$

where

$f_T(t)$ = pdf of the service life of pavement markings;
λ = $e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots+\beta_K X_K}$, scale parameter of Weibull distribution;
β_i = model parameters to be estimated;
X_i = exogenous variables; and
p = shape parameter of Weibull distribution to be estimated.

The mean and the variance of the Weibull distribution can be expressed as

$$E(T) = \lambda \cdot \Gamma\left(1 + \frac{1}{p}\right)$$

and

$$\mathrm{Var}(T) = \lambda^2\left\{\Gamma\left(1+\frac{2}{p}\right) - \Gamma^2\left(1+\frac{1}{p}\right)\right\} \tag{2}$$

where T is a random variable representing the service life of pavement markings and $\Gamma(z)$ is the gamma function,

$$\int_0^\infty t^{z-1}e^{-t}dt$$

When $p = 1$, the distribution becomes a negative exponential distribution. When $0 < p < 1$, the hazard rate is a decreasing function. When $p > 1$, the hazard rate is an increasing function. The hazard rate function is the conditional probability that a pavement marking will fail (its retroreflectivity is below the minimum requirement) between time t and $t + dt$, given that the pavement marking has not failed up to time t. The hazard rate function can be defined as follows:

$$h(t) = \frac{f_T(t)}{1 - F_T(t)} \tag{3}$$

where $h(t)$ is the hazard rate function of pavement markings, and $F_T(t)$ is the cumulative distribution function (cdf) of pavement markings.

Given the field data observations, there are two categories of data:

1. Data from the markings that have already failed at the time of data collection. This means that the retroreflectivity of the markings is below the minimum requirement at the time when the measurements were made. This type of data is called failed category.
2. Data from the markings that have not yet failed at the time of data collection. This means that the retroreflectivity of the markings is above the minimum requirement at the time when the measurements were made. This type of data is called the not failed category.

To use the MLE technique to estimate the model parameters for the retroreflectivity data from field collection, the likelihood function is specified. Because the data come from two categories, the likelihood of each observation can be expressed into two groups as shown in Equations 4 and 5.

Category 1 or failed category:

$$P\left[T < t_i\right] = \int_0^{t_i} f_T(t)dt = 1 - e^{-\left(\frac{t_i}{\lambda}\right)^p} = 1 - e^{-t_i^p \cdot \left[e^{\beta_0+\beta_1 X_1^i+\beta_2 X_2^i+\cdots+\beta_K X_K^i}\right]^{-p}} \tag{4}$$

Equation 4 represents the probability that the service life of the marking observation i is less than its age at the time of observation ($P[T < t_i]$). In other words, it is the probability that the i observation is in the failed category at the time of observation.

Category 2 or not failed category:

$$P\left[T > t_i\right] = \int_{t_i}^{\infty} f_T(t)dt = e^{-\left(\frac{t_i}{\lambda}\right)^p} = e^{-t_i^p \cdot \left[e^{\beta_0+\beta_1 X_1^i+\beta_2 X_2^i+\cdots+\beta_K X_K^i}\right]^{-p}} \tag{5}$$

Equation 5 represents the probability that the service life of the marking observation i is more than its age at the time of observation ($P[T > t_i]$). In other words, it is the probability that the i observation is in the not failed category at the time of observation.

As a result, the log-likelihood (LogL) function can be expressed as follows:

$$\mathrm{LogL}(\beta_0,\beta_1,\beta_2 \ldots,\beta_K) = \sum_{i=1}^{N_1}\ln\left[1 - e^{-t_i^p \cdot \left[e^{\beta_0+\beta_1 X_1^i+\beta_2 X_2^i+\cdots+\beta_K X_K^i}\right]^{-p}}\right] + \sum_{i=1}^{N_2}\ln\left[1 - e^{-t_i^p \cdot \left[e^{\beta_0+\beta_1 X_1^i+\beta_2 X_2^i+\cdots+\beta_K X_K^i}\right]^{-p}}\right] \tag{6}$$

where

$\text{LogL}(\beta_0, \beta_1, \beta_2, \ldots, \beta_K)$ = log-likelihood function,
N_1 = number of observations in failed category,
N_2 = number of observations in not failed category,
β_i = model parameter to be estimated,
t_i = age of the pavement marking observation i, and
X_j^i = value of the j independent variable of the observation i.

To obtain the mean values of each parameter, the log function is differentiated with respect to each model parameter and is equated to zero. The covariance matrix of the model parameters is estimated from the inverse of the negative of the Hessian matrix of the log-likelihood function.

Application of Service Life Model to Field Data

In this section, the proposed methodology is applied to the collected data to obtain the model parameters that can be used to estimate the mean service life of the thermoplastic pavement markings. There are three types of pavement markings investigated: (*a*) white shoulder (solid) lines, (*b*) white lane-separating (dotted) lines, and (*c*) yellow lines. On the application of the proposed MLE to estimate the service life, the following results are found.

The significant variables are shown in Table 2. The AADT per lane is the most important factor contributing to deterioration of pavement markings. Obviously, AADT per lane has a negative effect on the service life of pavement markings. As the traffic volume increases, the service life of pavement markings decreases as a result of wear and tear from traffic. The results indicate that the service life of yellow lines is not significantly affected by the amount of traffic. Yellow lines are used at medians to separate traffic directions and drivers in Thailand are required to drive on the shoulder lane except for passing maneuvers; therefore, traffic has no effect on the yellow lines. The mean service life can also be estimated from the MLE results. The mean of the Weibull distribution can be calculated with Equation 2. The average service lives of pavement markings are estimated by expressions in the following table and represented in Figure 4:

TABLE 2 MLE of Service Life of Thermoplastic Pavement Markings

Variable	Coefficient	Standard Error	*t*-Statistic	*p*-Value
MLE Results for Thermoplastic White Shoulder (Solid) Lines[a]				
Intercept	6.1534	0.1577	39.01	.000
AADT_LANE	-2.38×10^{-4}	0.13×10^{-4}	−5.64	.000
p (shape)	0.3476	0.0431	8.06	.000
MLE Results for Thermoplastic White Lane-Separating (Dotted) Lines[b]				
Intercept	5.8416	0.2781	21.01	.000
AADT_LANE	-2.30×10^{-4}	5.72×10^{-5}	−4.01	.000
p (shape)	0.3309	0.0583	5.68	.000
MLE Results for Thermoplastic Yellow Median Lines[c]				
Intercept	6.9740	1.040	6.71	.000
p (shape)	0.527593	0.0385	13.704	.000

[a]Censoring information: number in not failed state, 341; number in failed state, 1,311; log-likelihood = −765.264.
[b]Censoring information: number in not failed state, 134; number in failed state, 687; log-likelihood = −326.374.
[c]Censoring information: number in not failed state, 768; number in failed state, 995; log-likelihood = −1,168.349.

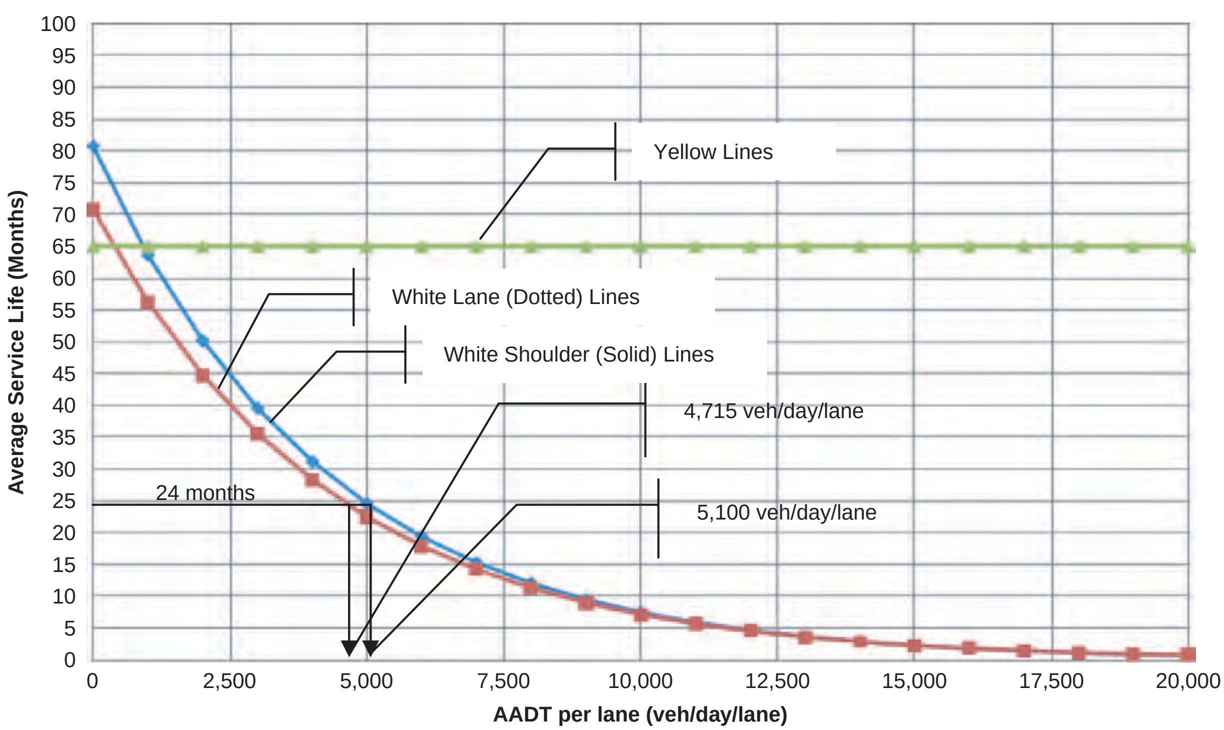


FIGURE 4 Average service life of thermoplastic markings in Thailand (veh = vehicles).

Marking Type	*Average Service Life (months)*
White shoulder line	$(5.152 \times e^{6.1534-2.38\times10^{-4}\ \mathrm{AADT\_LANE}})/30$
White lane-separating line	$(6.169 \times e^{5.842-2.2951\times10^{-4}\ \mathrm{AADT\_LANE}})/30$
Yellow median line	65

For thermoplastic white shoulder (solid) lines, the average service life is less than 24 months in most traffic conditions, except for low-AADT conditions [less than 5,100 vehicles per day per lane (vpdpl)]. Similarly for white lane-separating (dotted) lines, all of them have less than 24 months of service life by average, except for low AADT (less than 4,715 vpdpl). They also possess the lowest service life among the different marking types, because their installation location is subjected to lane-changing traffic. It is noteworthy that the service lives of white shoulder (solid) lines and white lane-separating (dotted) lines are almost alike for high traffic volumes (above 7,500 vpdpl). Yellow lines can provide an average service life of more than 24 months, as required by the DOH specification, because their installation location near medians is away from traffic. In other words, the installation location of the markings has a significant effect on the service life of the markings. This finding is relevant to the findings of Sitzabee (*6*) and Sitzabee et al. (*7*), which indicate that the lateral location of markings affects the service life of the markings. In addition, the DOH specification on yellow thermoplastic markings is less restrictive than that of white thermoplastic markings, which requires that yellow thermoplastic markings should have a minimum R_L of 100 mcd/lx/m^2 and white thermoplastic markings should have a minimum R_L of 150 mcd/lx/m^2 with the 15-m geometry measurement.

PAVEMENT MARKING MAINTENANCE OPTIMIZATION MODEL

Pavement marking maintenance is an important activity. Every year the DOH spends a lot of money on pavement markings. As a result, there is a need for an optimization program to estimate required annual budgets for a given percentage of control sections with retroreflectivity above the minimum DOH specification and to optimally allocate budgets to each highway control section. The DOH uses the jurisdiction boundary to decide the extent of each control section and most of these sections are less than 10 km in length. The optimization model is formulated as a mixed-integer program with the following model assumptions:

- The planning horizon is 4 years in length. Each year consists of four periods or quarters as shown in Figure 5.

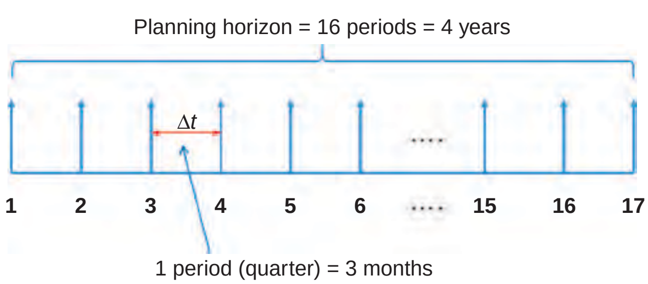


FIGURE 5 Planning horizon.

- The maintenance decision is made at the beginning of each period and each period is 3 months in length (Δt).
- The service lives and remaining lives are integral units of a period length. Remaining lives of pavement markings are known at the beginning of the planning horizon.
- In the same control section, all markings are simultaneously repainted. Remaining lives and service lives of markings are determined from the minimum remaining lives and service lives of all markings in the same control section.
- The warranty period of thermoplastic paints is 2 years. Pavement markings that are in the warranty period cannot be repainted.

The proposed pavement marking maintenance optimization model is described, followed by a case study.

Description of Proposed Budget Approximation Optimization Model

The budget approximation module is used to approximate required budgets in each year of the planning horizon. The module is formulated as a preemptive goal program to minimize the total maintenance cost while maintaining percentage of control sections with retroreflectivity above the minimum DOH requirement. Generally, the approach of goal programming is to establish a specific numeric goal or an aspiration level for each of the objectives, to formulate an objective function for each objective, and then to seek a solution that minimizes the (weighted) sum of deviations of these objective functions from their respective goals. There are three possible types of goals:

1. A lower one-sided goal sets a lower limit that one does not want to fall below.
2. An upper one-sided goal sets an upper limit that one does not want to exceed.
3. A two-sided goal sets a specific target that one does not want to miss on either side.

Goal programs can be categorized according to how goals compare in importance. In one case, called nonpreemptive goal programming, all goals are roughly of comparable importance. In another case, called preemptive goal programming, there is a hierarchy of priority levels for the goals, so that the goal of primary importance receives first-priority attention, that of secondary importance receives second-priority attention, and so forth if there are more than two priority levels. For further information on goal programs, see Hillier and Lieberman (*9*) and Sherali and Soyster (*10*). The proposed goal program will provide a compromising solution of maintenance budget. The first-priority goal is to maintain the percentage of control sections that are in good condition and the second-priority goal is to minimize total maintenance cost. The parameters and decision variables are first defined, followed by the proposed budget approximation formulation.

Parameters

The parameters are defined as follows:

N = total number of control sections;
T = total number of time periods;

$C_{i,t}$ = pavement marking maintenance cost of control section *i* in time period *t;*
P_{cr} = percentage of control sections in good condition in each time period;
S_i = service life of new pavement marking of control section *i;*
$\text{Warr}_{i,t}$ = 1 if control section *i* is still on a warranty in time period *t*, 0 otherwise;
M = sufficiently large positive number (e.g., 100); and
$z0_i$ = initial remaining life (in time period unit) of control section *i*.

Decision Variables

The decision variables are defined as follows:

$d\cos t^-$, $d\cos t^+$ = downside and upside deviation from the aspiration target of zero maintenance cost;
d_t^-, d_t^+ = downside and upside deviation from the aspiration target of proportion of control sections in good condition (i.e., with retroreflectivity above the minimum requirement) in time period *t;*
$x_{i,t}$ = 1 if control section *i* is selected for maintenance in time period *t*, 0 otherwise;
$y_{i,t}$ = 1 if control section *i* has its remaining life greater than or equal to 1 time period at time period *t* or control section *i* is selected for maintenance at time period *t*, 0 otherwise;
$z_{i,t}$ = remaining life (in time period unit) of control section *i* at the beginning of time period *t;* and
$\delta_{i,t}$ = 1 if $z_{i,t} \le 1$ and 0 if $z_{i,t} \le 0$.

Formulation

The proposed budget approximation formulation is

$$\text{lex min}\left\{zz_1; zz_2\right\} \tag{7}$$

where lex min is the lexicographic minimization and

$$zz_1 = \sum_{t=1}^{T} d_t^- \tag{7.1}$$

$$zz_2 = d\ \cos t^+ \tag{7.2}$$

subject to

$$z_{i,t} \le M\delta_{i,t} \qquad \forall i = 1, 2, \ldots, N; \forall t = 1, 2, \ldots, T \tag{8}$$

$$z_{i,t} \ge 1 - M\left(1 - \delta_{i,t}\right) \qquad \forall i = 1, 2, \ldots, N; \forall t = 1, 2, \ldots, T \tag{9}$$

$$z_{i,t+1} \le z_{i,t} - 1 + M\left(1 - \delta_{i,t}\right) \qquad \forall i = 1, 2, \ldots, N \quad \forall t = 1, 2, \ldots, T-1 \tag{10}$$

$$z_{i,t+1} \ge z_{i,t} - 1 - M\left(1 - \delta_{i,t}\right) \qquad \forall i = 1, 2, \ldots, N \quad \forall t = 1, 2, \ldots, T-1 \tag{11}$$

$$z_{i,t+1} \le S_i x_{i,t} - 1 + M\delta_{i,t} \qquad \forall i = 1, 2, \ldots, N \quad \forall t = 1, 2, \ldots, T-1 \tag{12}$$

$$z_{i,t+1} \ge S_i x_{i,t} - 1 - M\delta_{i,t} \qquad \forall i = 1, 2, \ldots, N \quad \forall t = 1, 2, \ldots, T-1 \tag{13}$$

$$\sum_{t=1}^{T}\sum_{i=1}^{N} C_{i,t} \cdot x_{i,t} + d\cos t^- - d\cos t^+ = 0 \tag{14}$$

$$\frac{100}{N} \cdot \sum_{i=1}^{N} y_{i,t} + d_t^- - d_t^+ = P_{cr} \qquad \forall t = 1, 2, \ldots, T \tag{15}$$

$$\sum_{m=t}^{t+7} x_{i,m} \le 1 \qquad \forall i \in = 1, 2, \ldots, N; \forall t = 1, 2, \ldots, T-7 \tag{16}$$

$$y_{i,t} \le x_{i,t} + \delta_{i,t} \qquad \forall i = 1, 2, \ldots, N; \forall t = 1, 2, \ldots, T \tag{17}$$

$$x_{i,t} \le 1 - \text{Warr}_{i,t} \qquad \forall i = 1, 2, \ldots, N; \forall t = 1, 2, \ldots, 7 \tag{18}$$

$$z_{i,1} = z0_i \qquad \forall i = 1, 2, \ldots, N \tag{19}$$

$$z_{i,t} \text{ integer} \qquad \forall i = 1, 2, \ldots, N; \forall t = 1, 2, \ldots, T \tag{20}$$

$$y_{i,t} \in \left\{0, 1\right\} \qquad \forall i = 1, 2, \ldots, N; \forall t = 1, 2, \ldots, T \tag{21}$$

$$x_{i,t} \in \left\{0, 1\right\} \qquad \forall i = 1, 2, \ldots, N; \forall t = 1, 2, \ldots, T \tag{22}$$

$$\delta_{i,t} \in \left\{0, 1\right\} \qquad \forall i = 1, 2, \ldots, N; \forall t = 1, 2, \ldots, T \tag{23}$$

$$d_t^- \ge 0, d_t^+ \ge 0 \qquad \forall t = 1, 2, \ldots, T \tag{24}$$

$$d\cos t^- \ge 0, d\cos t^+ \ge 0 \tag{25}$$

The objective functions in Equations 7.1 and 7.2 are minimized in lexicographical order, yielding two optimization programs that are solved in a sequence. First, the objective (Equation 7.1) is minimized subject to constraints (Equations 8 to 25). The aspiration level in the first program is the specified percentage (P_{cr}) of control sections with retroreflectivity above the minimum requirement. After results are obtained from the first optimization program, the optimal downside deviations from the desired percentage of good condition control sections (d_t^-*) are then fixed in the second program. The second program minimizes the objective (Equation 7.2) subject to constraints (Equations 8 to 25) and an additional constraint set (Equation 26):

$$d_t^- \le d_t^{-*} \qquad \forall t = 1, 2, \ldots, T \tag{26}$$

Constraint 26 implies that while minimizing total maintenance costs, no larger downside deviation from the aspiration level is retained. The aspiration level in the second program is the total cost of zero (see Constraint 14), which means that one wants to minimize the total maintenance costs. Constraints 8 to 9 enforce that $\delta_{i,t} = 1$ when the remaining service life is at least one time period and $\delta_{i,t} = 0$ otherwise (i.e., $\delta_{i,t} = 0$ if and only if $z_{i,t} \le 0$; otherwise $\delta_{i,t} = 1$ if and only if $z_{i,t} \le 1$). Constraints 10 to 13 determine the remaining service life in the next time period based on the remaining service life and the maintenance decision at the current time period (i.e., $z_{i,t+1} = z_{i,t} - 1$ if $\delta_{i,t} = 1$, and $z_{i,t+1} = x_{i,t}\, S_i - 1$ if $\delta_{i,t} = 0$).

Equation 14 implies that total maintenance cost is equal to $d\cos t^+ - d\cos t^-$. Because Constraint 25 indicates that $d\cos t^+$

and $d\cos t^-$ are nonnegative and the objective minimizes $d\cos t^+$, the optimal $d\cos t^+$ is the total maintenance cost, and $d\cos t^-$ equals 0. That is, Equation 14 is an upper one-sided goal that sets an upper limit of zero maintenance cost that one does not want to exceed, and the upside deviation from this goal is total maintenance cost. Equation 15 represents a lower one-sided goal constraint on percentage of control sections with retroreflectivity above the minimum requirement for each period. One does not want to fall under the desired percentage of control sections in good condition in each time period. Equation 16 guarantees that no control section is repainted within 2 years, which is a warranty period of thermoplastic markings. Equation 17 and the objective (Equation 7) determine the variables $y_{i,t}$ as follows. If control section i is not selected for maintenance at time period t (i.e., $x_{i,t} = 0$) and its remaining life is less than or equal to 0 (i.e., $\delta_{i,t} = 0$), then Equation 17 forces $y_{i,t}$ to 0, implying that the control section i is not in good condition at time t. If control section i is selected for maintenance at time period t (i.e., $x_{i,t} = 1$) or its remaining life is greater than 0 (i.e., $\delta_{i,t} = 1$), then the model will force $y_{i,t}$ to 1, implying that the control section i is in good condition at time t. Equation 18 states that the maintenance of the control sections from time periods 1 to 7, which are under warranty because of the repaint before the analysis period, is forbidden. Equation 19 set up the initial remaining service life for each control section. Constraint 20 is an integer constraint for variables $z_{i,t}$. Equations 21 to 23 are binary constraints for variables $y_{i,t}$, $x_{i,t}$, and $\delta_{i,t}$, respectively. Constraints 24 and 25 are nonnegativity constraints.

Budget Approximation Optimization Model

The proposed budget approximation optimization model is applied to estimate the required quarterly budget during the planning horizon for the study area, which consists of 109 control sections. The aspiration level of the percentage of control sections with retroreflectivity above the minimum requirement is set at 100%. Solutions may not attain to the aspiration level because some control sections are still under warranty, even though their retroreflectivity is below the minimum requirement. Therefore, these control sections cannot be repainted. The result of the application of the proposed model to the study area is shown in Figure 6.

The result shows that the aspiration level of 100% is not attainable during the planning horizon because some control sections are still under warranty even though their retroreflectivity is below the minimum requirement, which indicates an inconsistency between the DOH specification and the field practice in Thailand. The fifth quarter acquires the highest budget of 45.616 million bahts ($1.52 million; 30 bahts = $1.00, 2011 U.S. dollars) and the third quarter acquires the lowest budget of 3.488 million bahts ($0.116 million) in the planning horizon. In Figure 6, the pavement marking system gradually deteriorates from 95.41% with the approximated annual budget of 45.616 million bahts ($1.520 million) in Quarter 5 to 77.98% with the annual budget of 18.909 million bahts ($0.630 million) in Quarter 10 before it is gradually improved to reach its new high percentage in Quarter 15. This phenomenon appears to cycle over the planning period mostly because of the inconsistency between the DOH specification and the field practice.

Alternatively, Table 3 shows an annual view of the approximated budget and associated average percentages as opposed to the quarterly view in Figure 6. The annual budget for Year 2 is 3.01, 1.52, and 1.78 times higher than those for Years 1, 3, and 4, respectively, whereas the annual average percentage of good condition control sections in Year 2 is 7.11% and 10.19% better than those in Years 1 and 3, respectively, but 1.40% worse than that in Year 4. The annual

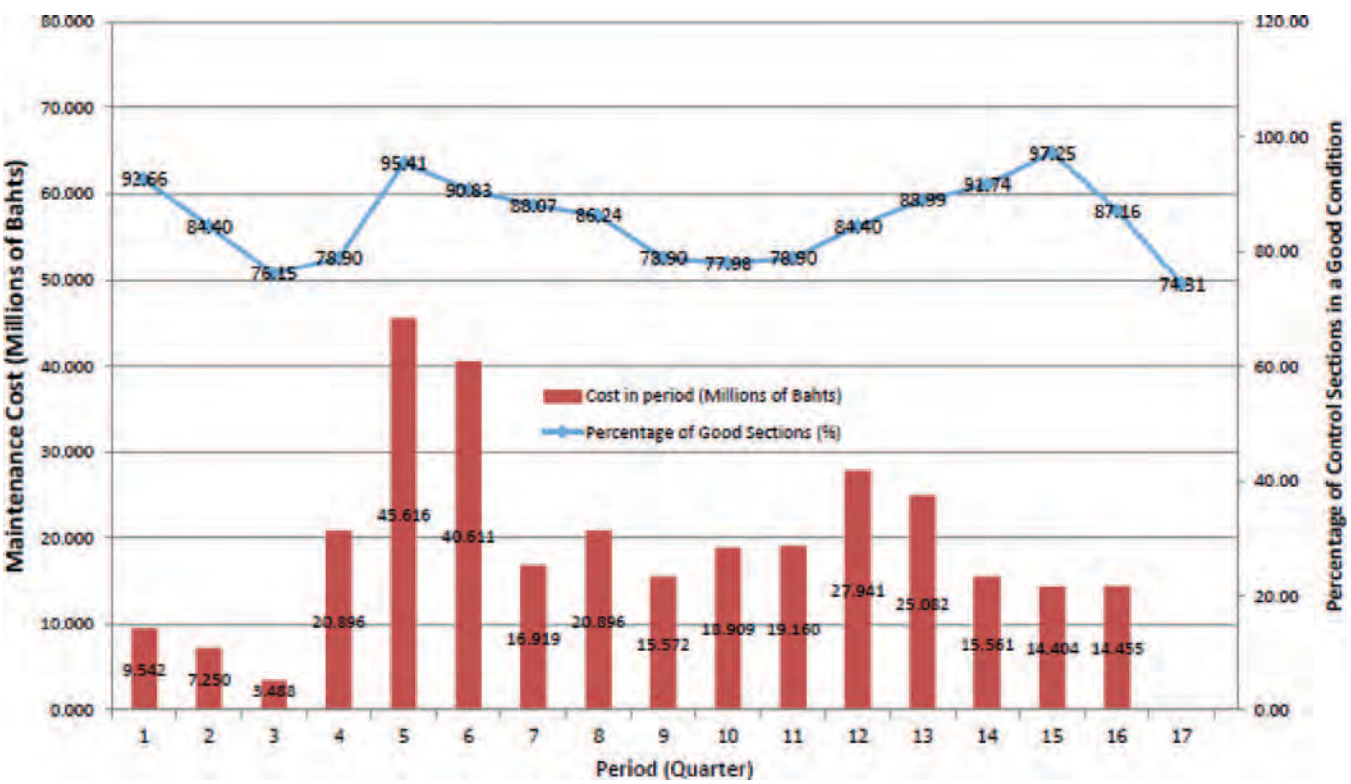


FIGURE 6 Pavement marking maintenance budget approximation result at study area.

TABLE 3 Approximated Annual Budget and Average Percentage of Control Sections in Good Condition over Planning Period

Year	Approximated Annual Budget [million bahts (US$ millions)]	Annual Average Percentage of Control Sections in Good Condition
1	41.176 (1.373)	83.03
2	124.042 (4.135)	90.14
3	81.582 (2.719)	79.95
4	69.502 (2.317)	91.54

average percentages in Table 3 are the best possible values without a budget constraint given the 2-year warranty specification of DOH. The approximated annual budgets are not uniform and greatly fluctuated over the 4 years. The proposed budget approximation optimization program can further be enhanced to smooth the annual budgets over the planning period by adding relevant constraints. For instance, the control sections belonging to certain areas and highway functional classifications should be guaranteed to receive the minimal annual budgets. This can be incorporated in the proposed model.

CONCLUSIONS AND RECOMMENDATIONS

In this study, field data collection on thermoplastic pavement marking retroreflectivity was conducted in Thailand to examine the service life distribution of thermoplastic pavement markings. Unlike most studies on pavement marking retroreflectivity, the duration model is employed instead of ordinary multiple regression models. This approach considers a probabilistic nature of the service life of the markings. The proposed model is based on the MLE technique. The data are categorized into failed markings and not failed markings, depending on their retroreflectivity, compared to the minimum requirements at the time of measurement. The likelihood function is developed from these two categories of data. On the completion of this research, it was found that AADT per lane has a significant negative effect on the service life of the thermoplastic white shoulder lines and the thermoplastic white lane-separating lines because it is an indicator of marking exposure. However, it does not have a significant effect on the deterioration of thermoplastic yellow median lines mainly because of Thailand's requirement to drive on the shoulder lane except for passing maneuvers. The other variables, including percentage of heavy vehicles and wearing surface type, were not found to have a significant effect. In terms of estimated average service life, thermoplastic yellow median lines can provide an average service life of more than 24 months, as required by the DOH specifications. The thermoplastic white lines at shoulders and lane-separating lines mostly have less than 24 months of service life. The white lane-separating lines have the lowest service life among pavement marking lines because they are subjected to lane-changing activities.

Furthermore, the budget approximation optimization model was developed to determine required annual budgets in each year over the planning horizon such that the percentages of good condition control sections over various time periods are maintained. The proposed model is a mixed-integer program based on the preemptive goal programming. The first-priority goal is to maintain percentage of control sections in a good condition and the second-priority goal is to minimize total maintenance cost. From the illustrative example, it was found that the aspiration level of 100% is not attainable during the planning horizon because certain control sections are still in the warranty period of 2 years, even though their retroreflectivity is below the minimum requirement. This indicates the impact of the inconsistency between the DOH specification and the field practice. As such, the deterioration of pavement markings in the study area and the estimated quarterly budgets, as well as annual budgets, greatly fluctuate over the planning period.

In the future, the authors plan to collect more data to validate the estimated average service life of thermoplastic pavement markings. Furthermore, an asset management program for pavement markings will be developed from the findings from this research to financially plan an optimal maintenance schedule for pavement markings. The proposed budget approximation optimization model can incorporate additional constraints, such as the smoothness of estimated annual budgets and the minimum annual budgets for pavement markings in certain areas and highway functional classifications. Also, the budget allocation optimization model can be developed to help DOH optimally allocate limited available budgets to the system of pavement markings.

ACKNOWLEDGMENTS

The authors acknowledge the Department of Highways, Thailand, for research funding support. The second author gratefully acknowledges the support from the Thailand Research Fund and King Mongkut's University of Technology Thonburi Research Fund.

REFERENCES

1. Smadi, O., N. Hawkins, I. Nlenanya, and B. Aldemir-Bektas. *Pavement Markings and Safety.* Trans Project Report 08-306. Center for Transportation Research and Education, Iowa State University, Ames, Nov. 2010.
2. Lee, J., T. L. Maleck, and W. C. Taylor. Pavement Marking Material Study in Michigan. *Institute of Transportation Engineering Journal,* Vol. 69, No. 7, 1999, pp. 44–50.
3. Abboud, N., and B. L. Bowman. Cost- and Longevity-Based Scheduling of Paint and Thermoplastic Striping. In *Transportation Research Record: Journal of the Transportation Research Board, No. 1794,* Transportation Research Board of the National Academies, Washington, D.C., 2002, pp. 55–62.
4. Sarasua, W. A., D. B. Clarke, and W. J. Davis. *Evaluation of Interstate Pavement Marking Retroreflectivity.* Report FHWA-SC-03-01. South Carolina Department of Transportation, Columbia, 2003.
5. Kopf, J. *Reflectivity of Pavement Markings: Analysis of Retroreflectivity Curves.* Report WA-RD 592.1. Washington State Transportation Center, University of Washington, Seattle, 2004.
6. Sitzabee, W. E. *A Spatial Asset Management Study Through an Analysis of Pavement Marking Performance.* Ph.D. dissertation. North Carolina State University, Raleigh, 2008.
7. Sitzabee, W. E., J. E. Hummer, and W. Rasdorf. Pavement Marking Degradation Modeling and Analysis. *ASCE Journal of Infrastructure Systems,* Vol. 15, No. 3, 2009, pp. 190–199.
8. Mishalani, R. G., and S. M. Madanat. Computation of Infrastructure Transition Probabilities Using Stochastic Duration Models. *ASCE Journal of Infrastructure Systems,* Vol. 8, No. 4, 2002, pp. 139–148.
9. Hillier, F. S., and G. J. Lieberman. *Introduction to Operations Research,* 8th ed. McGraw-Hill, New York, 2005.
10. Sherali, H. D., and A. L. Soyster. Preemptive and Nonpreemptive Multi-Objective Programming: Relationship and Counterexamples. *Journal of Optimization Theory and Applications,* Vol. 39, No. 2, 1983, pp. 173–186.

The Signing and Marking Materials Committee peer-reviewed this paper.

ภาคผนวก
การเสนอผลงานในที่ประชุมวิชาการ

AN ALGORITHM FOR THE COMBINED TRIP DISTRIBUTION AND PAIRED COMBINATORIAL LOGIT STOCHASTIC USER EQUILIBRIUM ASSIGNMENT PROBLEM

by

Ampol Karoonsoontawong*, Ph.D.
Assistant Professor
Department of Civil Engineering
King Mongkut's University of Technology Thonburi
126 Pracha-Utit Road, Bangmod, Thung-Khru, Bangkok, Thailand
E-mail: ampol.kar@kmutt.ac.th
* Corresponding Author

and

Dung-Ying Lin, Ph.D.
Assistant Professor
Department of Transportation and Communication Management Science
National Cheng Kung University
1 University Road, Tainan City, 70101, Taiwan
E-mail: dylin@mail.ncku.edu.tw

Submitted for Presentation at the 91st Annual Meeting of the Transportation Research Board

15 November 2011

ABSTRACT

The equivalent mathematical formulation of the combined doubly-constrained gravity-based trip distribution and paired-combinatorial-logit stochastic user equilibrium assignment problem (CDA-PCL-SUE) is proposed. Its first order conditions are shown to be equal to the gravity equations and PCL formula. The proposed solution method is a disaggregate simplicial decomposition algorithm that iterates two phases. Phase I employs the partial linearization descent algorithm to approximately solve the restricted CDA-PCL-SUE, and Phase II is the column generation phase. In Phase I, the partially linearized problem is decomposed into an entropy maximization problem on O-D flow space that can be solved by Bregman's balancing algorithm and a PCL SUE problem that can be solved by PCL formula. CDA-PCL-SUE is compared with its multinomial-logit counterpart (CDA-MNL-SUE) on two test networks at two congestion levels. We found that the relationship of O-D flow difference and dispersion factor is not necessarily clear, whereas the link flow patterns from the two models are more identical at higher dispersion factors on the two networks at two congestion levels. At a fixed dispersion factor, CDA-PCL-SUE assigns less flow to a path with a higher average similarity index and higher path cost than CDA-MNL-SUE does. CDA-PCL-SUE generally assigns less flow to links with more paths passing through than CDA-MNL-SUE does. The relationship between O-D flow allocation and the average similarity indices for O-D pairs is not obvious, since the gravity-based trip distribution equation is based on additional variables and the origin and destination demand constraints must be satisfied.

1. INTRODUCTION

The well-known four-step transportation planning process consists of trip generation, trip distribution, modal split and traffic assignment. These are typically solved in a fixed sequence. The output from one model is the input for the next. This sequential approach has its inherent drawback such as the lack of a unifying rationale that would explain all aspects of demand jointly *(1)*. Furthermore, to achieve a consistent output from the process, the four steps have to be repeated with a feedback mechanism although the convergence cannot be guaranteed. An alternative approach is to simultaneously consider certain steps in combined travel demand models. A comprehensive review of combined travel demand models can be found in *(1)-(2)*.

The combined distribution and assignment (CDA) problem is an instance of such combined travel demand models. CDA simultaneously determines the distribution of trips between origins and destinations in a transportation network and the assignment of trips to routes in each origin-destination pair. The trip distribution is mostly assumed to be a gravity model with a negative exponential deterrence function. The static trip assignment is either user equilibrium model (UE) or stochastic user equilibrium model (SUE). UE assumes that drivers have complete and accurate information on the state of the network when they make their route choices, and drivers select optimal routes to benefit themselves the most. SUE assumes that trip assignment follows a probabilistic route choice model. The multinomial logit-based SUE model (MNL-SUE) is widely adopted in the literature. Evans *(3)* formulated the CDA problem that integrates the gravity-model trip distribution and user-equilibrium trip assignment (CDA-UE). Erlander *(4)* formulated the CDA that integrates the gravity-model trip distribution and multinomial-logit stochastic-user-equilibrium assignment (CDA-MNL-SUE). Lundgren and Patriksson *(5)* outlined the solution algorithms for CDA-UE and CDA-MNL-SUE.

With the property of independence of irrelevant alternatives (IIA) in the MNL model, the MNL-SUE has an infamous deficiency in the incapability to account for similarities between different routes. That is, in the MNL-SUE, overlapping routes are treated as uncorrelated, and this may cause counterintuitive assignment results. Although the multinomial probit-based SUE model by *(6)* can account for similarity between different routes, it is not attractive due to the lack of closed form of probability function. Over the past years, researchers adopted other discrete choice model structures (a.k.a. extended logit models) to SUE in order to capture the similarity between routes on the perceptions and decisions of drivers while keeping the analytical tractability of the logit choice probability function. The SUE models based on the modifications of MNL are C-logit model *(7)* and path-size logit model *(8)*. The SUE models based on the generalized extreme value theory are paired combinatorial logit model *(9,10,23)*, cross-nested logit model *(11)*, logit kernel model *(12)*, link-nested logit model *(13)*, and generalized nested logit model *(14)*.

Chen et al. *(10)* pointed out that among these extended logit models, the paired combinatorial logit model (PCL) is considered the most suitable for adaptation to the route choice problem due to two features that can be employed to address the IIA property (i.e. the overlapping-route problem) in the MNL model. The first feature is that each pair of routes can have a similarity relationship that is completely independent of the similarity relationship of other route pairs. Second, the PCL model can be scaled to account for perception variance with respect to different trip lengths. In this paper, we proposed a combined gravity-model distribution and paired-combinatorial-logit stochastic-user-equilibrium assignment formulation (CDA-PCL-SUE) and developed a disaggregate simplicial decomposition algorithm. The trip distribution model is doubly constrained such that both the total flow generated at each origin node and the total flow attracted to each destination node are fixed and known.

The remainder of the paper is organized as follows. Section 2 proposes the equivalent mathematical formulation of the CDA-PCL-SUE problem and derives the first-order conditions that can be used in algorithmic design. Section 3 describes the solution algorithm and highlights the details of each algorithmic step. Section 4 presents empirical analysis undertaken with two sample networks. Section 5 concludes the paper.

2. EQUIVALENT MATHEMATICAL FORMULATION

We extend the paired combinatorial logit-based stochastic user equilibrium (PCL-SUE) equivalent mathematical program by *(9,10)* to account for doubly-constrained trip distribution. Denote by CDA-PCL-SUE the proposed combined distribution-assignment (CDA) equivalent mathematical model. The underlying route choice in CDA-PCL-SUE is a hierarchical route choice model that decomposes the choice probability into two levels. The upper level computes the marginal probabilities $P(kj)$ of choosing an unordered route pair k and j, based on the similarity index and the systematic utility. The lower level is a binary logit model that computes the conditional probabilities of choosing a route given the chosen route pair: $P(k|kj)$ and $P(j|kj)$. The underlying trip distribution in CDA-PCL-SUE is a doubly constrained model that requires the O-D flows out of an origin node and into a destination node to be equal to the known origin demands and destination demands, respectively.

The definitions of sets, parameters, decision variables and mathematical formulation are given below, followed by the first-order conditions that are shown to be identical to the PCL-SUE equations and gravity-model based trip distribution equations.

2.1. Descriptions of Equivalent Mathematical Formulation

Set

K_{rs} = set of routes between origin r and destination s

L_{rs} = set of unordered route pairs between origin r and destination s

R = set of origins

S = set of destinations

RS = set of origin-destination (O-D) pairs

A = set of arcs

Parameters

O_r = total trips originated from origin r

D_s = total trips destined to destination s

θ =dispersion coefficient

β_{kj}^{rs} =measure of dissimilarity index between routes k and j connecting O-D r-s ($\beta_{kj}^{rs} = 1 - \sigma_{kj}^{rs}$)

σ_{kj}^{rs} =measure of similarity index between routes k and j connecting O-D r-s

δ_{ka}^{rs} =1 if arc a is on route k connecting origin r to destination s; 0 otherwise

Decision Variables

x_a = flow on link a

t_a = travel time on link a

q_{rs} = demand between origin r and destination s

$f^{rs}_{k(kj)}$ = flow on route k of route pair kj between origin r and destination s

c^{rs}_k = travel time on route k connecting O-D pair r,s.

Note that $f^{rs}_{k(kj)} = f^{rs}_{k(jk)}$. For instance, given $K_{rs} = \{1,2,3\}$, the expression $f^{rs}_{k(kj)}\ \forall k \in K_{rs}, j \neq k$ includes six variables: $f^{rs}_{1(12)}$, $f^{rs}_{1(13)}$, $f^{rs}_{2(21)}$, $f^{rs}_{2(23)}$, $f^{rs}_{3(31)}$, and $f^{rs}_{3(32)}$. The expression $f^{rs}_{k(kj)}$ $\forall k = 1,...,|K_{rs}|-1; \forall j = k+1,...,|K_{rs}|$ includes three variables: $f^{rs}_{1(12)}$, $f^{rs}_{1(13)}$, and $f^{rs}_{2(23)}$.

Mathematical Formulation

$$\min z = z_1 + z_2 + z_3 \tag{1.1}$$

$$z_1 = \sum_{a \in A} \int_0^{x_a} t_a(w)dw \tag{1.1a}$$

$$z_2 = \frac{1}{\theta} \sum_{r \in R} \sum_{s \in S} \sum_{k \in K_{rs}} \sum_{\substack{j \in K_{rs} \\ j \neq k}} \beta^{rs}_{kj} f^{rs}_{k(kj)} \ln \frac{f^{rs}_{k(kj)}}{\beta^{rs}_{kj}} \tag{1.1b}$$

$$z_3 = \frac{1}{\theta} \sum_{r \in R} \sum_{s \in S} \sum_{k=1}^{|K_{rs}|-1} \sum_{j=k+1}^{|K_{rs}|} (1-\beta^{rs}_{kj})(f^{rs}_{k(kj)} + f^{rs}_{j(kj)}) \ln \frac{f^{rs}_{k(kj)} + f^{rs}_{j(kj)}}{\beta^{rs}_{kj}} \tag{1.1c}$$

Subject to

$$\sum_{k \in K_{rs}} \sum_{\substack{j \in K_{rs} \\ j \neq k}} f^{rs}_{k(kj)} = q_{rs} \quad \forall r \in R, s \in S \tag{1.2}$$

$$\sum_{s \in S} q_{rs} = O_r \quad \forall r \in R \tag{1.3}$$

$$\sum_{r \in R} q_{rs} = D_s \quad \forall s \in S \tag{1.4}$$

$$f^{rs}_{k(kj)} \geq 0 \quad \forall k \in K_{rs}, kj \in L_{rs}, r \in R, s \in S \tag{1.5}$$

$$x_a = \sum_{r \in R} \sum_{s \in S} \sum_{k \in K_{rs}} \sum_{\substack{j \in K_{rs} \\ j \neq k}} \delta^{rs}_{ak} \cdot f^{rs}_{k(kj)} \quad \forall a \in A \tag{1.6}$$

The objective function (1.1) is composed of three components, similar to the objective of the PCL-SUE model. Eq.(1.1a) accounts for the congestion effects. Eq.(1.1b) and (1.1c) are two entropy terms that represent the marginal and conditional probabilities in a hierarchical route choice model. Dissimilarity indices are incorporated into the objective function (Eq.1.1b and Eq. 1.1c), allowing the model to capture the similarity effect and stochastic perception error effect in addition to the congestion effect (Eq.1.1a). Eq.1.2 enforces the summation of all path flows connecting an O-D pair to be equal to the O-D flows (q_{rs}) of this O-D pair. Eq.1.3 and Eq.1.4 are the O-D flow balance constraints for the origin nodes and destination nodes, respectively. Eq.1.5 are the non-negativity constraints for all path flow variables. Eq.1.6 determines the link flow variable from the summation of all path flows passing through this link.

2.1.2. First-Order Conditions

We use the projection of CDA-PCL-SUE onto the set of route flows. That is, Eq.(1.3) and Eq.(1.4) are substituted by Eq. (1.2), yielding Eq.(1.7) and Eq.(1.8):

$$\sum_{s\in S}\sum_{k\in K_{rs}}\sum_{\substack{j\in K_{rs}\\ j\neq k}} f^{rs}_{k(kj)} = O_r \quad \forall r\in R \qquad : u_r \tag{1.7}$$

$$\sum_{r\in R}\sum_{k\in K_{rs}}\sum_{\substack{j\in K_{rs}\\ j\neq k}} f^{rs}_{k(kj)} = D_s \quad \forall s\in S \qquad : v_s \tag{1.8}$$

where u_r and v_s are the dual variables corresponding to Eq.(1.7) and Eq.(1.8), respectively. The projected program consists of the objective (1.1), the nonnegativity constraints (1.5) and Eq.(1.7) and Eq.(1.8). This is an equivalent CDA-PCL-SUE program, and its first-order conditions must be identical to the equilibrium equations and doubly-constrained gravity equations. These conditions can be derived by forming and analyzing the Lagrangian, which, for this program, is given by

$$L(f,q,u,v) = z_1 + z_2 + z_3 + z_4 + z_5$$

where

$$z_4 = \sum_{r\in R} u_r \Big(O_r - \sum_{s\in S}\sum_{k\in K_{rs}}\sum_{\substack{j\in K_{rs}\\ j\neq k}} f^{rs}_{k(kj)}\Big)$$

$$z_5 = \sum_{s\in S} v_s \Big(D_s - \sum_{r\in R}\sum_{k\in K_{rs}}\sum_{\substack{j\in K_{rs}\\ j\neq k}} f^{rs}_{k(kj)}\Big)$$

The minimum of this Lagrangian with respect to the path flow variables has to be subject to the nonnegativity constraints Eq.(1.5). The maximum of the Lagrangian with respect to u and v, is unconstrained. The first-order conditions for a saddle point of this Lagrangian program are given by

$$\frac{\partial L(\cdot)}{\partial f^{rs}_{k(kj)}} \geq 0 \quad \forall k\in K_{rs}, kj\in L_{rs}, r\in R, s\in S$$

$$f^{rs}_{k(kj)}\frac{\partial L(\cdot)}{\partial f^{rs}_{k(kj)}} = 0 \quad \forall k\in K_{rs}, kj\in L_{rs}, r\in R, s\in S$$

$$f^{rs}_{k(kj)} \geq 0 \quad \forall k\in K_{rs}, kj\in L_{rs}, r\in R, s\in S$$

$$\frac{\partial L(\cdot)}{\partial u_r} = 0 \quad \forall r\in R$$

$$\frac{\partial L(\cdot)}{\partial v_s} = 0 \quad \forall s\in S$$

$$\frac{\partial L}{\partial f^{rs}_{k(kj)}} = \frac{\partial z_1}{\partial f^{rs}_{k(kj)}} + \frac{\partial z_2}{\partial f^{rs}_{k(kj)}} + \frac{\partial z_3}{\partial f^{rs}_{k(kj)}} + \frac{\partial z_4}{\partial f^{rs}_{k(kj)}} + \frac{\partial z_5}{\partial f^{rs}_{k(kj)}}$$

$$= c^{rs}_k - u_r - v_s + \frac{1}{\theta} + \frac{\beta^{rs}_{kj}}{\theta}\ln\left(\frac{f^{rs}_{k(kj)}}{\beta^{rs}_{kj}}\right) + \left(\frac{1-\beta^{rs}_{kj}}{\theta}\right)\ln\left(\frac{f^{rs}_{k(kj)} + f^{rs}_{j(kj)}}{\beta^{rs}_{kj}}\right)$$

If $f^{rs}_{k(kj)}$>0, then $\dfrac{\partial L}{\partial f^{rs}_{k(kj)}} = 0$, yielding

$$\ln\left(\frac{f^{rs}_{k(kj)}}{\beta^{rs}_{kj}}\right)^{\frac{\beta^{rs}_{kj}}{\theta}} + \ln\left(\frac{f^{rs}_{k(kj)} + f^{rs}_{j(kj)}}{\beta^{rs}_{kj}}\right)^{\frac{1-\beta^{rs}_{kj}}{\theta}} = u_r + v_s - c^{rs}_k - \frac{1}{\theta}$$

$$\left(\frac{f^{rs}_{k(kj)}}{\beta^{rs}_{kj}}\right)^{\frac{\beta^{rs}_{kj}}{\theta}} \left(\frac{f^{rs}_{k(kj)} + f^{rs}_{j(kj)}}{\beta^{rs}_{kj}}\right)^{\frac{1-\beta^{rs}_{kj}}{\theta}} = \exp\left(u_r + v_s - c^{rs}_k - \frac{1}{\theta}\right)$$

Both sides are raised to the power of $\dfrac{\theta}{\beta^{rs}_{kj}}$:

$$f^{rs}_{k(kj)}\left(f^{rs}_{k(kj)} + f^{rs}_{j(kj)}\right)^{\frac{1-\beta^{rs}_{kj}}{\beta^{rs}_{kj}}} = {\beta^{rs}_{kj}}^{\frac{1}{\beta^{rs}_{kj}}} \cdot \exp\left(\frac{\theta u_r}{\beta^{rs}_{kj}} + \frac{\theta v_s}{\beta^{rs}_{kj}} - \frac{1}{\beta^{rs}_{kj}}\right) \cdot \exp\left(-\frac{\theta c^{rs}_k}{\beta^{rs}_{kj}}\right) \quad (2)$$

If $f^{rs}_{j(kj)}$>0, then $\dfrac{\partial L}{\partial f^{rs}_{j(kj)}} = 0$, yielding Eq.(3)

$$f^{rs}_{j(kj)}\left(f^{rs}_{k(kj)} + f^{rs}_{j(kj)}\right)^{\frac{1-\beta^{rs}_{kj}}{\beta^{rs}_{kj}}} = {\beta^{rs}_{kj}}^{\frac{1}{\beta^{rs}_{kj}}} \cdot \exp\left(\frac{\theta u_r}{\beta^{rs}_{kj}} + \frac{\theta v_s}{\beta^{rs}_{kj}} - \frac{1}{\beta^{rs}_{kj}}\right) \cdot \exp\left(-\frac{\theta c^{rs}_j}{\beta^{rs}_{kj}}\right) \quad (3)$$

The summation of Equations (2) and (3) yields Eq.(4) and Eq.(5).

$$\left(f^{rs}_{k(kj)} + f^{rs}_{j(kj)}\right)^{\frac{1}{\beta^{rs}_{kj}}} = {\beta^{rs}_{kj}}^{\frac{1}{\beta^{rs}_{kj}}} \cdot \exp\left(\frac{\theta u_r}{\beta^{rs}_{kj}} + \frac{\theta v_s}{\beta^{rs}_{kj}} - \frac{1}{\beta^{rs}_{kj}}\right) \cdot \left(\exp\left(-\frac{\theta c^{rs}_k}{\beta^{rs}_{kj}}\right) + \exp\left(-\frac{\theta c^{rs}_j}{\beta^{rs}_{kj}}\right)\right) \quad (4)$$

$$\left(f^{rs}_{k(kj)} + f^{rs}_{j(kj)}\right) = \beta^{rs}_{kj} \cdot \exp\left(\theta u_r + \theta v_s - 1\right) \cdot \left(\exp\left(-\frac{\theta c^{rs}_k}{\beta^{rs}_{kj}}\right) + \exp\left(-\frac{\theta c^{rs}_j}{\beta^{rs}_{kj}}\right)\right)^{\beta^{rs}_{kj}} \quad (5)$$

Substitute *m* and *l* for *k* and *j* in Equation (5) and add the summation operators:

$$\sum_{m=1}^{|K_{rs}|-1} \sum_{l=m+1}^{|K_{rs}|} \left(f^{rs}_{m(ml)} + f^{rs}_{l(ml)}\right) = \exp\left(\theta u_r + \theta v_s - 1\right) \sum_{m=1}^{|K_{rs}|-1} \sum_{l=m+1}^{|K_{rs}|} \beta^{rs}_{ml} \cdot \left(\exp\left(-\frac{\theta c^{rs}_m}{\beta^{rs}_{ml}}\right) + \exp\left(-\frac{\theta c^{rs}_l}{\beta^{rs}_{ml}}\right)\right)^{\beta^{rs}_{ml}} \quad (6)$$

Eq.(5) is divided by Eq.(6), yielding the marginal probability of choosing route pair *k,j* among all possible pairs *m,l*:

$$\frac{f^{rs}_{k(kj)} + f^{rs}_{j(kj)}}{\sum_{m=1}^{|K_{rs}|-1} \sum_{l=m+1}^{|K_{rs}|} \left(f^{rs}_{m(ml)} + f^{rs}_{l(ml)}\right)} = P(kj) = \frac{\beta^{rs}_{kj} \cdot \left(\exp\left(-\frac{\theta c^{rs}_k}{\beta^{rs}_{kj}}\right) + \exp\left(-\frac{\theta c^{rs}_j}{\beta^{rs}_{kj}}\right)\right)^{\beta^{rs}_{kj}}}{\sum_{m=1}^{|K_{rs}|-1} \sum_{l=m+1}^{|K_{rs}|} \beta^{rs}_{ml} \cdot \left(\exp\left(-\frac{\theta c^{rs}_m}{\beta^{rs}_{ml}}\right) + \exp\left(-\frac{\theta c^{rs}_l}{\beta^{rs}_{ml}}\right)\right)^{\beta^{rs}_{ml}}} \quad (7)$$

Substituting Eq.(4) in Eq.(2) yields:

$$f_{k(kj)}^{rs} = \frac{\beta_{kj}^{rs} \cdot \exp\left(\theta u_r + \theta v_s - 1\right) \cdot \exp\left(-\dfrac{\theta c_k^{rs}}{\beta_{kj}^{rs}}\right)}{\left(\exp\left(-\dfrac{\theta c_k^{rs}}{\beta_{kj}^{rs}}\right) + \exp\left(-\dfrac{\theta c_j^{rs}}{\beta_{kj}^{rs}}\right)\right)^{1-\beta_{kj}^{rs}}} \tag{8}$$

Equation (8) is divided by Equation (5), yielding Eq.(9) the conditional probability of choosing route *k* from route pair *k,j*:

$$\frac{f_{k(kj)}^{rs}}{f_{k(kj)}^{rs} + f_{j(kj)}^{rs}} = P(k \mid kj) = \frac{\exp\left(-\dfrac{\theta c_k^{rs}}{\beta_{kj}^{rs}}\right)}{\left(\exp\left(-\dfrac{\theta c_k^{rs}}{\beta_{kj}^{rs}}\right) + \exp\left(-\dfrac{\theta c_j^{rs}}{\beta_{kj}^{rs}}\right)\right)} \tag{9}$$

Substituting (8) in (1.2) yields Eq.(10):

$$q_{rs} = \exp\left(\theta u_r - 1\right) \cdot \exp(\theta v_s) \sum_{k \in K_{rs}} \sum_{\substack{j \in K_{rs} \\ j \neq k}} \frac{\beta_{kj}^{rs} \cdot \exp\left(-\dfrac{\theta c_k^{rs}}{\beta_{kj}^{rs}}\right)}{\left(\exp\left(-\dfrac{\theta c_k^{rs}}{\beta_{kj}^{rs}}\right) + \exp\left(-\dfrac{\theta c_j^{rs}}{\beta_{kj}^{rs}}\right)\right)^{1-\beta_{kj}^{rs}}} \quad \forall r \in R, s \in S \tag{10}$$

The O-D flow conservation constraints (Eq.1.3 and Eq.1.4) are substituted by Eq.(10), yielding Eq.11 and Eq.12:

$$\exp\left(\theta u_r - 1\right) = \frac{O_r}{\displaystyle\sum_{s \in S} \exp(\theta v_s) \sum_{k \in K_{rs}} \sum_{\substack{j \in K_{rs} \\ j \neq k}} \frac{\beta_{kj}^{rs} \cdot \exp\left(-\dfrac{\theta c_k^{rs}}{\beta_{kj}^{rs}}\right)}{\left(\exp(-\dfrac{\theta c_k^{rs}}{\beta_{kj}^{rs}}) + \exp(-\dfrac{\theta c_j^{rs}}{\beta_{kj}^{rs}})\right)^{1-\beta_{kj}^{rs}}}}$$

$$= \frac{O_r}{\displaystyle\sum_{s \in S} \exp(\theta v_s) g(c^{rs})} \qquad \forall r \in R \tag{11}$$

$$\exp(\theta v_s) = \frac{D_s}{\displaystyle\sum_{r \in R} \exp\left(\theta u_r - 1\right) \sum_{k \in K_{rs}} \sum_{\substack{j \in K_{rs} \\ j \neq k}} \frac{\beta_{kj}^{rs} \cdot \exp\left(-\dfrac{\theta c_k^{rs}}{\beta_{kj}^{rs}}\right)}{\left(\exp\left(-\dfrac{\theta c_k^{rs}}{\beta_{kj}^{rs}}\right) + \exp\left(-\dfrac{\theta c_j^{rs}}{\beta_{kj}^{rs}}\right)\right)^{1-\beta_{kj}^{rs}}}}$$

$$= \frac{D_s}{\displaystyle\sum_{r \in R} \exp\left(\theta u_r - 1\right) g(c^{rs})} \qquad \forall s \in S \tag{12}$$

where $g(c^{rs}) = \sum_{k \in K_{rs}} \sum_{\substack{j \in K_{rs} \\ j \neq k}} \frac{\beta_{kj}^{rs} \cdot \exp\left(-\frac{\theta c_k^{rs}}{\beta_{kj}^{rs}}\right)}{\left(\exp\left(-\frac{\theta c_k^{rs}}{\beta_{kj}^{rs}}\right) + \exp\left(-\frac{\theta c_j^{rs}}{\beta_{kj}^{rs}}\right)\right)^{1-\beta_{kj}^{rs}}}$ (13)

To obtain the result in a more familiar form, we denote

$$A_r = \frac{\exp(\theta u_r - 1)}{O_r}$$

$$B_s = \frac{\exp(\theta v_s)}{D_s}$$

Eq.(10) can now be written as a gravity-model based trip distribution model:

$$q_{rs} = A_r B_s O_r D_s g(c^{rs}) \qquad \forall r \in R,\ \forall s \in S$$

where $g(c^{rs})$ is the function defined in Eq.(13) and c^{rs} is the vector of route travel times of O-D pair *r,s*.

3. DISAGGREGATE SIMPLICIAL DECOMPOSITION ALGORITHM

The proposed algorithm for CDA-PCL-SUE is based on the disaggregate simplicial decomposition algorithm by *(5,22)*. The proposed algorithm alternates between two phases. In phase I (the restricted master problem), given known subsets of routes between O-D pairs $\hat{K}_{rs} \subseteq K_{rs}\ \forall r \in R, s \in S$, of the total sets of routes in the network, the corresponding restriction of CDA-PCL-SUE (denoted by CDA-PCL-SUE-R) is solved approximately using a partial linearization descent algorithm, which is a descent algorithm for continuous optimization problems *(15)*. Phase I is composed of three sub-phases: Phase I.1, Phase I.2 and Phase I.3. Phase I.1 is an entropy maximization problem that can be solved by Bregman's balancing algorithm to determine the auxiliary O-D flows. Phase I.2 applies the PCL formula to determine the auxiliary route flows. Phase I.3 is the line search for the next solutions (route flows and O-D flows) to the CDA-PCL-SUE-R. In phase 2 (the column generation problem), at the approximate solution to CDA-PCL-SUE-R, the subsets $\hat{K}_{rs}$ are augmented by the generation of new routes, through the solution of a set of shortest path problems, given appropriately chosen link costs.

3.1. Phase I: Restricted Master Problem

The problem CDA-PCL-SUE-R is solved by a partial linearization descent algorithm. The projection of CDA-PCL-SUE-R onto the set of feasible route flows is employed. Given a feasible route flow vector $f^n = \{f_{k(kj)}^{rs^n}\}$ at some iteration *n*, an approximation of CDA-PCL-SUE-R is roughly solved in order to define an auxiliary feasible solution and a search direction. The approximate problem is constructed by linearizing the first term (z_1) of the objective function of CDA-PCL-SUE-R. The effect of this linearization is that the link costs are fixed at their levels given the current flow f^n; i.e. $\frac{\partial z_1(f^n)}{\partial f_{k(kj)}^{rs}} = c_k^{rs^n}$. The corresponding route costs are calculated as: $c_k^{rs^n} = \sum_{a \in A} \delta_{ka}^{rs} \cdot t_a(x_a^n) \quad \forall k \in \hat{K}_{rs}, r \in R, s \in S$, where x_a^n is the flow on arc a corresponding to the route flow f^n. The partially linearized problem (denoted by CDA-PCL-SUE-R-PL) becomes:

Formulation of CDA-PCL-SUE-R-PL (the solution is denoted by $\underline{f}_{k(kj)}^{rs}$)

$$\min \tilde{z} = \tilde{z}_1 + \tilde{z}_2 + \tilde{z}_3 \tag{14.1}$$

$$\tilde{z}_1 = \sum_{r \in R} \sum_{s \in S} \sum_{k \in \hat{K}_{rs}} \sum_{\substack{j \in \hat{K}_{rs} \\ j \neq k}} c_k^{rs^n} \cdot f_{k(kj)}^{rs} \tag{14.1a}$$

$$\tilde{z}_2 = \frac{1}{\theta} \sum_{r \in R} \sum_{s \in S} \sum_{k \in \hat{K}_{rs}} \sum_{\substack{j \in \hat{K}_{rs} \\ j \neq k}} \beta_{kj}^{rs} f_{k(kj)}^{rs} \ln \frac{f_{k(kj)}^{rs}}{\beta_{kj}^{rs}} \tag{14.1b}$$

$$\tilde{z}_3 = \frac{1}{\theta} \sum_{r \in R} \sum_{s \in S} \sum_{k=1}^{|\hat{K}_{rs}|-1} \sum_{j=k+1}^{|\hat{K}_{rs}|} (1-\beta_{kj}^{rs})(f_{k(kj)}^{rs} + f_{j(kj)}^{rs}) \ln \frac{f_{k(kj)}^{rs} + f_{j(kj)}^{rs}}{\beta_{kj}^{rs}} \tag{14.1c}$$

Subject to

$$\sum_{s \in S} \sum_{k \in \hat{K}_{rs}} \sum_{\substack{j \in \hat{K}_{rs} \\ j \neq k}} f_{k(kj)}^{rs} = O_r \quad \forall r \in R \tag{14.2}$$

$$\sum_{r \in R} \sum_{k \in \hat{K}_{rs}} \sum_{\substack{j \in \hat{K}_{rs} \\ j \neq k}} f_{k(kj)}^{rs} = D_s \quad \forall s \in S \tag{14.3}$$

$$f_{k(kj)}^{rs} \geq 0 \quad \forall k \in \hat{K}_{rs}, kj \in \hat{L}_{rs}, r \in R, s \in S \tag{14.4}$$

It is noted that in CDA-PCL-SUE-R-PL only $f_{k(kj)}^{rs}$ are decision variables, since q_{rs} are substituted by $q_{rs} = \sum_{k \in \hat{K}_{rs}} \sum_{\substack{j \in \hat{K}_{rs} \\ j \neq k}} f_{k(kj)}^{rs}$. We next consider the following equivalent formulation to CDA-PCL-SUE-R-PL, which is the projection of CDA-PCL-SUE-R-PL onto the demand space, in order to solve the problem CDA-PCL-SUE-R-PL.

Equivalent Formulation to CDA-PCL-SUE-R-PL

$$\min U(q) \tag{15.1}$$

Subject to

$$\sum_{s \in S} q_{rs} = O_r \quad \forall r \in R \tag{15.2}$$

$$\sum_{r \in R} q_{rs} = D_s \quad \forall s \in S \tag{15.3}$$

$$q_{rs} \geq 0 \quad \forall r \in R, s \in S \tag{15.4}$$

where $U(q) = \tilde{z}_1 + \tilde{z}_2 + \tilde{z}_3$ (15.5)

Subject to

$$\sum_{k \in \hat{K}_{rs}} \sum_{\substack{j \in \hat{K}_{rs} \\ j \neq k}} f^{rs}_{k(kj)} = q_{rs} \quad \forall r \in R, s \in S \tag{15.6}$$

$$f^{rs}_{k(kj)} \geq 0 \quad \forall k \in \hat{K}_{rs}, kj \in \hat{L}_{rs}, r \in R, s \in S \tag{15.7}$$

This equivalent formulation utilizes the fact that the solution to (15.5)-(15.7) (i.e. the restricted PCL-SUE) is easily obtained by the use of the PCL formula: $f^{rs}_{k(kj)} = P(kj) \cdot P(k \mid kj) \cdot q_{rs}$. By performing the substitution of the PCL formula in (15.5) (i.e. $f^{rs}_{k(kj)} = P(kj)_n \cdot P(k \mid kj)_n \cdot q_{rs}$ and $f^{rs}_{j(kj)} = P(kj)_n \cdot P(j \mid kj)_n \cdot q_{rs}$ are substituted in $U(q) = \tilde{z}_1 + \tilde{z}_2 + \tilde{z}_3$), it can be proved that the implicit function $U(q)$ actually has the explicit form of the entropy maximization problem (problem (16.1)-(16.8) in Phase I.1). Hence, it is clear that CDA-PCL-SUE-R-PL is solved through the solution of the entropy maximization problem followed by the application of the PCL formula. Specifically, an optimal solution to the equivalent formulation of CDA-PCL-SUE-R-PL is obtained by Phases I.1-I.3.

<u>Phase I.1 Entropy Maximization Problem</u> (the solution is denoted by $\underline{q}^n_{rs}$ $\forall r \in R, s \in S$)

$$\min \sum_{r \in R} \sum_{s \in S} \gamma^n_{rs} \cdot \underline{q}_{rs} + \nu^n_{rs} \cdot \underline{q}_{rs} \cdot \ln \underline{q}_{rs} \tag{16.1}$$

Subject to

$$\sum_{s \in S} \underline{q}_{rs} = O_r \quad \forall r \in R \quad : \alpha_r \tag{16.2}$$

$$\sum_{r \in R} \underline{q}_{rs} = D_s \quad \forall s \in S \quad : \lambda_s \tag{16.3}$$

$$\underline{q}_{rs} \geq 0 \quad \forall r \in R, s \in S \tag{16.4}$$

where

$$\gamma^n_{rs} = \left\{ \begin{array}{l} \displaystyle\sum_{k \in \hat{K}_{rs}} \sum_{\substack{j \in \hat{K}_{rs} \\ j \neq k}} \left[P(kj)_n \cdot P(k \mid kj)_n \cdot \left(c^{rs^n}_k + \frac{1}{\theta} \beta^{rs}_{kj} \cdot \ln \left(\frac{P(kj)_n \cdot P(k \mid kj)_n}{\beta^{rs}_{kj}} \right) \right) \right] \\ \displaystyle + \sum_{k=1}^{|\hat{K}_{rs}|-1} \sum_{j=k+1}^{|\hat{K}_{rs}|} \frac{1}{\theta} (1 - \beta^{rs}_{kj}) \cdot P(kj)_n \cdot \ln \frac{P(kj)_n}{\beta^{rs}_{kj}} \end{array} \right\} \tag{16.5}$$

$$\nu^n_{rs} = \left\{ \begin{array}{l} \displaystyle\sum_{k \in \hat{K}_{rs}} \sum_{\substack{j \in \hat{K}_{rs} \\ j \neq k}} \frac{1}{\theta} \beta^{rs}_{kj} \cdot P(kj)_n \cdot P(k \mid kj)_n \\ \displaystyle + \sum_{k=1}^{|\hat{K}_{rs}|-1} \sum_{j=k+1}^{|\hat{K}_{rs}|} \frac{1}{\theta} (1 - \beta^{rs}_{kj}) \cdot P(kj)_n \end{array} \right\} \tag{16.6}$$

$$P(kj)_n = \frac{\beta_{kj}^{rs}\left(\exp(-\theta c_k^{rs^n} / \beta_{kj}^{rs}) + \exp(-\theta c_j^{rs^n} / \beta_{kj}^{rs})\right)^{\beta_{kj}^{rs}}}{\sum_{m=1}^{|\hat{K}_{rs}|-1} \sum_{l=m+1}^{|\hat{K}_{rs}|} \beta_{ml}^{rs}\left(\exp(-\theta c_m^{rs^n} / \beta_{ml}^{rs}) + \exp(-\theta c_l^{rs^n} / \beta_{ml}^{rs})\right)^{\beta_{ml}^{rs}}} \qquad (16.7)$$

$$P(k \mid kj)_n = \frac{\exp(-\theta c_k^{rs^n} / \beta_{kj}^{rs})}{\exp(-\theta c_k^{rs^n} / \beta_{kj}^{rs}) + \exp(-\theta c_j^{rs^n} / \beta_{kj}^{rs})} \qquad (16.8)$$

The entropy maximization problem (16.1)-(16.8) can be solved by Bregman's balancing method *(16)*, and the result is an auxiliary demand $\underline{q}^n = \{\underline{q}_{rs}^n\}$. The detailed development of Bregman's balancing method for (16.1-16.8) is described in Appendix. The balancing method is briefly described below.

<u>*Bregman's Balancing Method*</u>

Initialization of Balancing Method:

$$\underline{q}_{rs}^0 = \exp(-1 - \gamma_{rs}^n / \nu_{rs}^n) \quad \forall r \in R, s \in S$$

(see Appendix for the derivation of initial auxiliary O-D flows)

$t \leftarrow 0$ (t is iteration counter for the balancing method)

$i \leftarrow 1$ (i is the constraint counter of the entropy maximization problem)

General Step of Balancing Method (Balancing Constraint i):

Find the unique solution $\underline{q}_{rs}^{t+1}$ and ξ of:

$$\nu_{rs}^n \ln \underline{q}_{rs}^{t+1} - \nu_{rs}^n \ln \underline{q}_{rs}^t - \xi a_{i,rs} = 0 \quad \forall r \in R, s \in S \qquad (17)$$

and $\sum_{rs \in RS} a_{i,rs}\,\underline{q}_{rs}^{t+1} = b_i$. (18)

The derivation of Eq.(17) and Eq.(18) can be found in Appendix. Then, Eq.(17) can be written as Eq.(19).

$$\underline{q}_{rs}^{t+1} = \underline{q}_{rs}^t \exp\left(\frac{\xi a_{i,rs}}{\nu_{rs}^n}\right) \qquad \forall r \in R, s \in S \qquad (19)$$

which is then substitute into Eq.(18), yielding Eq.(20):

$$\sum_{rs} a_{i,rs} \cdot \underline{q}_{rs}^t \exp\left(\frac{\xi a_{i,rs}}{\nu_{rs}^n}\right) = b_i \qquad (20)$$

where $b_i = O_i$ and $\xi = \alpha_i$ if $1 \le i \le |R|$

$b_i = D_i$ and $\xi = \lambda_i$ if $|R| + 1 \le i \le |R| + |S|$

Determine ξ by Newton's method (since it cannot be solved analytically):

Let $h(\xi_\omega) = \sum_{rs} a_{i,rs} \cdot \underline{q}_{rs}^t \exp\left(\frac{\xi_\omega a_{i,rs}}{\nu_{rs}^n}\right) - O_i = 0$ and $h'(\xi_\omega) = \sum_s \frac{1}{\nu_{is}^n} \cdot \underline{q}_{is}^t \exp\left(\frac{\xi_\omega}{\nu_{is}^n}\right)$ if $1 \le i \le |R|$.

Let $h(\xi_\omega) = \sum_{rs} a_{i',rs} \cdot \underline{q_{rs}^t} \exp\left(\frac{\xi_\omega a_{i',rs}}{\nu_{rs}^n}\right) - D_{i'} = 0$ and $h'(\xi_\omega) = \sum_r \frac{1}{\nu_{ri'}^n} \cdot \underline{q_{ri'}^t} \exp\left(\frac{\xi_\omega}{\nu_{ri'}^n}\right)$, where $i' = i - |R|$, if $|R| + 1 \le i \le |R| + |S|$.

Initialization of Newton's Method: $\xi_0 \leftarrow 0$ and $\omega \leftarrow 0$

General Step of Newton's Method: $\xi_{\omega+1} = \xi_\omega - \dfrac{h(\xi_\omega)}{h'(\xi_\omega)}$

If $|\xi_{\omega+1} - \xi_\omega| \le \varepsilon$, stop and return $\xi = \xi_{\omega+1}$.

Otherwise, $\omega \leftarrow \omega + 1$

Then, determine $\underline{q_{rs}^{t+1}}$ from Eq.(19): $q_{rs}^{t+1} = q_{rs}^t \exp\left(\frac{\xi a_{i,rs}}{\nu_{rs}^n}\right) \quad \forall r \in R, s \in S$.

$i \leftarrow (i \operatorname{modulo}(|R| + |S|)) + 1$

$t \leftarrow t + 1$

For each pass of the algorithm (when all origins and destinations are balanced once), if $\underline{q_{rs}^t}$ is converged (see the employed convergence criterion in Section 3.3), terminate the algorithm.

Phase I.2 The solution ($\underline{f}_{k(kj)}^{rs^n}$) of CDA-PCL-SUE-R-PL is obtained by applying the PCL formula:

$$\underline{f}_k^{rs^n} = P(k)_n \cdot \underline{q}_{rs}^n \quad \forall k \in K_{rs}, r \in R, s \in S$$

where

$$P(k)_n = \sum_{\substack{j \in \hat{K}_{rs} \\ j \ne k}} P(kj)_n \cdot P(k \mid kj)_n$$

Phase I.3 (Line Search)

An approximate line search is then made with respect to $z = z_1 + z_2 + z_3$ (the objective function of CDA-PCL-SUE-R) in the (feasible) direction of $\underline{f}^n - f^n$ and $\underline{q}^n - q^n$, resulting in the new solution f^{n+1} and q^{n+1}. Note that $\underline{f}^n$ and $\underline{q}^n$ are the auxiliary solutions to the auxiliary (partially linearized) problem CDA-PCL-SUE-R-PL; whereas f^n and q^n are the current solution to CDA-PCL-SUE-R at iteration *n*. The process is repeated with *n*=*n*+1 until a convergence criterion (see the employed convergence criterion in Section 3.3) terminates the solution of CDA-PCL-SUE-R.

It is noted that the algorithm described above is an instance of the partial linearization algorithm which is analyzed in *(15,17)*. According to Patriksson *(17)*'s Lemma 1 and Patriksson *(15)*'s Theorem 2.1, $\underline{f}^n = f^n$ holds if and only if f^n solves CDA-PCL-SUE-R; otherwise, the direction of $\underline{f}^n = f^n$ is a feasible descent direction with respect to the objective function *z* of CDA-PCL-SUE-R. Since *z* is strictly convex in *f*, the sequence { f^n } of route flows converges to the unique solution of the restricted master problem from any feasible initial flow f^0; see Patriksson *(15)*'s Theorem 2.2; the same property holds for the sequence $\{q^n\}$ of demands.

3.2. Phase II: Column Generation Problem

The partial linearization algorithm in Phase I solves the restricted master problem, given the subsets of routes between O-D pairs $\hat{K}_{rs} \subseteq K_{rs}\ \forall r \in R, s \in S$. The quality of travel pattern solution obtained from Phase I depends on the quality of $\hat{K}_{rs}$ in approximating K_{rs}. Damberg et al. *(18)* suggested and evaluated two route generation strategies based on the calculation of shortest paths given the solution of the restricted master problem. We adopt Damberg et al.'s first route generation strategy for Phase II. Routes are generated from the solution of shortest path problems based on the deterministic travel times; i.e. random components of travel times are temporarily ignored. At the solution to this restricted master problem, the link travel times are updated accordingly, and the subsets $\hat{K}_{rs} \subseteq K_{rs}\ \forall r \in R, s \in S$ are augmented by the generation of new routes using the shortest path algorithm.

It is worth noting that the algorithm is not guaranteed to converge to the unique optimal solution of CDA-PCL-SUE. However, it is guaranteed to solve the restriction of CDA-PCL-SUE to any set of routes generated. In the proposed algorithm, it terminates when the root mean square error of link flows and O-D flows from two successive iterations are within a user-specified tolerance.

3.3. Pseudocode of Proposed Simplicial Decomposition Algorithm

<u>Initialization</u>
Generate an initial path for each O-D pair
Step 1. Set $x_a^0 = 0, t_a^0 = t_a(x_a^0), \forall a \in A$ and $\hat{K}_{rs}^0 = \emptyset\ \forall r \in R, s \in S$.
Step 2. Set iteration counter: *n*=1.
Step 3. Solve the shortest path problem for all origins $\bar{k}_{rs}^n$.

$$\hat{K}_{rs}^n = \hat{K}_{rs}^{n-1} \cup \{\bar{k}_{rs}^n\}\ \ \forall r \in R, s \in S$$

Step 4. Perform all-or-nothing traffic assignment: $f_{\bar{k}_{rs}^n}^{rs^n} = q_{rs}^n\ \ \forall r \in R, s \in S$ where $q_{rs}^n = q_{rs}^1$ is an initial O-D demand obtained from input file such that it satisfies the origin flow constraints and the destination flow constraints.
Step 5. Assign path flows to links: $x_a^n = \sum_{r \in R} \sum_{s \in S} \sum_{k \in \hat{K}_{rs}^n} f_k^{rs^n} \cdot \delta_{ka}^{rs}\ \ \forall a \in A$

<u>CDA-PCL-SUE Solver</u>
Step 6. Increase iteration counter: *n*=*n*+1
Step 7. Update link travel time: $t_a^n = t_a(x_a^{n-1})\ \ \forall a \in A$
Step 8. Solve a shortest path problem: $\bar{k}_{rs}^n\ \ \forall r \in R, s \in S$
Step 9. Determine whether $\bar{k}_{rs}^n$ exists in the path set K_{rs}^{n-1}:

If $\bar{k}_{rs}^{n} \notin \hat{K}_{rs}^{n-1}$, then $\hat{K}_{rs}^{n} = \hat{K}_{rs}^{n-1} \cup \{\bar{k}_{rs}^{n}\}$.

Otherwise, $\hat{K}_{rs}^{n} = \hat{K}_{rs}^{n-1}$.

Step 10. Update route costs: $c_k^{rs^n} = \sum_{a \in A} \delta_{ka}^{rs} \cdot t_a(x_a^{n-1}) \ \ \forall k \in \hat{K}_{rs}^{n}, r \in R, s \in S$

Step 11. Compute similarity index between routes k and j:

$$\sigma_{kj}^{rs} = \frac{L_{kj}^{rs}}{\sqrt{L_k^{rs} \cdot L_j^{rs}}} \forall k = 1,\ldots,|\hat{K}_{rs}^{n}|; j = 1,\ldots,|\hat{K}_{rs}^{n}|; r \in R, s \in S$$

Where L_{kj}^{rs} is the length of the common part of route *k* and *j*.

Then, compute dissimilarity index between routes *k* and *j*: $\beta_{kj}^{rs} = 1 - \sigma_{kj}^{rs}$

Step 12. Compute γ_{rs}^{n} and ν_{rs}^{n} for $\forall r \in R, s \in S$ from Eq.(16.5)-(16.6):

Step 13. Compute the auxiliary O-D flows by Bregman's balancing method (the O-D trip-demand solution $\underline{q}_{rs}^{n}$ $\forall r \in R, s \in S$ to CDA-PCL-SUE-R-PL):

Initialization: $\underline{q}_{rs}^{0} = \exp(-1 - \gamma_{rs}^{n} / \nu_{rs}^{n}) \quad \forall r, s$

$t \leftarrow -1$ (*t* is iteration counter for the balancing method)

Do

{ Repeat the following steps for each origin and destination *i*

{ $\omega \leftarrow -1$, $\xi_0 \leftarrow 0$ and $t \leftarrow t+1$

Do

{ $\omega \leftarrow \omega + 1$

$$\xi_{\omega+1} = \xi_\omega - \frac{\sum_{rs} a_{i,rs} \cdot \underline{q}_{rs}^{t} \exp\left(\frac{\xi_\omega a_{i,rs}}{\nu_{rs}^{n}}\right) - O_i}{\sum_{s} \frac{1}{\nu_{is}^{n}} \cdot \underline{q}_{is}^{t} \exp\left(\frac{\xi_\omega}{\nu_{is}^{n}}\right)} \text{ if } i \text{ is an origin}$$

$$\xi_{\omega+1} = \xi_\omega - \frac{\sum_{rs} a_{i,rs} \cdot \underline{q}_{rs}^{t} \exp\left(\frac{\xi_\omega a_{i,rs}}{\nu_{rs}^{n}}\right) - D_i}{\sum_{r} \frac{1}{\nu_{ri}^{n}} \cdot \underline{q}_{ri}^{t} \exp\left(\frac{\xi_\omega}{\nu_{ri}^{n}}\right)} \text{ if } i \text{ is a destination}$$

}while($\frac{|\xi_{\omega+1} - \xi_\omega|}{\xi_\omega} > \varepsilon_{Newton}$)

$$\xi = \xi_{\omega+1} \text{ and } \underline{q}_{rs}^{t+1} = \underline{q}_{rs}^{t} \exp\left(\frac{\xi a_{i,rs}}{\nu_{rs}^{n}}\right) \quad \forall r \in R, s \in S.$$

}

}while ($\sqrt{\frac{1}{|R||S|} \sum_{r \in R} \sum_{s \in S} \left(\underline{q}_{rs}^{t+1} - \underline{q}_{rs}^{t}\right)^2} > \varepsilon_{Bregman}, \forall r, s$)

Set $\underline{q}_{rs}^{n} = \underline{q}_{rs}^{T}$ $\forall r \in R, s \in S$, where *T*=number of iterations for Bregman's balancing algorithm.

Step 14. Compute auxiliary route flows (the route-flow solution $\underline{f}_{k(kj)}^{rs^n}$ to CDA-PCL-SUE-R-PL):

$$\underline{f}_k^{rs^n} = P(k)_n \cdot \underline{q}_{rs}^n \quad \forall k \in \hat{K}_{rs}, r \in R, s \in S$$

where $P(k)_n = \sum_{\substack{j \in \hat{K}_{rs} \\ j \neq k}} P(kj)_n \cdot P(k \mid kj)_n$. $P(kj)_n$ and $P(k \mid kj)_n$ are determined from Eq.(16.7)-(16.8).

Step 15.Assign auxiliary path flows to auxiliary link flows:

$$\underline{x}_a^n = \sum_{r \in R} \sum_{s \in S} \sum_{k \in \hat{K}_{rs}} \delta_{ak}^{rs} \cdot \underline{f}_k^{rs^n} \quad \forall a \in A$$

Step 16.Perform line search using the golden section algorithm:

$$\min_{0 \le \alpha_{LS} \le 1} z_1(x^n + \alpha_{LS}(\underline{x}^n - x^n)) + z_2(f^n + \alpha_{LS}(\underline{f}^n - f^n)) + z_3(f^n + \alpha_{LS}(\underline{f}^n - f^n))$$

Step 17.Update path flow and demand:

$$f_k^{rs^n} = f_k^{rs^{n-1}} + \alpha_{LS}^n(\underline{f}_k^{rs^n} - f_k^{rs^{n-1}}) \quad \forall k \in \hat{K}_{rs}^n, r \in R, s \in S$$

$$q_{rs}^n = q_{rs}^{n-1} + \alpha_{LS}^n(\underline{q}_{rs}^n - q_{rs}^{n-1}) \qquad \forall r \in R, s \in S$$

Step 18.Assign route flows to links:

$$x_a^n = \sum_{r \in R} \sum_{s \in S} \sum_{k \in \hat{K}_{rs}^n} \delta_{ak}^{rs} \cdot f_k^{rs^n} \quad \forall a \in A$$

Step 19.Determine the root mean square error of the link and O-D flows:

$$RMSE = \sqrt{\frac{0.5}{|A|}\sum_{a \in A}\left(x_a^n - x_a^{n-1}\right)^2 + \frac{0.5}{|RS|}\sum_{rs \in RS}\left(q_{rs}^n - q_{rs}^{n-1}\right)^2}$$

Step 20.Check convergence: $RMSE \le \varepsilon_{Simplicial}$, then stop; otherwise go to Step 6.

4. ILLUSTRATIVE EXAMPLES

Two test networks with two congestion levels are first described. Then, the results from CDA-PCL-SUE are compared with those from CDA-MNL-SUE in order to see the effects of congestion, stochastic perception error and similarity in overlapping paths on the O-D flow, link flow and route flow allocations.

4.1. Two Test Networks

Network 1

Network 1 is a simple network with five nodes, eight links and four O-D pairs as shown in Figure 1. The Bureau Public Road link cost function is employed:

$$t_a(x_a) = t_a^0\left[1 + \alpha_a\left(\frac{x_a}{s_a}\right)^{\beta_a}\right]$$

The parameters t_a^0, s_a, α_a and β_a are also given in Figure 1, and the length of link a is set to t_a^0. Two congestion levels are considered as follows. For higher-congestion level (lower-congestion level), origin demands of origin nodes 1 and 2 are 45 and 50 trips (22 and 25 trips), respectively;

destination demands of destination nodes 4 and 5 are 35 and 60 trips (17 and 30 trips), respectively. The employed tolerances are $\varepsilon_{Simplicial}$ = $\varepsilon_{Bregman} = \varepsilon_{Newton}$ = $\varepsilon_{LineSearch}$ =0.001.

Network 2

Network 2 is a network used in *(19)* with 13 nodes, 19 links and four O-D pairs as shown in Figure 2. The link cost functions are linear:

$$t_a(x_a) = \alpha_a + \beta_a x_a$$

The parameters α_a and β_a are given in Figure 2, and the length of link a is set to α_a. Two congestion levels are considered. For higher-congestion level (lower-congestion level), origin demands of origin nodes 1 and 4 are 1,200 and 800 trips (600 and 400 trips), respectively; destination demands of destination nodes 2 and 3 are 1,000 and 1,000 trips (500 and 500 trips), respectively. The employed tolerances are $\varepsilon_{Simplicial}$ = 0.01; $\varepsilon_{Bregman} = \varepsilon_{Newton}$ = $\varepsilon_{LineSearch}$ =0.001.

4.2. Comparison of CDA-PCL-SUE and CDA-MNL-SUE Results

The CDA-MNL-SUE results are obtained from the algorithm in Lundgren and Patriksson (1998). The algorithms for both CDA-PCL-SUE and CDA-MNL-SUE are implemented in C. These run on a computer with 1.73 GHz Intel Core i7 processor and 4 GB of RAM, running under Windows 7. The CPU times of all runs on Networks 1 and 2 are within 1 minute. We compare the results from CDA-PCL-SUE and CDA-PCL-MNL to examine the effects of congestion, travelers' stochastic perception error and path similarity to simultaneously solve doubly-constrained trip distribution problem and stochastic user equilibrium problem.

The dispersion parameters are set at various values for two congestion levels on both networks. The differences in O-D flows and link flows from the two combined distribution and assignment solutions is measured by the root mean square errors:

$$RMSE_L = \sqrt{\frac{1}{|A|}\sum_{a\in A}(x^*_{a,PCL} - x^*_{a,MNL})^2}$$

where $x^*_{a,PCL}$ and $x^*_{a,MNL}$ are the converged link flows in CDA-PCL-SUE and CDA-MNL-SUE, respectively.

$$RMSE_{OD} = \sqrt{\frac{1}{|RS|}\sum_{rs\in RS}(q^*_{rs,PCL} - q^*_{rs,MNL})^2}$$

where $q^*_{rs,PCL}$ and $q^*_{rs,MNL}$ are the converged O-D flows in CDA-PCL-SUE and CDA-MNL-SUE, respectively. Figure 3 shows the values of $RMSE_L$ and $RMSE_{OD}$ with various dispersion factors at two congestion levels on Networks 1 and 2. $RMSE_{OD}$ appears fluctuated at the higher-congestion level on both networks, whereas at the lower-congestion level $RMSE_{OD}$ appears smooth over the dispersion factors. At both congestion levels on both networks, $RMSE_L$ decreases with the increase of the dispersion factor. The decrease rate of $RMSE_L$ is greater when the dispersion factor is close to 0, and the decrease rate at the higher-congestion level is greater than that of the lower-congestion level on both networks. Based on our empirical results, the link flow patterns from CDA-PCL-SUE and CDA-MNL-SUE are closer as the dispersion factor increases on both congestion levels. The O-D flow patterns from both models differ in different degree over various dispersion factors. Figure 4 shows the converged O-D flows of CDA-PCL-SUE and CDA-MNL-SUE models on Networks 1 at the higher congestion level. Figure 5 shows the converged link flows from CDA-PCL-SUE and CDA-MNL-SUE on Networks 1 at the higher congestion level. This figure reiterates the findings that the link flow patterns from the two models are more identical at higher dispersion factors. Note that the similar graphs for Network 2 are not shown due to the space limit.

Since the proposed algorithm employs the column generation phase to generate paths, it is possible that the generated paths from CDA-PCL-SUE are not the same as those from CDA-MNL-SUE. Then, it may not be comparable in terms of route flows. However, we found that the dispersion factor of 0.125 yields the same path set in both models on both networks. Thus, this is employed for path flow comparison on both networks. Table 1 shows the path flow results obtained from CDA-PCL-SUE and CDA-MNL-SUE on Networks 1 and 2 at the higher-congestion level. As can be observed in Table 1, the path costs for each O-D pair in both CDA-PCL-SUE and CDA-MNL-SUE on both networks are not equal, and both models disperse travel demands to many paths for each O-D pair on the two test networks. These are the effects of travelers' stochastic perception error captured by both models. For each O-D pair, the similarity index is calculated for each route pair connecting this O-D pair. The similarity index of each route pair is completely independent of that of other route pairs. Prashker and Bekhor *(9)* indicated that this property is highly desirable for route choice models. Table 1 shows the average similarity index for each route, which is the mean value of all similarity indices involving this route. For example, on network 1, there are three routes in the generated path set for O-D 1-4, yielding three unordered route pairs. The similarity index between routes 1 and 2 is 0 and that between routes 1 and 3 is 0.4396. The average similarity index of route 1 connecting O-D 1-4 is the mean value of 0 and 0.4396, yielding 0.2198.

CDA-PCL-SUE generally considers a route with a high value of similarity as less attractive in route flow allocation. CDA-PCL-SUE accounts for the overlapping paths in route choice such that a path with a higher value of average similarity index and higher path cost will be assigned less flows. As can be seen in Table 1 for Network 1, in the CDA-MNL-SUE model, the cost of path 3 is 7.76% and 5.83% higher than paths 1 and 2, and assigns less flows to paths 3 (85.05% and 88.34% of flows assigned to paths 1 and 2, respectively). In contrast, CDA-PCL-SUE accounts for the path overlapping effect. The average similarity index of path 3 of O-D 1-4 is 101.09% higher than paths 1 and 2 connecting this O-D pair, and in the CDA-PCL-SUE model the cost of path 3 is 4.09% and 2.68% higher than paths 1 and 2. Then, CDA-PCL-SUE assigns much less flows to path 3 (51.82% and 53.24% of flows assigned to paths 1 and 2, respectively) than CDA-MNL-SUE does. Similar results can be observed on Network 2 in Table 1.

Table 2 shows the O-D flow results of the two models on Networks 1 and 2. Apparently, the O-D flows are distributed differently in the two models on both networks. As can be seen in Table 2, the total O-D flows out of each origin in both models are the same, and the total O-D flows into each destination in both models are equal. These are due to the doubly constrained trip distribution embedded in the two models. Table 2 also shows the average similarity index for each O-D pair, which is the mean value of the average similarity indices for all paths connecting this O-D pair. The weighted average path cost for each O-D pair is calculated by the summation of the products of path costs and route choice probabilities. We will explore the results to check whether we can relate the attractiveness of an O-D pair in doubly-constrained O-

D trip distribution in CDA-PCL-SUE to the average similarity index for each O-D pair and the weighted average path cost of each O-D pair. On Network 1, we consider the O-D flow distribution for origin node 1. From Table 2, the weighted average path cost of O-D 1-4 in CDA-PCL-SUE is 19.78% higher than that of O-D 1-5, whereas in CDA-MNL-SUE it is 23.09% higher. The average similarity index of O-D 1-4 is 31.26% higher than O-D 1-5. The O-D flows allocated to O-D 1-5 is 14.00% higher than O-D 1-4 in CDA-PCL-SUE, whereas in CDA-MNL-SUE, it is 41.34% higher. It seems that CDA-PCL-SUE may assign more flows to O-D 1-4 with higher similarity index than CDA-MNL-SUE does. Next, we consider the O-D flow distribution for destination node 5. The weighted average path cost of O-D 1-5 in CDA-PCL-SUE is 45.99% higher than that of O-D 2-5, whereas in CDA-MNL-SUE it is 50.06% higher. The average similarity index of O-D 1-5 is 284.71% higher than O-D 2-5. The O-D flows allocated to O-D 2-5 is 50.29% higher than O-D 1-5 in CDA-PCL-SUE, whereas in CDA-MNL-SUE, it is 27.67% higher. In this case, CDA-PCL-SUE assigns less flow to O-D 1-5 with higher similarity index than CDA-MNL-SUE does. Apparently, we cannot conclude how CDA-PCL-SUE distributes O-D flows among different O-D pairs, given weighted average path cost and average similarity index. Similar observations can be found in Table 2 for Network 2. This is because CDA-PCL-SUE also has the origin flow balance constraints and destination flow balance constraints that must be satisfied. In fact, the trip distribution in CDA-PCL-SUE can be determined by Eq.(10); i.e. it is based on the path costs, dispersion factor, dual variables of origin and destination flow balance constraints, and similarity indices. The average similarity indices and weighted average path costs are not directly employed in determining the trip distribution.

Table 3 shows the link flow results on Networks 1 and 2. The traffic flow patterns are different as the two models have different objective functions used in the trip distribution and route choice to capture the effects of congestion, stochastic perception error and path overlapping. Links with more paths passing through mostly have smaller flows assigned by CDA-PCL-SUE when compared with CDA-MNL-SUE such as links 1, 3, 4, 6 and 8 on Network 1; and links 1, 3, 6, 10, 12, 14, 15, 16, and 18 on Network 2. CDA-PCL-SUE assigns less number of flows to these links than CDA-MNL-SUE does.

5. SUMMARY AND CONCLUSIONS

We proposed the equivalent mathematical formulation (CDA-PCL-SUE) that combines the doubly-constrained gravity-model based trip distribution and the paired-combinatorial-logit stochastic user equilibrium assignment. The first-order conditions were derived to show that these conditions are equivalent to the paired-combinatorial-logit stochastic user equilibrium equations and doubly-constrained gravity equations. The proposed solution method for CDA-PCL-SUE is a disaggregate simplicial decomposition algorithm that iterates between two phases until convergence. Phase I approximately solves the restriction of CDA-PCL-SUE by the partial linearization descent algorithm. Phase I iterates three sub-phases until convergence: Phases I.1, I.2 and I.3. Phase I.1 is the entropy maximization problem that is solved by Bregman's balancing algorithm to obtain the auxiliary O-D flows. Phase I.2 applies the paired-combinatorial-logit formula to determine the auxiliary route flows. Phase I.3 performs the line search to obtain the next solution to the restriction of CDA-PCL-SUE. After achieving an approximate solution to the restriction of CDA-PCL-SUE, Phase 2 generates a new set of shortest paths in order to augment the path set used in Phase 1 in the next iteration.

In our illustrative example, two test networks with two congestion levels are employed. Network 1 is a simple network, and Network 2 is the network in *(19)*. The proposed algorithm is employed to determine the O-D flows, link flows and route flows on the two networks at two congestion levels. The results from CDA-PCL-SUE are compared to those from CDA-MNL-SUE in order to illustrate how CDA-PCL-SUE distributes O-D flows and route flows when accounting for similarity effects in addition to the congestion effect and stochastic-perception-error effect. When varying dispersion factors on the two test networks at two congestion levels, we found that the O-D flow patterns from CDA-PCL-SUE and CDA-MNL-SUE differ in different degree such that the relationship of O-D flow difference and dispersion factor cannot be concluded. The link flow patterns from CDA-PCL-SUE and CDA-MNL-SUE are more identical at higher dispersion factors on the two test networks at both congestion levels.

At the dispersion factor of 0.125 where the generated path sets from CDA-PCL-SUE and CDA-MNL-SUE are the same, the path flow patterns from the two models on the two test networks at the higher congestion level are compared. We illustrated that CDA-PCL-SUE assigns less flows to a path with a higher average similarity index and higher path cost than CDA-MNL-SUE does because CDA-PCL-SUE considers the similarity effect whereas CDA-MNL-SUE does not. That is, CDA-PCL-SUE generally considers a route with a high value of similarity as less attractive in route flow allocation, whereas CDA-MNL-SUE does not.

Next, we cannot conclude the relationship between the O-D flow allocation and the average similarity indices and weighted average path costs for O-D pairs for CDA-PCL-SUE. This is because the average similarity indices and weighted average path costs for O-D pairs are not directly employed in the doubly-constrained gravity-based trip distribution equations. In terms of link flow patterns, we found that CDA-PCL-SUE generally assigns less flows to links with more paths passing through than CDA-MNL-SUE. This reiterates that CDA-PCL-SUE can account for similarity in path overlapping while CDA-MNL-SUE cannot.

The future research directions are the following. The proposed CDA-PCL-SUE and solution method can be used to better represent the lower-level problem in modeling capacity flexibility of transport networks in *(20)* by substituting CDA-MNL-SUE with CDA-PCL-SUE so that it can account for similarity in path overlapping. The model can be modified for the singly-constrained gravity-based trip distribution version. Furthermore, the proposed model can be extended to incorporate trip generation and modal split. Other extended logit models such as cross-nested logit and generalized nested logit may be employed in the combined distribution and assignment.

ACKNOWLEDGMENTS

The first author gratefully acknowledges the support from The Thailand Research Fund under contract number MRG5380066 and King Mongkut's University of Technology Thonburi Research Fund. The second author would like to acknowledge the National Science Council, Taiwan, R.O.C. for the funding support under project number NSC 100-2410-H-006-069-MY3.

APPENDIX: PROOF OF VALID APPLICATION OF BREGMAN'S BALANCING METHOD IN PHASE I.1

The entropy maximization problem (16.1)-(16.8) can be rewritten as

$$\min f(\underline{q}) = \sum_{rs \in RS} \gamma_{rs}^{n} \cdot \underline{q_{rs}} + v_{rs}^{n} \cdot \underline{q_{rs}} \cdot \ln \underline{q_{rs}} \tag{A.1}$$

$$\text{subject to } \sum_{rs \in RS} a_{i,rs} \underline{q_{rs}} = b_i \quad \forall i = 1,2,...,m \tag{A.2}$$

$$\underline{q} \in E^{|RS|}, \underline{q} > 0 \tag{A.3}$$

where RS is the set of O-D pairs; m is the total number of origins and destinations (i.e. $|S|+|R|$). $a_{i,rs}$ equals to 1 if i is the origin of O-D pair r-s and equals to 0 otherwise for i=1,2,…,$|S|$; $a_{i,rs}$ equals to 1 if $i-|S|$ is the destination of O-D pair r-s and equals to 0 otherwise for i=$|S|$+1 ,…, $|S|+|R|$. E is the set of real numbers. Denote by S the set $\{\underline{q} \in E^{|RS|} \mid \underline{q} > 0\}$ and $\gamma_{rs}^{n} > 0, v_{rs}^{n} > 0$ $\forall rs \in RS$. The constraint set (A.2) can be written as $A\underline{q} = b$ where A is the coefficient matrix or $A_{i.}\underline{q} = b_i \quad \forall i = 1,2,...,m$, where $A_{i.}$ is the i^{th}-row of the coefficient matrix A.

The function $f(\underline{q}) = \sum_{rs \in RS} \gamma_{rs}^{n} \cdot \underline{q_{rs}} + v_{rs}^{n} \cdot \underline{q_{rs}} \cdot \ln \underline{q_{rs}}$ is a strictly convex differentiable function over the convex set S. Bregman *(21)* showed that if $f(\underline{q})$ is a strictly convex differentiable function over the convex set $S \subset E^{|RS|}$, the function $D(\underline{q}, \underline{p}) = f(\underline{q}) - f(\underline{p}) - \nabla f(\underline{p}) \cdot (\underline{q} - \underline{p})$ satisfies conditions (i)-(iv) in *(21)*. Condition (vi) and condition 1 in *(21)* are satisfied if S is closed and $f(\underline{q})$ is continuously differentiable. Substituting $f(\underline{q}) = \sum_{rs \in RS} \gamma_{rs}^{n} \cdot \underline{q_{rs}} + v_{rs}^{n} \cdot \underline{q_{rs}} \cdot \ln \underline{q_{rs}}$ in the function $D(\underline{q}, \underline{p})$ yields:

$$D(\underline{q}, \underline{p}) = \sum_{rs \in RS} v_{rs}^{n} \underline{p_{rs}} - v_{rs}^{n} \underline{q_{rs}} + v_{rs}^{n} \underline{q_{rs}} (\ln \underline{q_{rs}} - \ln \underline{p_{rs}}) \tag{A.4}$$

(21) showed that $\sum_{j=1}^{p} v_j y_j - v_j x_j + v_j x_j (\ln x_j - \ln y_j)$ with $v_j = 1, \forall j$ satisfies condition (v) in *(21)*. Clearly, $D(\underline{q}, \underline{p}) = \sum_{rs \in RS} v_{rs}^{n} \underline{p_{rs}} - v_{rs}^{n} \underline{q_{rs}} + v_{rs}^{n} \underline{q_{rs}} (\ln \underline{q_{rs}} - \ln \underline{p_{rs}})$ with v_{rs}^{n} >0 $\forall rs \in RS$ also satisfies condition (v) in *(21)*. Thus, the function in (A.4) satisfies conditions (i)-(vi) and condition 1 in *(21)*. Bregman's balancing method is valid for the function $D(\underline{q}, \underline{p})$ defined over $S \times S$ and satisfying conditions (i)-(vi). Condition 1 is satisfied; then, the relaxation sequence $\{\underline{q^t}\}$ has a unique limiting point $\underline{q^*} \in S$.

Given $\underline{q^t} \in S$, $\underline{q^{t+1}}$ which is the D-projection of the point $\underline{q^t}$ onto the set $A_{i_t(\underline{q^t})} = \{\underline{q} \in E^{|RS|} \mid \sum_{rs \in RS} a_{i_t(\underline{q^t}),rs} \underline{q_{rs}} = b_{i_t(\underline{q^t})}\}$ can be determined (where $i_t(\underline{q^t})$ is the index from the control of relaxation $\{ i_0(\underline{q^0}), i_1(\underline{q^1}),... \}$):

$$D(\underline{q^{t+1}}, \underline{q^t}) = \min_{z \in A_i \cap S} D(z, \underline{q^t}) \quad \text{where } A_i = \{\underline{q} \in E^{|RS|} \mid \sum_{rs \in RS} a_{i,rs} \underline{q_{rs}} = b_i\}$$

$$= \min D(z, \underline{q}^t) = f(z) - f(\underline{q}^t) - \nabla f(\underline{q}^t)(z - \underline{q}^t)$$

$$\text{subject to } \sum_{rs \in RS} a_{i,rs} z_{rs} = b_i$$

$$z_{rs} \geq 0 \quad \forall rs \in RS$$

Since $f(\underline{q}^t)$ and $\nabla f(\underline{q}^t) \cdot \underline{q}^t$ are constant, they can be dropped from the objective function without affecting the optimal solution. Lagrangian of this program is

$$L(z, \xi) = f(z) - \nabla f(\underline{q}^t) \cdot z + \xi(b_i - \sum_{rs \in RS} a_{i,rs} z_{rs})$$

$$\text{subject to } z_{rs} \geq 0 \quad \forall rs \in RS$$

The first-order conditions of the Lagrangian problem are:

$$\nabla_z L(\underline{q}^{t+1}, \xi) \geq 0 \tag{A.5a}$$

$$\underline{q}^{t+1} \cdot \nabla_z L(\underline{q}^{t+1}, \xi) = 0 \tag{A.5b}$$

$$\nabla_\lambda L(\underline{q}^{t+1}, \xi) = 0 \tag{A.5c}$$

$$\underline{q}_{rs}^{t+1} \geq 0 \ \forall rs \in RS \tag{A.5d}$$

From (A.5a) and (A.5b), if $\underline{q}^{t+1}$ >0, then

$$\nabla_z L(\underline{q}^{t+1}, \xi) = \nabla f(\underline{q}^{t+1}) - \nabla f(\underline{q}^t) - \xi A_{i\cdot} = 0 \tag{A.6}$$

Since $\nabla f(\underline{q}) = \gamma_{rs}^n + v_{rs}^n + v_{rs}^n \ln \underline{q}_{rs}$, then (A.6) becomes

$$\underline{q}_{rs}^{t+1} = \underline{q}_{rs}^t \exp(\frac{\xi a_{i,rs}}{v_{rs}^n}) \quad \forall rs \tag{A.7}$$

The equation (A.5c) is

$$\sum_{rs \in RS} a_{i,rs} \underline{q}_{rs}^{t+1} = b_i \tag{A.8}$$

The substitution of (A.7) into (A.8) yields

$$\sum_{rs \in RS} a_{i,rs} \cdot \underline{q}_{rs}^t \exp(\frac{\xi a_{i,rs}}{v_{rs}^n}) = b_i \tag{A.9}$$

where $b_i = O_r$ and $\xi = \alpha_r$ if constraint i is associated with origin r; $b_i = D_s$ and $\xi = \lambda_s$ if constraint i is associated with destination s.

As can be seen from (A.7), if $\underline{q}_{rs}^t \in S$, then also $\underline{q}_{rs}^{t+1} \in S$ (recall $S = \{\underline{q} \in E^{|RS|} \mid \underline{q} > 0\}$). Consequently, with a suitable relaxation control, the conditions of Theorem 3 in *(21)* are satisfied. Therefore, the relaxation sequence $\{\underline{q}^t\}$ obtained from the balancing process will converge to the point $\underline{q}^*$, that is, to the solution of the entropy maximization problem in Phase I.1, if the point of absolute minimum of the function (11.1a), i.e. $\underline{q}_{rs}^0 \ \forall rs \in RS$, is chosen as the initial approximation. $\underline{q}_{rs}^0$ are determined from $\nabla f(\underline{q}^0) = 0$. Thus,

$$\underline{q}_{rs}^0 = \exp(-1 - \gamma_{rs}^n / v_{rs}^n) \ \ \forall rs \in RS \tag{A.10}$$

REFERENCES

1. Zhou, Z., A. Chen, and S.C. Wong (2009) Alternative formulations of a combined trip generation, trip distribution, modal split, and trip assignment model, *European Journal of Operational Research*, Vol.198, No.1, 129-138.
2. Boyce, D. and H. Bar-Gera (2004) Multiclass combined models for urban travel forecasting, *Network and Spatial Economics*, Vol.4, 115-124.
3. Evans, S.P. (1976) Derivation and analysis of some models for combining trip distribution and assignment, *Transportation Research*, Vol.10, 37-57.
4. Erlander, S. (1990) Efficient behavior and the simultaneous choices of origins, destinations and routes, *Transportation Research Part B*, Vol.24, 363-373.
5. Lundgren, J.T. and M. Patriksson (1998) The combined distribution and stochastic assignment problem, *Annals of Operations Research*, Vol.82, 309-329.
6. Daganzo, C.F. and Y. Sheffi (1977) On Stochastic Models of Traffic Assignment, *Transportation Science*, Vol.11, 253-274.
7. Cascetta, E., A. Nuzzolo, F. Russo, and A. Vitetta (1996) A modified logit route choice model overcoming path overlapping problems: specification and some calibration results for interurban networks. In Proceedings of the 13th International Symposium on Transportation and Traffic Theory, Pergamon, Lyon, France, 697-711.
8. Ben-Akiva, M. and M. Bierlaire (1999) Discrete choice methods and their applications to short term travel decisions. In Handbook of Transportation Science, Kluwer, Dordrecht, The Netherlands, 5-12.
9. Prashker, J.N. and S. Bekhor (2000) Congestion, Stochastic, and Similarity Effects in Stochastic User-Equilibrium Models, *Transportation Research Record 1733*, 80-87.
10. Chen, A., P. Kasikitwiwat, and Z. Ji (2003) Solving the overlapping problem in route choice with paired combinatorial logit model, *Transportation Research Record 1857*, 65-73.
11. Bekhor, S., L. Reznikova and T. Toledo (2007) Application of the cross nested logit route choice model in stochastic user equilibrium traffic assignment, *Presented at the 86th Annual meeting of the Transportation Research Board*, Washington, D.C., 2007.
12. Bekhor, S., M.E. Ben-Akiva, and M. S. Ramming (2002) Adaptation of logit kernel to route choice situation, *Transportation Research Record 1805*, 78-85.
13. Vovsha, P. and S. Bekhor (1998) Link-Nested Logit Model of Route Choice: Overcoming Route Overlapping Problem, *Transportation Research Record 1645*, 133-142.
14. Bekhor, S. and J. N. Prashker (2001) Stochastic user equilibrium formulation for generalized nested logit model, *Transportation Research Record 1752*, 84-90.
15. Patriksson, M. (1993a) Partial linearization methods in nonlinear programming, *Journal of Optimization Theory and Applications*, Vol.78, 227-246.
16. Lamond, B. and N.F. Stewart (1981) Bregman's Balancing Method, *Transportation Research Part B*, Vol.15, No.4, 239-248.

17. Patriksson, M. (1993b) A unified description of iterative algorithms for traffic equilibria, *European Journal of Operational Research*, Vol.17, 154-176.

18. Damberg, O., J.T. Lundgren and M. Patriksson (1996) An algorithm for the stochastic user equilibrium problem, *Transportation Research Part B*, Vol.30, 115-131.

19. Nguyen, S. and D. Dupuis (1984) An Efficient Method for Computing Traffic Equilibria in Networks with Asymmetric Transportation Costs, *Transportation Science*, Vol.18, 185-202.

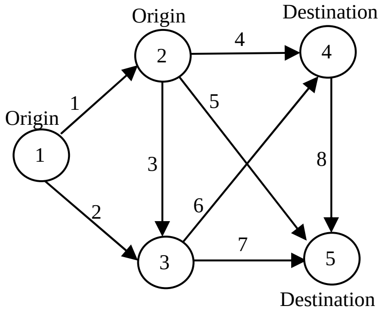
20. Chen, A. and P. Kasikitwiwat (2011) Modeling capacity flexibility of transportation networks, *Transportation Research Part A: Policy and Practice*, Volume 45, Issue 2, 105-117.

21. Bregman, L.M. (1967) The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming, *U.S.S.R. Computational Mathematics and Mathematical Physics*, Vol.7, 200-217.

22. Larsson, T. and M. Patriksson (1992) Simplicial decomposition with disaggregated representation for the traffic assignment problem, *Transportation Science*, Vol.26, 4-17.

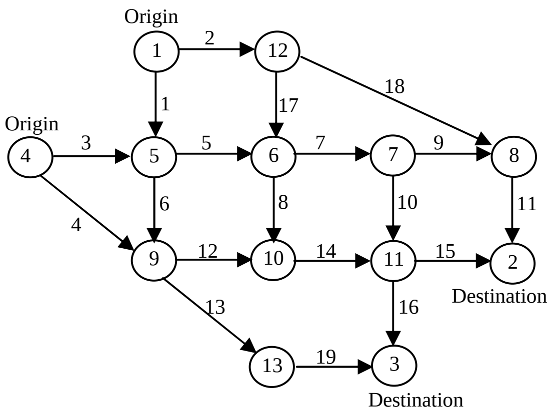
23. Pravinvongvuth, S. and A. Chen (2005) Adaptation of the Paired Combinatorial Logit Model to the Route Choice Problem, *Transportmetrica*, Vol.1, No.3, 223-240.

LIST OF FIGURES AND TABLES



Link	s_a	t_a^0	α_a	β_a
(1,2)	25	4.0	0.15	4.0
(1,3)	25	5.2	0.15	4.0
(2,3)	30	1.0	0.15	4.0
(2,4)	15	5.0	0.15	4.0
(2,5)	15	5.0	0.15	4.0
(3,4)	15	4.0	0.15	4.0
(3,5)	15	4.0	0.15	4.0
(4,5)	30	1.0	0.15	4.0

Figure 1 Test Network 1



Link	α_a	β_a
(1,5)	7	0.0125
(1,12)	9	0.01
(4,5)	9	0.01
(4,9)	12	0.1
(5,6)	3	0.0075
(5,9)	9	0.0075
(6,7)	5	0.0125
(6,10)	13	0.005
(7,8)	5	0.0125
(7,11)	9	0.0125

Link	α_a	β_a
(8,2)	9	0.0125
(9,10)	10	0.005
(9,13)	9	0.1
(10,11)	6	0.0025
(11,2)	9	0.005
(11,3)	8	0.01
(12,6)	7	0.0125
(12,8)	14	0.01
(13,3)	11	0.01

Figure 2 Test Network 2

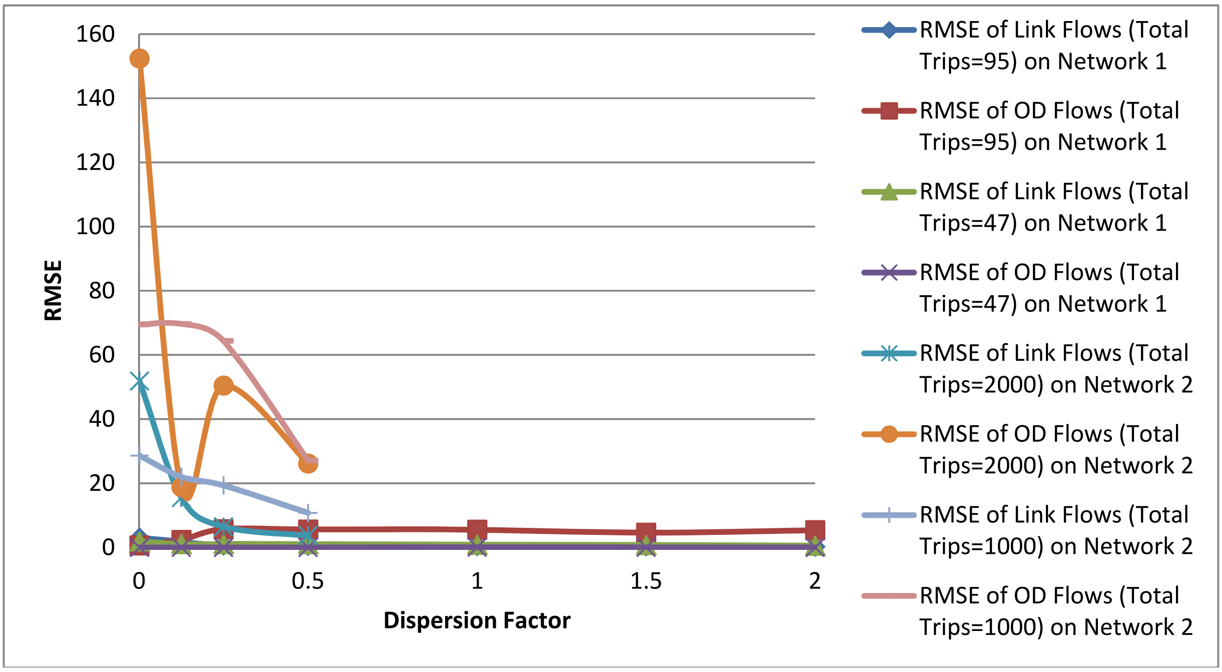


Figure 3 Root Mean Square Errors on Two Test Networks

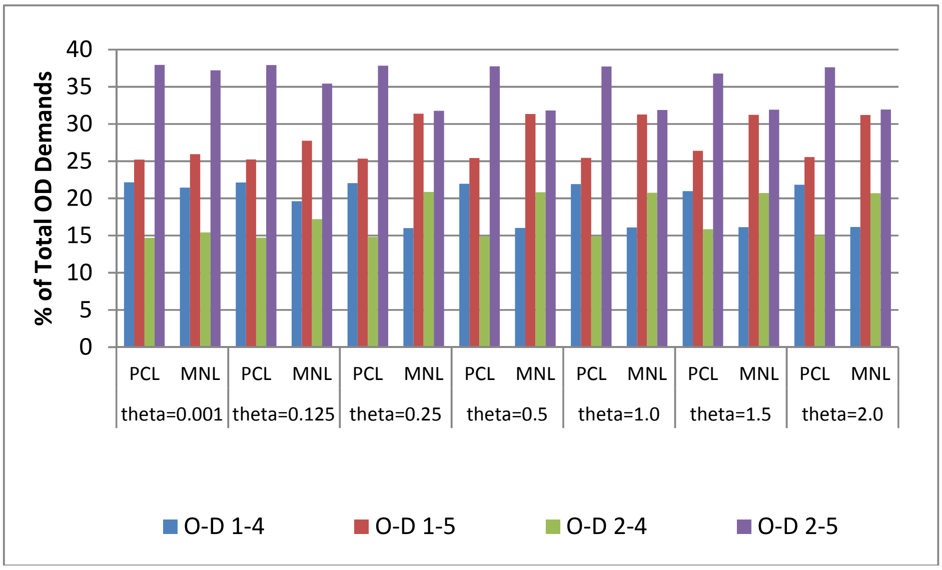


Figure 4 O-D Demands on Network 1 (Higher-Congestion Level)

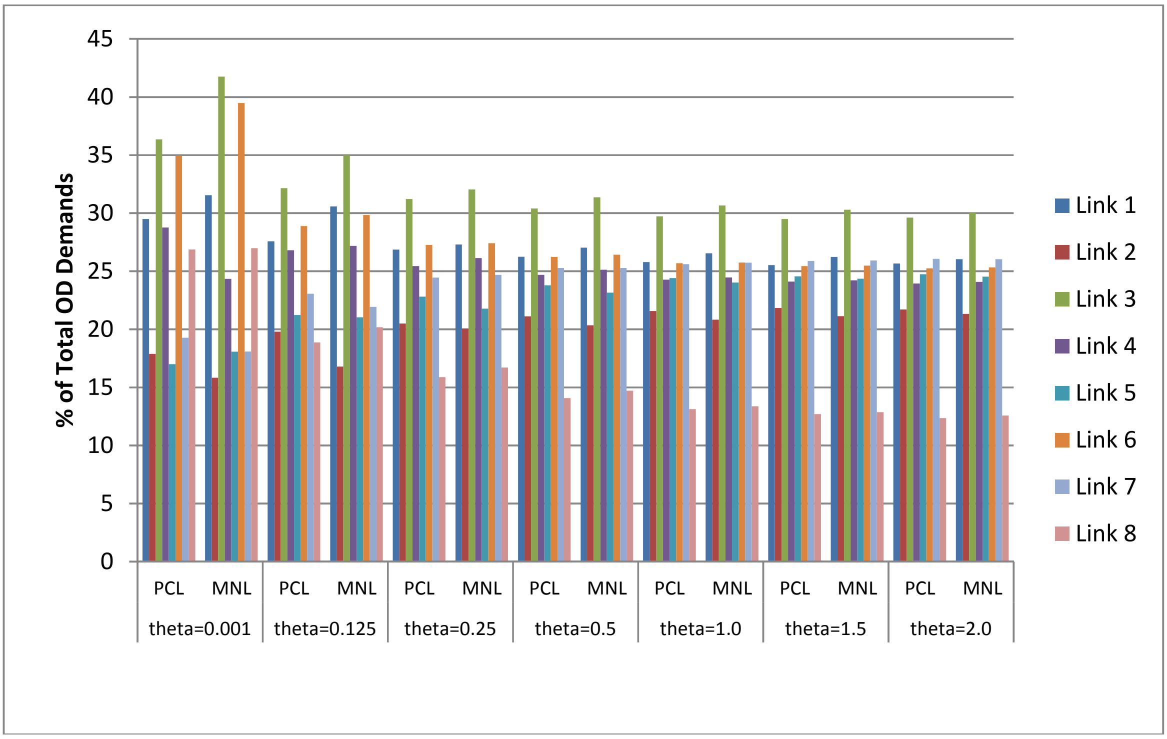


Figure 5 Link Flows on Network 1 (Higher-Congestion Level)

Table 1 Path Flow Results of CDA-PCL-SUE and CDA-MNL-SUE on Two Test Networks

O-D	Path No.	Link Sequence	Average Similarity Index	Route Choice Probability		Path Flow		Path Cost	
				PCL	MNL	PCL	MNL	PCL	MNL
Network 1 (Total O-D Demand = 90 Trips, Dispersion Factor = 0.125)									
1-4	1	1-4	0.2198	0.4014	0.3555	8.4398	6.6279	15.9593	16.6825
	2	2-6	0.2198	0.3907	0.3422	8.2145	6.3813	16.1777	16.9872
	3	1-3-6	0.4420	0.2079	0.3023	4.3734	5.6373	16.6120	17.9777
1-5	1	1-5	0.2811	0.2984	0.3250	7.1710	8.5644	12.1739	12.4549
	2	2-7	0.0000	0.4415	0.3632	10.5941	9.5702	12.1817	11.5661
	3	1-3-6-8	0.3072	0.1231	0.1433	2.9374	3.7782	17.6311	19.0027
	4	1-4-8	0.3072	0.1370	0.1685	3.2699	4.4407	16.9784	17.7075
2-4	1	4	0.0000	0.5204	0.5404	7.2707	8.8368	11.2365	11.5888
	2	3-6	0.0000	0.4796	0.4596	6.7017	7.5167	11.8892	12.8840
2-5	1	5	0.0000	0.3600	0.3393	12.9972	11.4164	7.4511	7.3613
	2	3-7	0.0609	0.3136	0.3350	11.3171	11.2729	7.8932	7.4630
	3	3-6-8	0.1164	0.1455	0.1497	5.2219	5.0370	12.9083	13.9091
	4	4-8	0.0556	0.1809	0.1760	6.4915	5.9201	12.2556	12.6139
Network 2 (Total O-D Demand = 2000 Trips, Dispersion Factor = 0.125)									
1-2	1	1-5-7-9-11	0.1477	0.1856	0.1928	103.55	115.60	59.08	58.62
	2	2-17-8-14-15	0.1199	0.1806	0.1696	96.86	101.71	59.75	59.64
	3	2-18-11	0.2676	0.6338	0.6376	380.31	382.25	49.26	49.05
1-3	1	1-5-7-10-16	0.3411	0.2269	0.2079	173.32	124.86	63.67	63.93
	2	2-17-8-14-16	0.3075	0.2089	0.1843	122.53	110.64	64.90	64.89
	3	1-6-13-19	0.2049	0.1927	0.1830	115.41	109.83	66.68	64.95
	4	1-5-8-14-16	0.4844	0.1640	0.2092	129.15	125.65	63.79	63.88
	5	1-6-12-14-16	0.4311	0.2076	0.2156	78.87	129.47	63.01	63.64
4-2	1	3-5-7-9-11	0.2758	0.2564	0.2110	112.87	84.51	59.74	59.23
	2	3-6-12-14-15	0.4210	0.2021	0.2173	92.53	87.03	58.52	58.99
	3	4-12-14-15	0.3167	0.1771	0.1513	79.57	60.58	61.71	61.89
	4	3-5-8-14-15	0.4677	0.1756	0.2109	80.52	84.43	59.30	59.23
	5	3-5-7-10-15	0.3899	0.1888	0.2095	53.79	83.89	59.18	59.28
4-3	1	4-13-19	0.0884	0.1654	0.1399	68.02	55.89	70.53	68.46
	2	3-6-12-14-16	0.4088	0.2320	0.2369	113.69	94.68	63.67	64.25
	3	3-5-8-14-16	0.3728	0.2165	0.2299	103.32	91.85	64.45	64.49
	4	3-5-7-10-16	0.3069	0.2496	0.2284	59.69	91.26	64.33	64.54
	5	4-12-14-16	0.3933	0.1365	0.1649	35.99	65.87	66.86	67.14

Table 2 O-D Flow Results of CDA-PCL-SUE and CDA-MNL-SUE on Two Test Networks

O-D No.	Origin Node	Destination Node	Average Similarity Index	Weighted Average Path Cost		O-D Flow	
				PCL	MNL	PCL	MNL
Network 1 (Total O-D Demand = 90 Trips, Dispersion Factor = 0.125)							
1	1	4	0.2939	16.180	17.178	21.028	18.646
2	1	5	0.2239	13.508	13.956	23.972	26.354
3	2	4	0.0000	11.550	12.184	13.972	16.354
4	2	5	0.0582	9.253	9.300	36.028	33.646
Network 2 (Total O-D Demand = 2000 Trips, Dispersion Factor = 0.125)							
1	1	2	0.1784	52.977	52.689	580.71	599.56
2	1	3	0.3538	64.389	64.220	619.29	600.44
3	4	2	0.3742	59.658	59.591	419.29	400.44
4	4	3	0.3140	65.573	65.435	380.71	399.56

Table 3 Link Flow Results of CDA-PCL-SUE and CDA-MNL-SUE on Two Test Networks

Link	Number of Paths Passing Through*				Link Flow		Link Cost	
	O-D 1	O-D 2	O-D 3	O-D 4	PCL	MNL	PCL	MNL
Network 1 (Total O-D Demand = 90 Trips, Dispersion Factor = 0.125)								
1	2	3	0	0	26.191	29.04858	4.723	5.09364
2	1	1	0	0	18.809	15.95142	5.450	5.32929
3	1	1	1	2	30.551	33.24212	1.161	1.22616
4	1	1	1	1	25.472	25.82563	11.236	11.5888
5	0	1	0	1	20.168	19.98084	7.451	7.36131
6	2	1	1	1	27.449	28.3505	10.728	11.6579
7	0	1	0	1	21.911	20.84303	6.732	6.23683
8	0	2	0	2	17.921	19.17613	1.019	1.02504
Network 2 (Total O-D Demand = 2000 Trips, Dispersion Factor = 0.125)								
1	1	4	0	0	600.30	605.40	14.50	14.57
2	2	1	0	0	599.70	594.60	15.00	14.95
3	0	0	4	3	616.42	617.65	15.16	15.18
4	0	0	1	2	183.58	182.35	30.36	30.23
5	1	2	3	2	816.21	802.05	9.12	9.02
6	0	2	1	1	400.51	421.01	12.00	12.16
7	1	1	2	1	503.22	500.12	11.29	11.25
8	1	2	1	1	532.38	514.27	15.66	15.57
9	1	0	1	0	216.42	200.11	7.71	7.50
10	0	1	1	1	286.80	300.02	12.58	12.75
11	2	0	1	0	596.72	582.36	16.46	16.28
12	0	1	2	2	400.66	437.63	12.00	12.19
13	0	1	0	1	183.43	165.72	27.34	25.57
14	1	3	3	3	933.04	951.90	8.33	8.38
15	1	0	4	0	403.28	417.64	11.02	11.09
16	0	4	0	4	816.57	834.28	16.17	16.34
17	1	1	0	0	219.39	212.35	9.74	9.65
18	1	0	0	0	380.31	382.25	17.80	17.82
19	0	1	0	1	183.43	165.72	12.83	12.66

* Note: For Network 1, O-D 1, 2, 3 and 4 are 1-4, 1-5, 2-4 and 2-5, respectively.
For Network 2, O-D 1, 2, 3 and 4 are 1-2, 1-3, 4-2 and 4-3, respectively.

TABU SEARCH HEURISTICS FOR INVENTORY ROUTING PROBLEM WITH ROUTE DURATION LIMITS AND STOCHASTIC INVENTORY CAPACITY CONSTRAINTS

by

Ampol Karoonsoontawong*, Ph.D.
Assistant Professor
Department of Civil Engineering, Faculty of Engineering
King Mongkut's University of Technology Thonburi
126 Pracha Utid Rd., Bangmod, Thung Khru
Bangkok 10140, Thailand
E-mail*: ampol.kar@kmutt.ac.th
(Corresponding Author)

and

Avinash Unnikrishnan, Ph.D.
Assistant Professor
Department of Civil and Environmental Engineering
West Virginia University
Room 627 ESB ; P.O. Box 6103
Morgantown, WV 26506-6103
Email : Avinash.Unnikrishnan@mail.wvu.edu

Submitted for Presentation at the 92nd Annual Meeting of the Transportation Research Board and Publication in the Transportation Research Record Series

15 November 2012

(2 Tables + 3 Figures) × 250 = 1,250 words
Text = 6,171 words
Total = 7,421 words

ABSTRACT

This paper studies the inventory management and routing problem in a two-level supply chain where a single plant serves a set of warehouses, which in turn serve a set of customers with stochastic demands. A set partitioning based probabilistic chance constrained nonlinear integer program is provided for the combined continuous inventory control and multi-depot vehicle routing problem while accounting for probability of inventory capacity violation, order quantity capacity, service levels, vehicle capacity restrictions and route duration limits. Two tabu search heuristics, differing in the way initial solutions are generated, are applied to solve the problem. Computational tests on standard tests networks reveal that integrating the inventory management and routing decisions by solving the combined inventory management and routing problem may yield cost savings of up to 14% over the sequential approach where both problems are solved separately. The best objective function value obtained by the tabu search heuristic was found to increase with increase in customer demand variance but decrease with increase in order quantity capacity and route duration limit. Variance of the customer demand was found to have significant impact on the solution quality.

INTRODUCTION

We consider a two-level supply chain, in which a single plant serves a set of warehouses, which in turn serve a set of end customers with stochastic demands. Inventory control decisions and vehicle routing decisions are made at the operational level for each warehouse. The inventory control problem (ICP) determines optimal order quantity, reorder point and safety stock, so that the total ordering and holding costs are minimal. The multi-depot vehicle routing problem (MDVRP) determines an optimal set of vehicle routes for each depot to satisfy demands such that the routing costs are minimal. Typically, these two problems are solved sequentially. Indeed, ICP and MDVRP are interrelated. The inventory control decisions for a warehouse depend on the demands incurred at this warehouse, which are determined from the demands of customers assigned to this warehouse. The MDVRP decisions aim at minimizing routing costs without considering the impact of the customer assignment on the ordering and holding costs at warehouses. Therefore there is significant potential to optimize the supply chain costs by solving ICP and MDVRP simultaneously (a.k.a. inventory routing problem: IRP).

(1) provides a detailed review of the IRP variants and their solution methods. Numerous studies focus on IRP application in a Vendor Managed Inventory (VMI) setting where a single vendor delivers goods to multiple customers and coordinates the routing and delivery decisions so that the customer always has sufficient inventory. Depending on the nature of the time horizon for the decision making – IRP can be classified into single day, multi-day or a long term horizon operational problem. Normally the long term horizon problem use frequency as the decision variable and the shorter duration studies are normally time based. This paper is different from the past works as in our work the customers can be served by one among multiple warehouses. Moreover we do not adopt a VMI approach. In our model, the inventories are located at warehouses and not at the customers.

Similar to *(2,3)*, we assume that each warehouse follows the continuous inventory control policy, and we explicitly consider the probabilities of unfulfilled demands, the probabilities of inventory capacity violation and the order quantity capacity. The considered policy does not penalize unfulfilled demands. Rather, a reorder point is determined such that after order submission to the plant the inventory level should cover the demand generated during the lead time with probability. Since the cost of alternative storage space especially in the urban areas is high, it is essential to control the level of service associated with the inventory capacity. The probabilities of inventory capacity violation are employed in the chance constrained stochastic programming framework. The vehicle capacity restrictions are common in the urban areas, and this can be taken into account by setting order quantity capacity and through capacity constraints in the routing problem. In MDVRP, we explicitly consider the route duration limit which arises in a number of applications such as perishable goods delivery problems and time-critical delivery problems.

The contributions of this study are three-fold. First, the model for the combined continuous inventory control and MDVRP accounting for route duration limits and stochastic inventory capacity constraints is formulated. Second, tabu search heuristics are developed. Third, the performances of the proposed tabu search algorithms are compared against each other and against the sequential approach on hypothetical test networks based on *(4)*'s test problems.

FORMULATION

The inventory routing model is developed based on the works by *(3)* and *(5)*. This combined model is a set partitioning-based formulation that has the stochastic inventory capacity constraints and the order quantity capacity constraints. Daily delivery demands of customers are assumed independent and normally distributed. Each customer is served on exactly a route by a warehouse, and a single commodity is considered. The proposed model embeds the continuous inventory control policy, which is briefly reviewed here. At any warehouse *i*, we assume a continuous inventory control policy (Q_i, RP_i) to meet normally distributed random demand $\tilde{D}_i$ with the mean of ED_i (product units per day) and the variance of VD_i (squared product unit per day). ED_i and VD_i are variables, since they depend on the customers assigned to each warehouse *i*. Q_i is the order quantity at warehouse *i*, and RP_i is the reorder point at warehouse *i*. The plant takes a lead time LT_i to fulfill an incoming order from warehouse *i*. When the inventory level falls below RP_i, an order of Q_i units is triggered, which is received after LT_i time units. When an order is submitted to the plant, the inventory level should cover the demand generated during the lead time LT_i, with probability 1-α (called service level):

$$\text{Prob}(\tilde{D}_i \cdot LT_i \le RP_i) = 1 - \alpha \qquad (1)$$

where $\tilde{D}_i \cdot LT_i$ is the normally distributed random demand generated during the lead time at warehouse *i* with the mean $ED_i \cdot LT_i$ and variance $VD_i \cdot LT_i$. Eq.(1) can be standardized; then, RP_i can be determined:

$$RP_i = ED_i \cdot LT_i + z_{1-\alpha}\sqrt{LT_i}\sqrt{VD_i} \qquad (2)$$

$z_{1-\alpha}$ is assumed fixed for the entire network, determining a homogeneous service level for the whole system. $z_{1-\alpha}\sqrt{LT_i}\sqrt{VD_i}$ is the average safety stock. Given that HC_i is the holding cost per time unit for warehouse *i* (\$/unit/day), and OC_i is the fixed ordering cost (\$/order), the expected holding and ordering cost rate (\$/day) is:

$$HC_i \cdot z_{1-\alpha}\sqrt{LT_i}\sqrt{VD_i} + \frac{1}{2}HC_i \cdot Q_i + \frac{OC_i}{Q_i}ED_i \qquad (3)$$

The first term in Eq.(3) is the average safety stock cost. The last two terms in Eq.(3) represent the costs of the known Economic Order Quantity (EOQ) model. This is the average inventory and ordering cost incurred due to the ordering process, if the order size is always Q_i. The peak inventory levels take place when the orders arrive at warehouse, and equal to $RP_i - \tilde{D}_i \cdot LT_i + Q_i$. When setting maximum probability β to violate the inventory capacity $I_i^{\max}$ at peak levels, the inventory capacity constraint can be written as chance constraints *(3)*:

$$\text{Prob}(RP_i - \tilde{D}_i \cdot LT_i + Q_i \le I_i^{\max}) \ge 1 - \beta \quad \forall i \in V_{WH} \qquad (4)$$

Eq.(4) can be rewritten as nonlinear inequalities *(2,3)*, which are the stochastic inventory capacity constraints:

$$Q_i + (z_{1-\alpha} + z_{1-\beta}) \cdot \sqrt{LT_i} \cdot \sqrt{VD_i} \le I_i^{\max} \ \ \forall i \in V_{WH} \tag{5}$$

Sets

V_{CUS} = set of customer locations

V_{WH} = set of warehouse locations

P_i = set of all feasible routes (with respect to route duration limit and vehicle capacity restriction) associated with warehouse *i*

Parameters

μ_j =mean of daily demand for customer *j*

σ_j^2 = variance of daily demand for customer *j*

n_{WH} = number of warehouse locations

n_{CUS} = number of customers to be served

RC_i = transportation unit cost between the plant and warehouse *i* ($/unit/day)

$Q_i^{\max}$ = order quantity capacity for warehouse *i*

$I_i^{\max}$ = inventory capacity for warehouse *i*

LT_i = lead time that the plant takes to fulfill an incoming order from warehouse *i*

OC_i = fixed ordering cost at warehouse *i* ($/order)

HC_i = holding cost per day per product unit at warehouse *i* ($/unit-day)

$z_{1-\alpha}, z_{1-\beta}$ = values of standard normal distribution that accumulates the probability 1-α and 1-β

a_{jik} = 1 if route *k* associated with warehouse *i* visits customer *j*; 0 otherwise

d_{ik} = cost of route *k* associated with warehouse *i*

Decision Variables

y_{ik} = 1 if route *k* associated with warehouse *i* is chosen; 0 otherwise.

Q_i = order quantity for warehouse *i*

ED_i = mean of served daily demand by warehouse *i*

VD_i = variance of served daily demand by warehouse *i*

Z = total costs

Model

$$\begin{aligned}\min Z = &\sum_{i\in V_{WH}} \sum_{k\in P_i} d_{ik} \cdot y_{ik} + \sum_{i\in V_{WH}} \sum_{j\in V_{CUS}} \sum_{k\in P_i} a_{jik} \cdot RC_i \cdot \mu_j \cdot y_{ik} + \sum_{i\in V_{WH}} \left(\frac{OC_i}{Q_i} ED_i \right) \\ &+ \sum_{i\in V_{WH}} \left(HC_i \frac{Q_i}{2} + HC_i \cdot z_{1-\alpha} \cdot \sqrt{LT_i} \cdot \sqrt{VD_i} \right)\end{aligned} \tag{6.1}$$

Subject to

$$\sum_{i \in V_{WH}} \sum_{k \in P_i} a_{jik} \cdot y_{ik} = 1 \qquad \forall j \in V_{CUS} \qquad (6.2)$$

$$Q_i + (z_{1-\alpha} + z_{1-\beta}) \cdot \sqrt{LT_i} \cdot \sqrt{VD_i} \le I_i^{\max} \qquad \forall i \in V_{WH} \qquad (6.3)$$

$$\sum_{j \in V_{CUS}} \sum_{k \in P_i} \mu_j \cdot a_{jik} \cdot y_{ik} = ED_i \qquad \forall i \in V_{WH} \qquad (6.4)$$

$$\sum_{j \in V_{CUS}} \sum_{k \in P_i} \sigma_j^2 \cdot a_{jik} \cdot y_{ik} = VD_i \qquad \forall i \in V_{WH} \qquad (6.5)$$

$$y_{ik} \in \{0,1\} \qquad \forall i \in V_{WH}, \forall k \in P_i \qquad (6.6)$$

$$0 \le Q_i \le Q_i^{\max} \qquad \forall i \in V_{WH} \qquad (6.7)$$

(6.1) calculates the total costs Z composed four terms - total MDVRP costs, total direct transportation costs between the plant and warehouses, total expected ordering costs and total expected holding costs, respectively. Eqs.(6.2) enforce that each customer is served on exactly a route by a warehouse. Eqs.(6.3) are non-linear constraints assuring that the inventory capacity for each warehouse is satisfied at least with probability 1-β and that the reorder point can cover the stochastic demand during the lead time with probability 1-α. Eqs.(6.4)-(6.5) determine the mean and variance of the served demands assigned to each warehouse (assume that demands are independent and normally distributed across the customers). Eqs.(6.7) constrain the order quantity to be within the order quantity capacity, which is assumed homogeneous for each warehouse, and can be set as the vehicle (from plant to warehouse) capacity.

The VRP is NP-hard, which is a special case of the IRP. Thus, IRP is also NP-hard. The proposed formulation potentially contains an exponential number of variables (y_{ik}), and there exists nonlinearity in Eqs.(6.1) and Eqs.(6.3), yielding a non-convex non-linear mixed-integer program. In effect, there is not an efficient solution method that guarantees an optimal solution, and this essentially requires a metaheuristic approach. In this paper, we propose tabu search heuristics.

TABU SEARCH HEURISTICS

In this paper, we modify the tabu search heuristic for MDVRP by *(6)* in order to incorporate the continuous inventory control policy for warehouses in the two-level supply chain, accounting for route duration limits and stochastic inventory capacity constraints. Let $G = (V, A)$ be a directed graph. $V = \{V_{WH}, V_{CUS}\}$ is a vertex set where $V_{WH} = \{v_{01}, v_{02}, ..., v_{0n_{WH}}\}$ is the set of warehouse (or depot) locations and $V_{CUS} = \{v_1, v_2, ..., v_{n_{CUS}}\}$ is the set of customers. $A = \{(v_i, v_j) : i \ne j\}$ is an arc set. Vertex $v_{0i} \in V_{WH}$ denotes a warehouse where m_i identical vehicles are based. m_i is assumed unlimited. Vertex $v_j \in V_{CUS}$ denotes a customer. With every arc (v_i, v_j) is associated a fixed nonnegative distance c_{ij}. $V^i = \{v_0^i, v_1^i, ..., v_{n_{CUS}^i}^i\}$ is the vertex set associated with warehouse *i*; v_0^i a warehouse vertex; n_{CUS}^i the number of customers assigned to warehouse *i*. Customer v_j has an independent and normally distributed demand with the mean μ_j and variance σ_j^2. Each city v_j requires a fixed service time δ_j, and each warehouse v_{0i} has no service time.

A least cost solution is determined such that:

- Total cost is minimized, including direct transport cost between the plant and warehouses, MDVRP costs from warehouses to customers, ordering costs, and inventory holding costs.
- The order quantity from warehouse v_{0i} to the plant may not exceed its maximum value $Q_{0i}^{\max}$.
- When an order is submitted to the plant by a warehouse, the reorder point can cover the stochastic demand generated during the lead time with probability 1-α.
- For each warehouse, the inventory level at peak levels may violate the inventory capacity with the maximum probability β.
- A route starts and ends at a warehouse.
- Each Customer in V_{CUS} is visited exactly once by exactly a vehicle based at a warehouse.
- The total average daily demands served by a vehicle based at warehouse v_{0i} may not exceed the vehicle capacity $RD_{0i}^{\max}$.
- The duration (travel plus service times) of any route beginning at warehouse v_{0i} and ending at the last customer visited on this route may not exceed the route duration limit $L_{0i}^{\max}$.

The tabu search algorithm consists of two phases: (1) construction of an initial solution and (2) solution improvement as shown in Figure 1. Inspired by *(7)*, we maintain the following information in our implementation in order to save computational efforts:

- For every route r_1 and warehouse i_1, the sum of the average delivery quantities currently assigned to this route is $q_{r_1}^{i_1}$; the duration (travel plus service time) of round-trip route r_1 beginning and ending at warehouse i_1 is $rl_{r_1}^{i_1}$; the duration (travel plus service time) of route r_1 beginning at warehouse i_1 and ending at the last customer visited in route r_1 is $pl_{r_1}^{i_1}$.
- For every warehouse i_1, the sum of average currently served demands is ED_{i_1}; the sum of currently served demand variances is VD_{i_1}.

With such information maintained, it is easy to verify the route feasibility of inserting a customer into route r_1 associated with warehouse i_1; i.e., check whether $pl_{r_1}^{i_1} \le L_{0i}^{\max}$ and $q_{r_1}^{i_1} \le RD_{0i}^{\max}$.

Heuristic Approximation for (Q_i, RP_i)

When the means and variances of currently served demands (ED_i and VD_i) for warehouses are known, the continuous inventory control policies (Q_i, RP_i) with stochastic inventory capacity constraints and order quantity capacity constraints can be heuristically approximated. In our formulation, there are constraints on Q_i (see constraints 6.3 and 6.7). Two decision variables for the continuous ICP are order quantities (Q_i) and reorder points (RP_i). RP_i can be determined by

Eq.(2) when ED_i and VD_i are known. The heuristic approximation of an optimal order quantity for warehouse i (Q_i^*) is described below.

If constraints (6.3) and (6.7) are removed, Q_i^* can be approximated through the first order optimality condition. When the constraints on Q_i are taken into account, the first order optimality conditions for a constrained minimum is employed to approximate Q_i^*. Constraints (6.3) and (6.7) can be written in the standard form as:

$$-Q_i \geq (z_{1-\alpha} + z_{1-\beta})\sqrt{LT_i}\sqrt{VD_i} - I_i^{\max} \qquad \forall i \in V_{WH} : u_{1i} \qquad (7.1)$$

$$-Q_i \geq -Q_i^{\max} \qquad \forall i \in V_{WH} : u_{2i} \qquad (7.2)$$

$$Q_i \geq 0 \qquad \forall i \in V_{WH} : u_{3i} \qquad (7.3)$$

where u_{1i}, u_{2i} and u_{3i} are dual variables associated with Eq.(7.1)-(7.3).

The Karush-Kuhn-Tucker (KKT) conditions for the minimum program (6.1), (7.1)-(7.3) where only Q_i are decision variables, are:

$$\frac{\partial Z(Q^*)}{\partial Q_i} = -u_{1i} - u_{2i} + u_{3i} \qquad \forall i \in V_{WH} \qquad (8.1)$$

$$u_{1i} \geq 0; u_{2i} \geq 0; u_{3i} \geq 0 \qquad \forall i \in V_{WH} \qquad (8.2)$$

$$u_{1i} \cdot \left(Q_i^* + (z_{1-\alpha} + z_{1-\beta})\sqrt{LT_i}\sqrt{VD_i} - I_i^{\max}\right) = 0 \qquad \forall i \in V_{WH} \qquad (8.3)$$

$$u_{2i} \cdot (Q_i^* - Q_i^{\max}) = 0 \qquad \forall i \in V_{WH} \qquad (8.4)$$

$$u_{3i} \cdot Q_i^* = 0 \qquad \forall i \in V_{WH} \qquad (8.5)$$

$$-Q_i^* \geq (z_{1-\alpha} + z_{1-\beta})\sqrt{LT_i}\sqrt{VD_i} - I_i^{\max} \qquad \forall i \in V_{WH} : u_{1i} \qquad (8.6)$$

$$-Q_i^* \geq -Q_i^{\max} \qquad \forall i \in V_{WH} : u_{2i} \qquad (8.7)$$

$$Q_i^* \geq 0 \qquad \forall i \in V_{WH} : u_{3i} \qquad (8.8)$$

For any warehouse with served demands, the optimal order quantity is naturally greater than zero. Then, Eq.(8.5) implies that u_{3i} equal to 0. Then, Eq.(8.1) become:

$$\frac{\partial Z(Q^*)}{\partial Q_i} = -u_{1i} - u_{2i} \qquad (8.1a)$$

This implies that the stationary point with the property $\frac{\partial Z}{\partial Q_i} = 0$ can be either within the feasible range of Q_i or greater than the feasible range of Q_i. The stationary point cannot be less than the feasible range of Q_i; otherwise, $\frac{\partial Z(Q^*)}{\partial Q_i}$ becomes positive, given that $Z(Q)$ is assumed convex with respect to Q_i. When the stationary point is within the feasible range of Q_i, the minimal point is the stationary point. Eq.(8.3) and (8.4) imply that $u_{1i} = 0$ and $u_{2i} = 0$, and Eq.(8.1) yields $\frac{\partial Z(Q^*)}{\partial Q_i} = 0$. When the stationary point is greater than the feasible range of Q_i, the minimal point is not the stationary point. Eq.(8.1a) and the assumed convexity of $Z(Q)$ imply that the minimal point is at the boundary of either Eq.(8.6) or Eq.(8.7). Thus, Q_i^* can be determined from the equation:

$$Q_i^* = \min\left\{\sqrt{\frac{2OC_i \cdot ED_i}{HC_i}}, \min\left\{Q_i^{\max}, I_i^{\max} - (Z_{1-\alpha} + Z_{1-\beta})\sqrt{LT_i}\sqrt{VD_i}\right\}\right\} \quad (9)$$

Phase I: Construction of an initial solution

Step I.1. Each customer is assigned to its nearest warehouse. Then, for each warehouse, sort assigned customers in increasing order of the angle that they make with the warehouse and a horizontal line.

Step I.2. Create initial vehicle routes for each warehouse. This will be described in the next subsections.

Step I.3. Determine RP_i and Q_i, using Eq.(2) and (9), respectively.

Step I.4. Determine the objective function value of the initial solution, using Eq.(6.1)

We consider two alternatives to create initial routes in Step I.2: initial solution types 1 and 2 (based on *(8)* and *(9)*, respectively).

Construction of Initial Solution Type 1

For each warehouse i=1,…, n_{WH}, do

(a) Let v_j^i be a customer randomly chosen among those closest to warehouse i (vertex v_0^i)

(b) Set m_i =1

(c) Using the customer vertex sequence $(v_j^i, v_{j+1}^i, ..., v_{n_{CUS}^i}^i, v_1^i, ..., v_{j-1}^i)$, perform the following steps for every customer assigned to warehouse i to obtain an initial routing solution, $S_{MDVRP} = \{S_{SDVRP}^i \forall i \in V_{WH}\}$:

- Insert each customer into the route m_i based at warehouse i (vertex v_0^i) using the generalized insertion (GENI) algorithm by *(10)*.
- If the insertion of customer in the route m_i would result in the violation of vehicle capacity or route duration limit, set $m_i = m_i + 1$.

Construction of Initial Solution Type 2

For each warehouse i=1,…, n_{WH}, do

(a) Let v_j^i be a customer randomly chosen among those closest to the depot

(b) Using the customer vertex sequence $(v_j^i, v_{j+1}^i, ..., v_{n_{CUS}^i}^i, v_1^i, ..., v_{j-1}^i)$, construct a tour on all vertices assigned to warehouse i by means of GENI procedure and Unstringing and Stringing (US) procedure *(10)*.

(c) Start with warehouse i (vertex v_0^i), create m_i vehicle routes by following the tour. The first vehicle contains all customers starting from the first customer on the tour and up to, but excluding, the first customer v whose inclusion in the route would cause a violation of the capacity or route duration limit. This process is repeated, starting from the city v, and until all customers have been included into routes. The initial MDVRP solution is $S_{MDVRP} = \{S_{SDVRP}^i \forall i \in V_{WH}\}$.

Phase II: Solution Improvement

The initial solution is an input in Phase II consisting of 3 sub-phases (see Figure 1). Three basic procedures that are employed in these sub-phases are first described including one-route, two-route and three-route procedures, followed by the descriptions of three sub-phases. Then, the selection of routes for two-route and three route procedures in the three sub-phases is described.

One-Route Procedure

The one-route procedure is a post-optimizer on single-vehicle routes. In this study, the US algorithm by *(10)* is employed while maintaining route duration feasibility and vehicle capacity feasibility. Since the procedure improves the sequence of customers on a particular route without reassigning any customer to different warehouses, ED_i and VD_i are unaffected. Thus, the optimal order quantity and reorder point as well as ordering and holding costs are not changed.

Two-Route Procedure

The two-route procedure moves vertices belonging to two different routes assigned to one or two warehouses. Let $(v_{h_1}, v_{j_1}, v_{k_1}, v_{l_1})$ and $(v_{h_2}, v_{j_2}, v_{k_2}, v_{l_2})$ be two sequences of four consecutive vertices (possibly including a warehouse) from route r_1 based at warehouse i_1 and route r_2 based at warehouse i_2, respectively. Similar to *(6,11,12)*, the following 6 moves are attempted as long as a warehouse is not moved, and vehicle capacity feasibility and route duration feasibility are maintained. The six moves are described together with the calculation of changes in relevant q_r^i, rl_r^i, pl_r^i, ED_i, and VD_i.

(a)Insert v_{j_1} between v_{h_2} and v_{j_2}

The two vertex sequences become $(v_{h_1}, v_{k_1}, v_{l_1})$ and $(v_{h_2}, v_{j_1}, v_{j_2}, v_{k_2}, v_{l_2})$, respectively. The changes in the round-trip lengths are $\Delta_{rl_1} = -c_{h_1,j_1} - c_{j_1,k_1} + c_{h_1,k_1} - \delta_{j_1}$ and $\Delta_{rl_2} = -c_{h_2,j_2} + c_{h_2,j_1} + c_{j_1,j_2} + \delta_{j_1}$. If v_{j_1} is the last customer visited on route r_1, $\Delta_{pl_1} = -c_{h_1,j_1} - \delta_{j_1}$; otherwise, $\Delta_{pl_1} = \Delta_{rl_1}$. If v_{h_2} is the last customer visited on route r_2, $\Delta_{pl_2} = c_{h_2,j_1} + \delta_{j_1}$; otherwise, $\Delta_{pl_2} = \Delta_{rl_2}$. The changes in the average delivery demands are $\Delta q_{r_1}^{i_1} = -\mu_{j_1}$ and $\Delta q_{r_2}^{i_2} = \mu_{j_1}$. If $i_1 \neq i_2$, $\Delta ED_{i_1} = -\mu_{j_1}$, $\Delta ED_{i_2} = \mu_{j_1}$, $\Delta VD_{i_1} = -\sigma_{j_1}^2$ and $\Delta VD_{i_2} = \sigma_{j_1}^2$. Otherwise, $\Delta ED_{i_1} = \Delta ED_{i_2} = 0$ and $\Delta VD_{i_1} = \Delta VD_{i_2} = 0$.

(b)Insert v_{j_2} between v_{h_1} and v_{j_1}

The two vertex sequences become $(v_{h_1}, v_{j_2}, v_{j_1}, v_{k_1}, v_{l_1})$ and $(v_{h_2}, v_{k_2}, v_{l_2})$, respectively. The changes in the round-trip lengths are $\Delta_{rl_1} = -c_{h_1,j_1} + c_{h_1,j_2} + c_{j_2,j_1} + \delta_{j_2}$ and $\Delta_{rl_2} = -c_{h_2,j_2} - c_{j_2,k_2} + c_{h_2,k_2} - \delta_{j_2}$. If v_{h_1} is the last customer visited on route *r1*, $\Delta_{pl_1} = c_{h_1,j_2} + \delta_{j_2}$; otherwise, $\Delta_{pl_1} = \Delta_{rl_1}$. If v_{j_2} is the last customer visited on route *r2*, $\Delta_{pl_2} = -c_{h_2,j_2} - \delta_{j_2}$; otherwise, $\Delta_{pl_2} = \Delta_{rl_2}$. The changes in the average delivery demands are $\Delta q_{r_1}^{i_1} = \mu_{j_2}$ and $\Delta q_{r_2}^{i_2} = -\mu_{j_2}$. If $i_1 \neq i_2$, then $\Delta ED_{i_1} = \mu_{j_2}$, $\Delta ED_{i_2} = -\mu_{j_2}$, $\Delta VD_{i_1} = \sigma_{j_2}^2$ and $\Delta VD_{i_2} = -\sigma_{j_2}^2$. Otherwise, $\Delta ED_{i_1} = \Delta ED_{i_2} = 0$ and $\Delta VD_{i_1} = \Delta VD_{i_2} = 0$.

(c) Swap v_{j_1} and v_{j_2}

The two vertex sequences become $(v_{h_1}, v_{j_2}, v_{k_1}, v_{l_1})$ and $(v_{h_2}, v_{j_1}, v_{k_2}, v_{l_2})$. The changes in the round-trip lengths are $\Delta_{rl_1} = -c_{h_1,j_1} - c_{j_1,k_1} + c_{h_1,j_2} + c_{j_2,k_1} - \delta_{j_1} + \delta_{j_2}$ and $\Delta_{rl_2} = -c_{h_2,j_2} - c_{j_2,k_2} + c_{h_2,j_1} + c_{j_1,k_2} - \delta_{j_2} + \delta_{j_1}$. If v_{j_1} is the last customer visited on route *r1*, $\Delta_{pl_1} = -c_{h_1,j_1} + c_{h_1,j_2} - \delta_{j_1} + \delta_{j_2}$; otherwise, $\Delta_{pl_1} = \Delta_{rl_1}$. If v_{j_2} is the last customer visited on route *r2*, $\Delta_{pl_2} = -c_{h_2,j_2} + c_{h_2,j_1} - \delta_{j_2} + \delta_{j_1}$; otherwise, $\Delta_{pl_2} = \Delta_{rl_2}$. The changes in the average delivery demands are $\Delta q_{r_1}^{i_1} = -\mu_{j_1} + \mu_{j_2}$ and $\Delta q_{r_2}^{i_2} = \mu_{j_1} - \mu_{j_2}$. If $i_1 \neq i_2$, then $\Delta ED_{i_1} = -\mu_{j_1} + \mu_{j_2}$, $\Delta ED_{i_2} = \mu_{j_1} - \mu_{j_2}$, $\Delta VD_{i_1} = -\sigma_{j_1}^2 + \sigma_{j_2}^2$ and $\Delta VD_{i_2} = \sigma_{j_1}^2 - \sigma_{j_2}^2$. Otherwise, $\Delta ED_{i_1} = \Delta ED_{i_2} = 0$ and $\Delta VD_{i_1} = \Delta VD_{i_2} = 0$.

(d)Insert (v_{j_1}, v_{k_1}) between (v_{h_2}, v_{j_2})

The two vertex sequences become (v_{h_1}, v_{l_1}) and $(v_{h_2}, v_{j_1}, v_{k_1}, v_{j_2}, v_{k_2}, v_{l_2})$. The changes in the round-trip lengths are $\Delta_{rl_1} = -c_{h_1,j_1} - c_{j_1,k_1} - c_{k_1,l_1} + c_{h_1,l_1} - \delta_{j_1} - \delta_{k_1}$ and $\Delta_{rl_2} = -c_{h_2,j_2} + c_{h_2,j_1} + c_{j_1,k_1} + c_{k_1,j_2} + \delta_{j_1} + \delta_{k_1}$. If v_{k_1} is the last customer visited on route *r1*, $\Delta_{pl_1} = -c_{h_1,j_1} - c_{j_1,k_1} - \delta_{j_1} - \delta_{k_1}$; otherwise, $\Delta_{pl_1} = \Delta_{rl_1}$. If v_{h_2} is the last customer visited on route *r2*, $\Delta_{pl_2} = c_{h_2,j_1} + c_{j_1,k_1} + \delta_{j_1} + \delta_{k_1}$; otherwise, $\Delta_{pl_2} = \Delta_{rl_2}$. The changes in the average delivery demands are $\Delta q_{r_1}^{i_1} = -\mu_{j_1} - \mu_{k_1}$ and $\Delta q_{r_2}^{i_2} = \mu_{j_1} + \mu_{k_1}$. If $i_1 \neq i_2$, $\Delta ED_{i_1} = -\mu_{j_1} - \mu_{k_1}$, $\Delta ED_{i_2} = \mu_{j_1} + \mu_{k_1}$, $\Delta VD_{i_1} = -\sigma_{j_1}^2 - \sigma_{k_1}^2$ and $\Delta VD_{i_2} = \sigma_{j_1}^2 + \sigma_{k_1}^2$. Otherwise, $\Delta ED_{i_1} = \Delta ED_{i_2} = 0$ and $\Delta VD_{i_1} = \Delta VD_{i_2} = 0$.

(e)Insert (v_{j_2}, v_{k_2}) between (v_{h_1}, v_{j_1})

The two vertex sequences become $(v_{h_1}, v_{j_2}, v_{k_2}, v_{j_1}, v_{k_1}, v_{l_1})$ and (v_{h_2}, v_{l_2}). The changes in the round-trip lengths are $\Delta_{rl_1} = -c_{h_1,j_1} + c_{h_1,j_2} + c_{j_2,k_2} + c_{k_2,j_1} + \delta_{j_2} + \delta_{k_2}$ and $\Delta_{rl_2} = -c_{h_2,j_2} - c_{j_2,k_2} - c_{k_2,l_2} + c_{h_2,l_2} - \delta_{j_2} - \delta_{k_2}$. If v_{h_1} is the last customer visited on route *r1*, $\Delta_{pl_1} = c_{h_1,j_2} + c_{j_2,k_2} + \delta_{j_2} + \delta_{k_2}$; otherwise, $\Delta_{pl_1} = \Delta_{rl_1}$. If v_{k_2} is the last customer visited on route *r2*,

$\Delta_{pl_2} = -c_{h_2,j_2} - c_{j_2,k_2} - \delta_{j_2} - \delta_{k_2}$; otherwise, $\Delta_{pl_2} = \Delta_{rl_2}$. The changes in the average delivery demands are $\Delta q_{r_1}^{i_1} = \mu_{j_2} + \mu_{k_2}$ and $\Delta q_{r_2}^{i_2} = -\mu_{j_2} - \mu_{k_2}$. If $i_1 \neq i_2$, $\Delta ED_{i_1} = \mu_{j_2} + \mu_{k_2}$, $\Delta ED_{i_2} = -\mu_{j_2} - \mu_{k_2}$, $\Delta VD_{i_1} = \sigma_{j_2}^2 + \sigma_{k_2}^2$ and $\Delta VD_{i_2} = -\sigma_{j_2}^2 - \sigma_{k_2}^2$. Otherwise, $\Delta ED_{i_1} = \Delta ED_{i_2} = 0$ and $\Delta VD_{i_1} = \Delta VD_{i_2} = 0$.

(f)Swap (v_{j_1}, v_{k_1}) and (v_{j_2}, v_{k_2})

The two vertex sequences become $(v_{h_1}, v_{j_2}, v_{k_2}, v_{l_1})$ and $(v_{h_2}, v_{j_1}, v_{k_1}, v_{l_2})$. The changes in the round-trip lengths are $\Delta_{rl_1} = -c_{h_1,j_1} - c_{j_1,k_1} - c_{k_1,l_1} + c_{h_1,j_2} + c_{j_2,k_2} + c_{k_2,l_1} - \delta_{j_1} - \delta_{k_1} + \delta_{j_2} + \delta_{k_2}$ and $\Delta_{rl_2} = -c_{h_2,j_2} - c_{j_2,k_2} - c_{k_2,l_2} + c_{h_2,j_1} + c_{j_1,k_1} + c_{k_1,l_2} - \delta_{j_2} - \delta_{k_2} + \delta_{j_1} + \delta_{k_1}$. If v_{k_1} is the last customer visited on route *r1*, $\Delta_{pl_1} = -c_{h_1,j_1} - c_{j_1,k_1} + c_{h_1,j_2} + c_{j_2,k_2} - \delta_{j_1} - \delta_{k_1} + \delta_{j_2} + \delta_{k_2}$; otherwise, $\Delta_{pl_1} = \Delta_{rl_1}$. If v_{k_2} is the last customer visited on route *r2*, $\Delta_{pl_2} = c_{h_2,j_1} + c_{j_1,k_1} - c_{h_2,j_2} - c_{j_2,k_2} + \delta_{j_1} + \delta_{k_1} - \delta_{j_2} - \delta_{k_2}$; otherwise, $\Delta_{pl_2} = \Delta_{rl_2}$. The changes in the average delivery demands are $\Delta q_{r_1}^{i_1} = -\mu_{j_1} - \mu_{k_1} + \mu_{j_2} + \mu_{k_2}$ and $\Delta q_{r_2}^{i_2} = \mu_{j_1} + \mu_{k_1} - \mu_{j_2} - \mu_{k_2}$. If $i_1 \neq i_2$, $\Delta ED_{i_1} = -\mu_{j_1} - \mu_{k_1} + \mu_{j_2} + \mu_{k_2}$, $\Delta ED_{i_2} = \mu_{j_1} + \mu_{k_1} - \mu_{j_2} - \mu_{k_2}$, $\Delta VD_{i_1} = -\sigma_{j_1}^2 - \sigma_{k_1}^2 + \sigma_{j_2}^2 + \sigma_{k_2}^2$ and $\Delta VD_{i_2} = \sigma_{j_1}^2 + \sigma_{k_1}^2 - \sigma_{j_2}^2 - \sigma_{k_2}^2$. Otherwise, $\Delta ED_{i_1} = \Delta ED_{i_2} = 0$ and $\Delta VD_{i_1} = \Delta VD_{i_2} = 0$.

Three-Route Procedure

The three-route procedure is an exchange scheme involving three routes *(6)*. Let $(v_{h_1-1}, v_{h_1}, v_{h_1+1})$, $(v_{h_2-1}, v_{h_2}, v_{h_2+1}, ..., v_{j_2}, v_{k_2})$ and (v_{j_3}, v_{k_3}) be three sequences of consecutive vertices (possibly including a warehouse) from routes *r1*, *r2* and *r3* with at least 3, 4 and 3 vertices respectively, based at warehouses *i1*, *i2* and *i3*. For routes *r2* and *r3*, consider the sequences of two vertices (v_{j_2}, v_{k_2}) and (v_{j_3}, v_{k_3}) where $v_{j_2} \neq v_{h_2}$ and $v_{k_2} \neq v_{h_2}$. Then the following combination of moves is attempted as long as vehicle capacity feasibility and route duration feasibility are maintained, and a warehouse is not moved: insert v_{h_1} between v_{j_2} and v_{k_2}, and insert v_{h_2} between v_{j_3} and v_{k_3}. The move is described together with the calculation of changes in relevant q_r^i, rl_r^i, pl_r^i, ED_i, and VD_i. After three-route exchange, the three vertex sequences become (v_{h_1-1}, v_{h_1+1}), $(v_{h_2-1}, v_{h_2+1}, ..., v_{j_2}, v_{h_1}, v_{k_2})$ and $(v_{j_3}, v_{h_2}, v_{k_3})$. The changes in the round-trip lengths are $\Delta_{rl_1} = -c_{h_1-1,h_1} - c_{h_1,h_1+1} + c_{h_1-1,h_1+1} - \delta_{h_1}$, $\Delta_{rl_2} = -c_{h_2-1,h_2} - c_{h_2,h_2+1} + c_{h_2-1,h_2+1} - \delta_{h_2} - c_{j_2,k_2} + c_{j_2,h_1} + c_{h_1,k_2} + \delta_{h_1}$ and $\Delta_{rl_3} = -c_{j_3,k_3} + c_{j_3,h_2} + c_{h_2,k_3} + \delta_{h_2}$. If v_{h_1} is the last customer visited on route *r1*, $\Delta_{pl_1} = -c_{h_1-1,h_1} - \delta_{h_1}$; otherwise, $\Delta_{pl_1} = \Delta_{rl_1}$. If v_{h_2} is the last customer visited on route *r2*, $\Delta_{pl_2} = -c_{h_2-1,h_2} - \delta_{h_2} - c_{j_2,k_2} + c_{j_2,h_1} + c_{h_1,k_2} + \delta_{h_1}$. If v_{j_2} is the last customer visited on route *r2*, $\Delta_{pl_2} = -c_{h_2-1,h_2} - c_{h_2,h_2+1} + c_{h_2-1,h_2+1} - \delta_{h_2} + c_{j_2,h_1} + \delta_{h_1}$. Otherwise, $\Delta_{pl_2} = \Delta_{rl_2}$. If v_{j_3} is the last customer visited on route *r3*, $\Delta_{pl_3} = c_{j_3,h_2} + \delta_{h_2}$. Otherwise, $\Delta_{pl_3} = \Delta_{rl_3}$. The changes in the average delivery demands are $\Delta q_{r_1}^{i_1} = -\mu_{h_1}$, $\Delta q_{r_2}^{i_2} = \mu_{h_1} - \mu_{h_2}$ and $\Delta q_{r_3}^{i_3} = \mu_{h_2}$. If $i_1 \neq i_2 \neq i_3$, then

$\Delta ED_{i_1} = -\mu_{h_1}$, $\Delta ED_{i_2} = \mu_{h_1} - \mu_{h_2}$, $\Delta ED_{i_3} = \mu_{h_2}$, $\Delta VD_{i_1} = -\sigma^2_{h_1}$, $\Delta VD_{i_2} = \sigma^2_{h_1} - \sigma^2_{h_2}$ and $\Delta VD_{i_3} = \sigma^2_{h_2}$. If $i_1 = i_2 = i_3$, then $\Delta ED_{i_1} = \Delta ED_{i_2} = 0$ and $\Delta VD_{i_1} = \Delta VD_{i_2} = 0$. If $i_1 = i_2 \neq i_3$, $\Delta ED_{i_1} = \Delta ED_{i_2} = -\mu_{h_2}$, $\Delta ED_{i_3} = \mu_{h_2}$, $\Delta VD_{i_1} = \Delta VD_{i_2} = -\sigma^2_{h_2}$ and $\Delta VD_{i_3} = \sigma^2_{h_2}$. If $i_1 = i_3 \neq i_2$, $\Delta ED_{i_1} = \Delta ED_{i_3} = -\mu_{h_1} + \mu_{h_2}$, $\Delta ED_{i_2} = \mu_{h_1} - \mu_{h_2}$, $\Delta VD_{i_1} = \Delta VD_{i_3} = -\sigma^2_{h_1} + \sigma^2_{h_2}$ and $\Delta VD_{i_2} = \sigma^2_{h_1} - \sigma^2_{h_2}$. If $i_1 \neq i_2 = i_3$, $\Delta ED_{i_1} = -\mu_{h_1}$, $\Delta ED_{i_2} = \Delta ED_{i_3} = \mu_{h_1}$, $\Delta VD_{i_1} = -\sigma^2_{h_1}$ and $\Delta VD_{i_2} = \Delta VD_{i_3} = \sigma^2_{h_1}$.

Sub-Phase II.1: Fast Improvement

The algorithm attempts to improve upon the incumbent by repeatedly applying the following three steps:

- Inter-warehouse: Apply two-route procedure between routes of two different warehouses.
- Intra-warehouse: Apply two-route procedure between routes of the same warehouse.
- Three-Route: Exchange vertices between three routes, using three-route procedure.

These steps are repeated until the incumbent does not improve for $n^{fast}_{\max}$ consecutive iterations. For each of the three steps, any move that yields an improvement is immediately implemented. Otherwise, the best non-tabu deteriorating move is implemented. Whenever a move is implemented, the one-route procedure is applied to all routes involved in the move.

Sub-Phase II.2: Intensification

This phase intensifies the search for better route, starting with the best known solution and working on one warehouse at the time. It applies the intra-warehouse step to each warehouse in turn until no improvement to the incumbent has been produced for $n^{\text{intens}}_{\max}$ consecutive iterations. Whenever a move is implemented, the one-route procedure is applied to all routes involved in the move.

Sub-Phase II.3: Diversification

The effect of the diversification phase is to perform a broader exploration of the solution space. The following two steps are repeated 20 times.

- First, we seek the best reinsertion of a vertex from its current route into a route belonging to a different warehouse; that is, apply the first move type of the two-route procedure limiting to only two routes associated with different warehouses. Choosing the same vertex for reinsertion is prohibited for the next 10 applications of this step. Whenever a move is implemented, the one-route procedure is applied to all routes involved in the move.
- Second, the inter-warehouse and intra-warehouse steps of the fast improvement sub-phase are applied for $n^{FastDiver}_{\max}$ consecutive iterations without improvement to the solution values obtained in the first step. Here the length of the interval during which a move is

tabu is randomly chosen in [15,20] and no aspiration criterion is used. Whenever a move is implemented, the one-route procedure is applied to all routes involved in the move.

Selection of Routes for Two-Route and Three-Route Procedures in the Three Sub-Phases

The selection of routes to which two-route and three-route procedures are applied is described *(6)*. To define the distance between a route and a warehouse or between two routes, each route is represented by its center of gravity. In inter-warehouse, we consider exchanges between each warehouse i and the $\left\lfloor \frac{n_{WH}}{2} \right\rfloor + 1$ warehouses closest to it. For each pair of warehouses i_1 and i_2, we consider exchanges between the $\left\lceil \frac{m_{i_1}}{2} \right\rceil$ routes of warehouse i_1 closest to warehouse i_2 and the $\left\lceil \frac{m_{i_2}}{2} \right\rceil$ routes of warehouse i_2 closest to warehouse i_1. In intra-warehouse, we consider all pairs of routes for each warehouse. In three-route procedure, the three routes r_1, r_2, and r_3 are selected as follows. All routes with at least 3 vertices are considered for route r_1. Route r_2 is the closest neighbor of route r_1 and has at least 4 vertices. Route r_3 is the closest neighbor of route r_2 with $r_3 \neq r_1$, and route r_3 has at least 3 vertices.

Throughout Phase II, the incumbent and its value are recorded. The current solution is not necessarily the best known because the deteriorations of the objective function are allowed. Whenever a customer is moved from its current route, moving this customer back into the same route is declared tabu for θ iterations, where θ is randomly chosen in $\left[\theta_{\min}^{FIND}, \theta_{\max}^{FIND}\right] = [4,10]$. Random tabu durations help avoid cycling. A tabu status may be overridden if implementing the corresponding move yields a better incumbent.

COMPUTATIONAL EXPERIENCES

The tabu search heuristics are implemented in C++. These run on a computer with 1.73 GHz Intel Core i7 processor and 4 GB of RAM, running under Windows 7. The data are first described. Then, the computational results of two experiments are discussed.

Data

For IRP, there is not the standard set of instances for testing algorithms. We generated instances similar to the types used in VRP. The customer locations are generated from *(4)*'s VRP instances, yielding only four sets of distinct customer locations: C1, C2, R1 and RC1. In C1 and C2, the customer locations are clustered. In R1, the customer locations are randomly generated from a uniform distribution. In RC1, the customer locations are a combination of randomly generated and clustered points. In the same manner as *(5)*, we create five instances corresponding to each group of customer locations (denoted by 50a, 50b, 75a, 75b and 100): the first 50 customers, the last 50 customers, the first 75 customers, the last 75 customers, and all 100 customers. Thus, there are 20 instances of customer locations. The service times are set at 10 time units for all customers. The average demands of customers are equal to the demands used in *(4)*. The demand variances are based on the coefficients of variance randomly generated from the range [0.45, 0.55].

For the warehouse locations, we created two sets of 4 warehouse locations for each customer instance. The first and second sets of candidate warehouse locations are denoted by wh1 and wh2, respectively. We randomly generated the warehouse locations from a uniform distribution, so that two criteria are satisfied. First, each customer location could be reached by a singleton route with the associated route duration to the last customer of at most 80 time units (M=80) from at least one warehouse. Second, each warehouse location must be assigned at least 10, 15 and 20 customers for the 50, 75 and 100 customer instances, respectively, when assigning customers to their nearest warehouse. For all warehouse instances, homogeneous unit holding costs of the four warehouses are \$0.3, \$0.6, \$0.9, \$1.2 per product unit per day; the homogeneous ordering costs \$450, \$900, \$1350 and \$1800 per order. For all warehouses, the lead times are two days; inventory capacity 2000 product units; order quantity capacity 2000 product units; unit transport cost from the plant to warehouses is zero. The distance matrix is determined based on Euclidean distance between all vertex pair. The traveling speed is assumed 1 distance unit per time unit, and routing cost is assumed \$1 per travel time unit to cover variable vehicle costs. Personnel costs and other vehicle related fixed costs are assumed to be considered outside the inventory-routing decision. The route duration limits are 100 time units. The number of available vehicles for each warehouse is unlimited with the homogeneous capacity of 100 product units, which are less constrained than the route duration limit constraints in all test problems. We identify each instance by an ID. The first part of the ID specifies the problem group (R1, C1, C2 or RC1). The second part specifies the customer subset (50a, 50b, 75a, 75b or 100). The third part specifies the set of warehouse locations (wh1 or wh2). Thus, there are 40 problem instances.

Computational Results

We calibrate the two tabu search algorithms by varying $n_{\max}^{fast}$, $n_{\max}^{\text{intens}}$ and $n_{\max}^{FastDiver}$ on a test problem, and found that the algorithm parameters suggested by *(6)* perform best ($n_{\max}^{fast}$=75, $n_{\max}^{\text{intens}}$=300 and $n_{\max}^{FastDiver}$=50). We conduct two experiments. The first experiment compares the performances of the type-1 and type-2 tabu search heuristics in terms of computational time and solution quality against the sequential approach. The sequential approach first solves MDVRP with route duration limits, whose routing solutions are input to the continuous ICP with stochastic inventory capacity constraints and order quantity capacity constraints. In the second experiment, the sensitivity analysis is performed on problem instance RC1-100-wh1 by varying the route duration limit (M=80 and 100), order quantity capacity ($Q_{0i}^{\max}$=800, 1000 and 2000) and demand variance (-30%, 0% and +30% changes). The demand variances are the product of the original demand variance and demand variance factor (DVarF); thus, -30%, 0% and +30% changes in demand variances correspond to DVarF values of 0.7, 1.0 and 1.3, respectively.

Table 1 shows the best objective values found and total computational time by type-1 and type-2 tabu search algorithms and the sequential approach on the 40 problem instances. As expected, the proposed tabu search algorithms outperform the sequential approach on all test problem instances. The computational times of all runs are less than two minutes. On 50-customer and 75-customer problem instances, the two tabu search algorithms perform approximately equally well, since the type-1 tabu search outperform the type-2 tabu search on about 50 percent of problem instances. On 100-customer instances, the type-2 tabu search outperforms the type-1 tabu search on almost all instances except an instance R1-100-wh2.

Next, the sensitivity analysis is performed to see how the solution changes with route duration limit, order quantity capacity and demand variance on problem instance RC1-100-wh1. We employ the type-2 tabu search in all runs as it performs best on this problem instance. Figure 2a shows the best objective value found when varying route duration limits, order quantity capacity and demand variances. It can be seen that the best objective value increases with the increase of demand variance, but decreases with the increase of order quantity capacity and route duration limit. The best objective value is composed of three cost components: vehicle routing, holding and ordering. Figures 2b-2d shows the three cost components when varying route duration limit, order quantity capacity and/or demand variance. As can be seen from Figure 2b, the total ordering costs increases with the decrease of the order quantity capacity, whereas the total holding costs decreases with the decrease of the order quantity capacity. Intuitively, when the order quantity is more constrained, the warehouse manager has to order more often and ordering costs are higher. Meanwhile, the peak inventory levels are lower and the total holding costs are less. Furthermore, Figure 2b shows that the routing costs increases with the decrease of route duration limit. Once the longer route duration limit is allowed, each vehicle route may serve more customers, and the routing costs is less. Figures 2c-2d show that the holding costs increase with the increase of demand variance, but it is unclear how the routing and ordering costs change with the demand variances. This is as expected as the demand variance is only directly related to the holding costs (see Eq.(6.1)). The demand variance can influence the customer assignments to different warehouses, resulting in different routing costs and ordering costs.

Figure 3a shows the continuous inventory control policies at four warehouses in the best solution when varying order quantity capacity at $L_{0i}^{\max}$=80 and DVarF=1.0. When the order quantity capacity ($Q_i^{\max}$=2000) is equal to the inventory capacity, the optimal order quantity is equal to the EOQ formula according to Eq.(9). When the order quantity capacity decreases to 1000 and 800, the customers as well as associated mean demands are reassigned between warehouses 1002 and 1003. As such the reorder points and safety stocks of warehouses 1000 and 1001 are unaffected with the change of order quantity capacity, but those of warehouses 1002 and 1003 are affected. The optimal order quantities for the case $Q_i^{\max}$= 800 and $Q_i^{\max}$=1000 are equal to $Q_i^{\max}$ according to Eq.(9).

Figure 3b shows the continuous inventory control policies at four warehouses in the best solution when varying demand variance at $L_{0i}^{\max}$=80 and $Q_i^{\max}$=800. The customers as well as associated mean demands assigned to the four warehouses are unaffected with the change of demand variance. The safety stock levels and reorder points at the four warehouses increase with the increase of demand variance, whereas the available inventory capacities at the four warehouses decrease with the increase of demand variance. This is intuitive as the safety stock is positively related to demand variance, and the reorder point includes the safety stock as shown in Eq.(2). The available inventory capacity is negatively related to demand variance (available inventory capacity = $I_i^{\max} - (Z_{1-\alpha} + Z_{1-\beta})\sqrt{LT_i}\sqrt{VD_i}$). The optimal order quantities are equal to the order quantity capacity according to Eq.(9). Table 2 shows the MDVRP policies for the four warehouses when $L_{0i}^{\max}$=80 and $L_{0i}^{\max}$=100. The number of routes is decreases with the increase of route duration limits. This is because the available vehicle capacity in each route is large enough to serve additional customers. As can be noticed in Table 2, each route has the travel time to last customer less than or equal to the route duration limit, and the mean demand of each route is less than the vehicle capacity.

CONCLUSIONS

This paper studies a two-level supply chain where a single plant supplies a single commodity to a set of warehouses which in turn serve a set of customers with stochastic demands. This paper provides a nonlinear integer programming formulation modeling the continuous inventory control policies at the warehouses and the routing of goods from the warehouses to the customers with route duration limits. The model accounts for the probability of available inventory meeting the demand during the lead time, probability of violation of inventory capacity, and restrictions on order quantity volume. Two tabu search heuristics – type 1 and type 2, differing primarily in the way initial solutions are generated are developed to solve the combined model. The optimal order quantity at each warehouse is approximated using the KKT conditions.

Computational runs are conducted on variations of the standard Solomon test instances. Type-2 tabu search was found to outperform type-1 tabu search for the 100 customer instance. For smaller customer instances, both the heuristics were found to perform equally well. Integrating the inventory management and routing decisions by solving the combined inventory management and routing problem was found to yield cost savings of up to 14% over the sequential approach where both problems are solved separately. The best objective function value obtained by the tabu search heuristic was found to increase with increase in customer demand variance, decrease with increase in order quantity capacity and route duration limit. Variance of the customer demand was found to have significant impact on the solution quality.

This paper can be extended in multiple directions. The immediate next step is to integrate warehouse facility location problem into the combined inventory management and routing model. Possible extensions include considering time-dependent travel times, stockout costs and delivery time windows.

ACKNOWLEDGMENTS

The first author gratefully acknowledges the support from The Thailand Research Fund under contract number MRG5380066, Office of the Higher Education Commission and King Mongkut’s University of Technology Thonburi Research Fund.

REFERENCES

1. Bertazzi, L., Speranza, M.G., and M.W.P. Savelsbergh. Inventory routing, in:Vehicle routing: Latest advances and new challenges, Golden, B., Raghavan, R., E. Wasil (editors.), Springer, 2008, 49-72.
2. Miranda, P.A. and R.A. Garrido. Incorporating inventory control decisions into a strategic distribution network design model with stochastic demand. *Transportation Research Part E 40*, 2004, 183-207.
3. Miranda, P.A. and R.A. Garrido. A Simultaneous Inventory Control and Facility Location Model with Stochastic Capacity Constraints. *Networks and Spatial Economics* 6, 2006, 39-53.

4. Solomon, M.M. Algorithm for the vehicle routing and scheduling problem with time window constraints. *Operations Research* 35, 1987, 254-265.
5. Berger, R.T., C.R. Coullard, and M.S. Daskin. Location-Routing Problems with Distance Constraints. *Transportation Science* 141(1), 2007, 29-43.
6. Renaud, J., G. Laporte and F.F. Boctor. A Tabu Search Heuristic for the Multi-Depot Vehicle Routing Problem. *Computers and Operations Research* 23(3), 1996, 229-235.
7. Campbell, A.M. and M. Savelsbergh. Efficient Insertion Heuristics for Vehicle Routing and Scheduling Problems. *Transportation Science 38(3)*, 2004, 369-378.
8. Cordeau, J-F., M. Gendreau, and G. Laporte . A tabu search heuristic for periodic and multi-depot vehicle routing problems, *Networks* 30(2), 1997, 105-119.
9. Gendreau, M., A. Hertz, and G. Laporte. A Tabu Search Heuristic for the Vehicle Routing Problem, *Management Science 40*, 1994, 1276-1290.
10. Gendreau, M., A. Hertz, and G. Laporte. New Insertion and Postoptimization Procedures for the Traveling Salesman Problem. *Operations Research* 40(6), 1992, 1086-1094.
11. Or, I. Traveling salesman-type combinatorial problems and their relation to logistics of regional blood banking, Ph.D. dissertation, 1976, Northwestern University.
12. Lin, S. Computer solutions of traveling salesman problem, Bell System Technical Journal 44(10), 2245-2269, 1965.

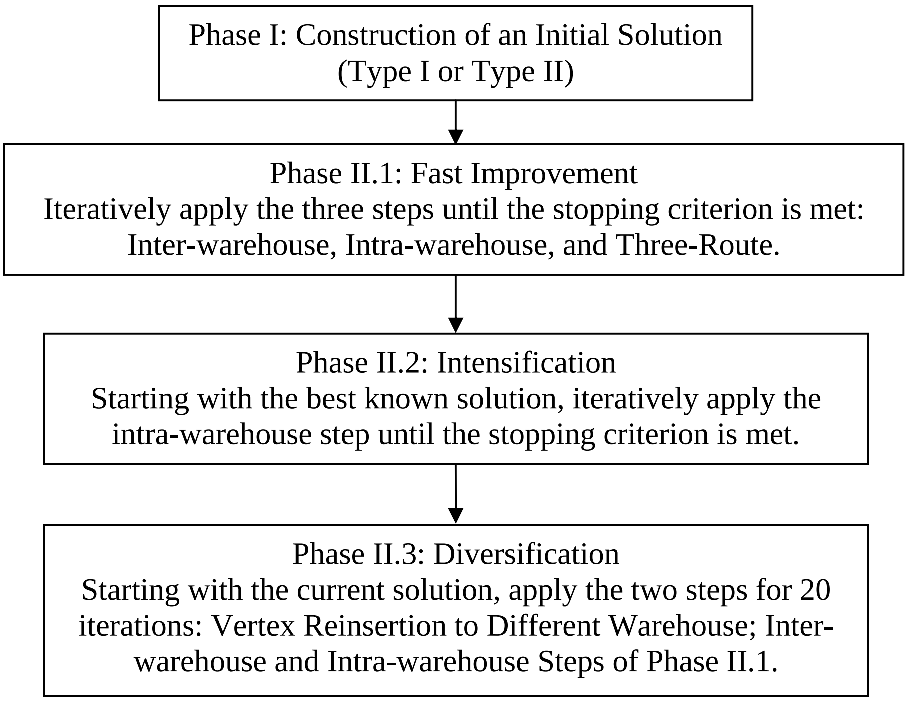
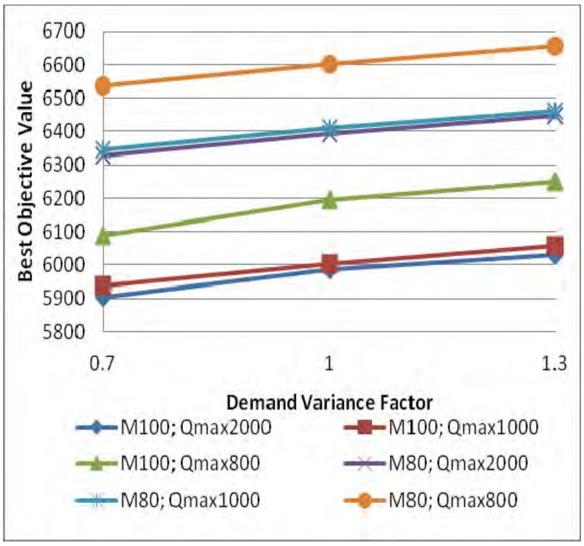
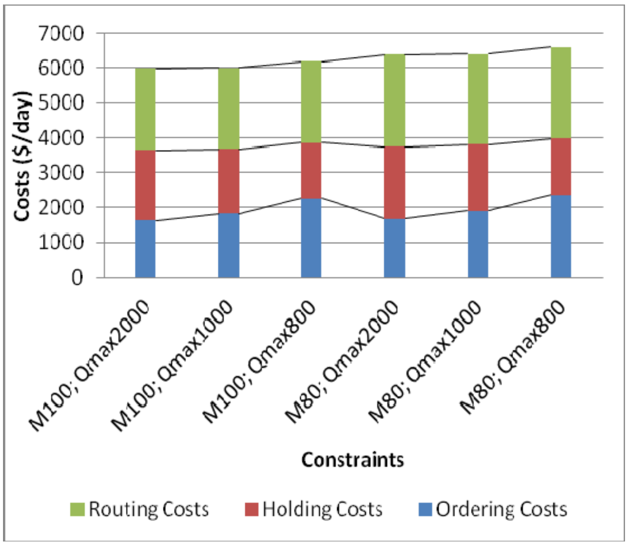


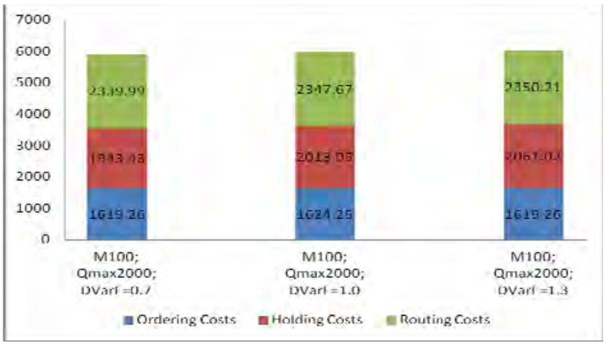
Figure 1. Flowchart of Proposed Tabu Search Heuristics



a) Best Objective Values with Varying Route Duration Limits, Order Quantity Capacity and Demand Variances



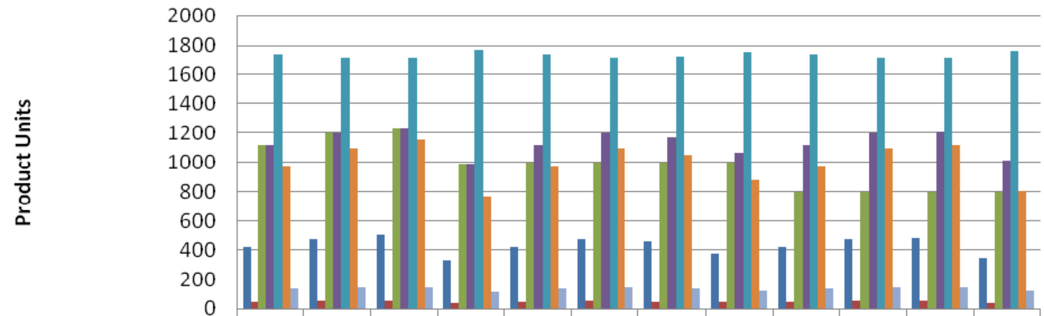
b) Three Component Costs With Varying Route Duration Limits and Order Quantity Capacity (DVarF=1.0)



c) Three Component Costs With Varying Demand Variance (Order Quantity Capacity=2000 and Route Duration Limit = 100)

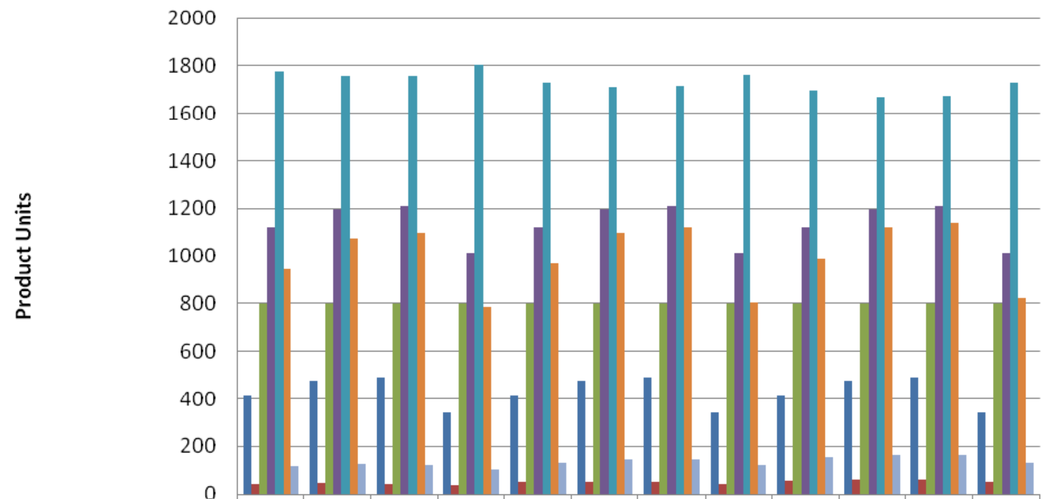
d) Three Component Costs With Varying Demand Variance (Order Quantity Capacity=800 and Route Duration Limit = 100)

Figure 2. Best Objective Values and Three Cost Components



	Qmax2000				Qmax1000				Qmax800			
	depot 1000	depot 1001	depot 1002	depot 1003	depot 1000	depot 1001	depot 1002	depot 1003	depot 1000	depot 1001	depot 1002	depot 1003
Mean Demand (unit/day)	417	476	505	326	417	476	455	376	417	476	487	344
SD of Demand (unit/day)	49	53	53	42	49	53	50	45	49	53	52	43
Optimal Order Quantity	1118	1195	1231	989	1000	1000	1000	1000	800	800	800	800
EOQ	1118	1195	1231	989	1118	1195	1168	1062	1118	1195	1209	1016
Avail Icap	1731	1709	1708	1768	1731	1709	1722	1751	1731	1709	1711	1764
Reorder Point	969	1098	1156	768	969	1098	1049	877	969	1098	1118	806
Safety Stock	135	146	146	116	135	146	139	125	135	146	144	118

a) With Varying Order Quantity Capacity (Route Duration Limit=80; DVarF=1.0)



	DVarF=0.7				DVarF=1.0				DVarF=1.3			
	depot 1000	depot 1001	depot 1002	depot 1003	depot 1000	depot 1001	depot 1002	depot 1003	depot 1000	depot 1001	depot 1002	depot 1003
Mean Demand (unit/day)	417	476	487	344	417	476	487	344	417	476	487	344
SD of Demand (unit/day)	41	44	44	36	49	53	52	43	55	60	59	49
Optimal Order Quantity	800	800	800	800	800	800	800	800	800	800	800	800
EOQ	1118	1195	1209	1016	1118	1195	1209	1016	1118	1195	1209	1016
Avail Icap	1775	1756	1758	1802	1731	1709	1711	1764	1693	1668	1671	1730
Reorder Point	947	1074	1095	787	969	1098	1118	806	987	1118	1139	823
Safety Stock	113	122	121	99	135	146	144	118	153	166	165	135

b) With Varying Demand Variances (Route Duration Limit=80; Order Quantity Capacity=800)

Figure 3. Continuous Inventory Control Policies at Four Warehouses

Table 1. Computational Results of Sequential MDVRP and ICP, and Combined ICP and MDVRP

	Sequential MDVRP and ICP		Combined MDVRP and ICP					
			Init. Sol. Type 1			Init. Sol. Type 2		
	Best Obj. ($/day)	CPU Time (min)	Best Obj. ($/day)	% Improve.	CPU Time (min)	Best Obj. ($/day)	% Improve.	CPU Time (min)
50-Customers Problems								
C1-50a-wh1	3,727.02	1.36	**3,599.25**	**3.43%**	1.96	3,619.87	2.87%	1.67
C1-50a-wh2	3,821.86	1.26	**3,727.73**	**2.46%**	0.89	3,747.72	1.94%	0.79
C1-50b-wh1	4,083.87	0.66	**3,728.71**	**8.70%**	0.61	3,795.24	7.07%	0.78
C1-50b-wh2	3,999.61	0.74	**3,478.46**	**13.03%**	0.85	3,823.94	4.39%	0.71
C2-50a-wh1	3,840.98	0.98	3,550.59	7.56%	0.88	**3,528.95**	**8.12%**	1.02
C2-50a-wh2	3,514.08	0.94	**3,223.65**	**8.26%**	0.95	3,369.86	4.10%	0.96
C2-50b-wh1	4,211.79	0.79	3,956.97	6.05%	0.53	**3,905.52**	**7.27%**	0.62
C2-50b-wh2	4,108.18	0.58	3,627.97	11.69%	0.69	**3,588.85**	**12.64%**	0.64
R1-50a-wh1	3,570.89	0.76	3,367.28	5.70%	0.65	**3,235.19**	**9.40%**	0.83
R1-50a-wh2	3,693.61	0.74	3,584.06	2.97%	0.55	**3,485.92**	**5.62%**	0.71
R1-50b-wh1	3,715.19	0.84	3,284.29	11.60%	0.80	**3,190.68**	**14.12%**	0.98
R1-50b-wh2	3,927.35	1.21	3,723.42	5.19%	0.65	**3,588.96**	**8.62%**	0.76
RC1-50a-wh1	4,228.72	0.61	4,085.40	3.39%	0.53	**4,029.67**	**4.71%**	0.62
RC1-50a-wh2	4,414.56	1.12	3,844.28	12.92%	0.68	**3,814.29**	**13.60%**	0.53
RC1-50b-wh1	3,707.06	0.64	**3,270.56**	**11.77%**	0.74	3,284.03	11.41%	0.82
RC1-50b-wh2	3,623.14	0.74	3,320.46	8.35%	0.31	**3,203.86**	**11.57%**	0.63
75-Customers Problems								
C1-75a-wh1	4,966.38	1.80	**4,839.85**	**2.55%**	1.76	4,885.40	1.63%	1.49
C1-75a-wh2	5,152.99	1.35	5,152.99	0.00%	1.06	**5,014.38**	**2.69%**	0.99
C1-75b-wh1	5,545.76	1.09	5,393.16	2.75%	1.17	**5,366.82**	**3.23%**	1.23
C1-75b-wh2	5,375.33	1.03	5,249.51	2.34%	1.02	**5,100.32**	**5.12%**	0.78
C2-75a-wh1	5,306.20	1.18	**4,991.70**	**5.93%**	1.05	5,022.92	5.34%	1.22
C2-75a-wh2	4,985.98	1.26	**4,695.13**	**5.83%**	1.19	4,834.15	3.05%	1.25
C2-75b-wh1	5,528.38	1.22	**5,311.50**	**3.92%**	0.96	5,330.93	3.57%	1.17
C2-75b-wh2	5,338.21	1.28	5,210.52	2.39%	1.06	**5,138.19**	**3.75%**	1.01
R1-75a-wh1	4,703.12	0.95	4,632.99	1.49%	1.02	**4,535.88**	**3.56%**	1.29
R1-75a-wh2	4,593.40	0.93	**4,412.45**	**3.94%**	0.88	4,497.09	2.10%	1.14
R1-75b-wh1	4,788.27	1.34	**4,378.67**	**8.55%**	1.07	4,523.30	5.53%	1.11
R1-75b-wh2	4,763.60	1.81	4,536.31	4.77%	1.19	**4,490.63**	**5.73%**	1.35
RC1-75a-wh1	4,988.65	1.05	4,827.74	3.23%	0.61	**4,815.07**	**3.48%**	0.99
RC1-75a-wh2	5,236.32	1.07	**5,104.13**	**2.52%**	0.91	5,236.32	0.00%	0.88
RC1-75b-wh1	4,971.99	1.00	4,785.77	3.75%	0.68	**4,760.34**	**4.26%**	0.82
RC1-75b-wh2	4,987.55	1.03	**4,745.55**	**4.85%**	0.87	4,750.22	4.76%	1.02
100-Customers Problems								
C1-100-wh1	6,061.05	1.71	5,815.44	4.05%	1.90	**5,761.22**	**4.95%**	1.75
C1-100-wh2	6,214.90	1.64	6,110.84	1.67%	1.42	**6,005.50**	**3.37%**	1.62
C2-100-wh1	6,142.82	1.50	5,872.85	4.39%	1.33	**5,872.55**	**4.40%**	1.25
C2-100-wh2	6,590.24	1.41	6,507.64	1.25%	1.28	**6,380.97**	**3.18%**	1.47
R1-100-wh1	5,616.66	1.24	5,264.36	6.27%	1.23	**5,227.34**	**6.93%**	1.82
R1-100-wh2	5,574.31	1.97	**5,430.94**	**2.57%**	1.70	5,471.92	1.84%	1.49
RC1-100-wh1	6,112.11	1.43	6,016.59	1.56%	1.36	**5,984.98**	**2.08%**	1.47
RC1-100-wh2	6,324.67	1.32	6,258.55	1.05%	1.15	**6,127.76**	**3.11%**	1.39

Table 2. Multi-Depot Vehicle Routing Policies for Four Warehouses with Varying Route Duration Limits (Order Quantity Capacity =2000 and DVarF=1.0)

	M80			
	depot 1000	depot 1001	depot 1002	depot 1003
No. of Routes	6	8	8	8
Routes	1000-98-55-69-82	1001-6-7-79-8	1002-65-90-96-94	1003-67-93-71
	1000-88-60-78-73	1001-46-4-45-5-3	1002-95-92-91-80	1003-72-54-81
	1000-14-47-17-16-15	1001-42-44	1002-66-56-84-64	1003-62
	1000-59-97-75	1001-1-43-40	1002-83-57-24-22	1003-51-85-63
	1000-9-13-87	1001-36-35-37	1002-20-49-19-18	1003-76-89
	1000-10-11-12-53	1001-38-39-41	1002-48-21-23-25	1003-33-32-30-28-26
		1001-70-61-68	1002-77-58	1003-27-29-31-34
		1001-100-2	1002-74-86-52-99	1003-50
Mean Demands	61; 68; 80; 70; 43; 95	80; 90; 20; 70; 70; 60; 53; 33	71; 46; 70; 87; 80; 70; 27; 54	26; 34; 3; 27; 56; 70; 80; 30
Travel Times to Last Customer	75.46; 77.83; 78.00;	73.52; 64.66; 55.24; 74.49;	76.91; 77.24; 76.56; 69.41;	57.10; 79.49; 15.83; 61.36;
	76.68; 62.90; 72.08	79.54; 77.58; 62.61; 44.77	72.52; 78.78; 65.74; 78.36	52.26; 79.94; 75.18; 13.61
Travel Times (begin and end at depot)	81.78; 102.04; 97.65;	81.59; 73.66; 88.78; 113.54;	101.98; 93.52; 88.23; 90.99;	80.64; 112.91; 21.66; 74.36;
	119.76; 85.75; 77.18	122.95; 113.60; 86.69; 45.77	104.80; 109.78; 98.55; 87.42	79.43; 105.02; 90.48; 17.21

	M100			
	depot 1000	depot 1001	depot 1002	depot 1003
No. of Routes	6	6	6	6
Routes	1000-98-69-90-65-82	1001-2-6-7-8-46	1002-91-92-94-96-80	1003-67-93-71
	1000-53-88-60-79-78	1001-4-45-5-3-1	1002-64-84-95-56-66	1003-85-62
	1000-12-47-17-16-15-13	1001-42-44-43-40-39	1002-83-22-24-57	1003-51-76-89-63
	1000-97-75-59	1001-36-35-37-38	1002-20-49-19-18-48-21	1003-33-32-34
	1000-99-86-74-87-9	1001-81-54-72-41	1002-23-25-77-58	1003-31-29-27-26-28-30
	1000-73-14-11-10	1001-70-61-68-55-100	1002-52	1003-50
Mean Demands	67; 98; 100; 70; 84; 85	90; 100; 80; 100; 54; 72	89; 76; 87; 100; 77; 3	26; 5; 81; 50; 100; 30
Travel Times to last customer	87.24; 97.47; 93.83;	66.00; 64.47; 98.24;	92.71; 97.31; 76.47;	57.10; 43.57; 93.68;
	90.76; 97.51; 87.34	93.14; 99.39; 99.68	96.52; 96.94; 21.66	57.32; 97.25; 13.61
Travel Times (begin and end at depot)	93.56; 114.50; 112.70;	71.66; 73.96; 136.45;	108.99; 103.39; 88.55;	80.64; 49.40; 106.68;
	118.07; 112.38; 99.42	133.34; 135.40; 112.41	127.92; 129.74; 33.32	72.62; 115.36; 17.21

A Modified Wall Building-Based Compound Approach for Container Loading Problem

by

Ampol Karoonsoontawong*, Ph.D.
Assistant Professor
Department of Civil Engineering, Faculty of Engineering
King Mongkut's University of Technology Thonburi
126 Pracha Utid Rd., Bangmod, Thung Khru
Bangkok 10900, Thailand
E-mail*: ampol.kar@kmutt.ac.th
(Corresponding Author)

and

Krongthong Heebkhoksung
Graduate Research Assistant
Department of Civil Engineering, Faculty of Engineering
King Mongkut's University of Technology Thonburi
126 Pracha Utid Rd., Bangmod, Thung Khru
Bangkok 10900, Thailand
E-mail: avil_never@hotmail.com

Submitted for Presentation at the 92nd Annual Meeting of the Transportation Research Board and Publication in the Transportation Research Record Series

15 November 2012

ABSTRACT

Efficient loading of containers would raise current productivity for the shipment of mixed, boxed cargo and this paper considers the knapsack container loading problem. Given a rectangular-shaped container, rectangular-shaped boxes with different sizes are packed such that total loaded volume is maximized. All boxes with the same origin-destination pair may be rotated in six orthogonal directions without load-related and positioning constraints. The modified wall-building based compound approach performs 36 modified wall-building heuristics based on three existing ranking functions, two existing priority rules and six orthogonal rotations of containers, while recording the best solution. The six orthogonal rotations of containers are equivalent to filling the container in six ways (four wall building methods and two floor building methods). Three weakly heterogeneous real-world test problems from a furniture company in Thailand are employed. There is not a winning heuristic that performs best on the three test problems. The typical wall-building approach (type-1 container rotation) does not perform well when compared with considering all six orthogonal rotations of container. In terms of the number of containers, the proposed compound approach can save up to 33% on the three test problems, and the highest fill percentages in the best solution founds are improved by up to 36%, when compared with the manual solutions. The proposed approach outperforms the existing tree heuristic. The highest fill percentages by the proposed approach are up to 6% higher than those by the tree search heuristic, whereas the CPU times by the proposed approach are up to 31% of those by the tree search heuristic.

INTRODUCTION

Container loading is a crucial function for efficient supply chain *(1)*. An inefficient container loading may result in inevitably additional container costs as well as unsatisfactory customer service level. The problem considered in this paper is the knapsack container loading problem. Given a rectangular-shaped container and rectangular-shaped boxes with associated volumes, the subset of boxes is selected to be packed in the container such that the total volume is maximized (i.e. the wasted space in the container is minimized). The cargo boxes may be rotated in any orthogonal directions without load-related and positioning constraints. It is noted that in principle, the empty spaces could be filled out with foam rubber to ensure a proper support of the boxes *(2)*. All boxes have the same origin-destination pair. It is assumed that the cargo weights are dominated by cargo volume in container packing, so box weights are not considered in the algorithm. It is also assumed that the boxes are packed without overlapping, and the widths, depths and heights of the boxes are integers. The modified wall-building based compound approach is proposed in this paper. It considers six orthogonal rotations of container together with three existing ranking functions and two existing priority rules for determining layer depths and strip heights, resulting in 36 modified wall-building heuristics. The compound approach performs the 36 heuristics while recording the best solution found.

In the next section, the literature review is provided, followed by the description of the modified wall-building based compound approach. The computational results of the three real-world case studies are discussed. The best solutions by the proposed approach are compared to the manual solutions and the best solutions found by the tree search heuristic by *(2)*. Then, the summary and conclusions are provided.

LITERATURE REVIEW

The container loading problem was first studied by Gilmore and Gomory *(3)*. Dyckhoff *(4)* and Wascher et al. *(5)* proposed the general classification of cutting and packing problems. Pisinger *(2)* categorized the packing and loading literature into four categories based on the objective function and side constraints: strip packing, knapsack container loading, bin-packing and multi-container loading. Firstly, in the strip packing problem (e.g. *(6)*), the container has known width and height but unlimited depth, and the problem is to pack all boxes such that the container depth is minimized. This problem category is applicable to multi-drop situations where the load should be divided into different sections associated with different destinations (e.g. *(7)*). Secondly, in the knapsack container loading problem (e.g. *(2)*,*(8)*), we select a subset of boxes with associated profits to be packed in a single container such that the total profit is maximized. If the box profit is set to the box volume, this problem minimizes the wasted space in the container. Thirdly, in the bin packing problem (e.g. *(9)*,*(25)*-*(27)*), all boxes have to be packed into a minimum number of containers with fixed dimensions. Lastly, in the multi-container loading problem (e.g. *(10)*,*(28)*), all boxes are packed into a minimum number of containers, which are chosen from the containers with varying dimensions, such that the total shipping cost is minimized. In addition, Bischoff and Ratcliff *(7)* and Bortfeldt and Wascher *(1)* provide the review of practical requirements that may be incorporated into the problem: container-related, item-related, cargo-related, positioning, and load-related constraints.

Since the container loading problem is an NP-hard problem *(2)*, there does not exist an efficient algorithm to obtain the exact solution in polynomial time. Christensen and Rousoe *(11)* provide the thorough review of heuristics for the container loading problem. The heuristics for the container loading problem can be categorized into three categories: construction algorithms, tree search algorithms, and metaheuristic algorithms.

First, in the category of construction algorithms, a general solution strategy is to divide the container into smaller pieces, and then each piece is separately packed. The three-dimensional solution space is often reduced to one or two dimensions. The construction algorithms can be further categorized into five subcategories: wall-building, layer-building, stack-building, block-building, and guillotine-cuts. The most common dividing procedure is wall-building first introduced by George and Robinson *(12)*. A wall is constructed by making a vertical strip through the container. The depth of a strip is defined by the depth of the first box placed in the wall, and as the strip is filled, the boxes will create a wall-like formation. Bischoff and Marriott *(6)* extended the wall building algorithm, and proposed a hybrid approach where 14 heuristics based on various ranking functions are performed to determine the best solution. The second sub-category is the layer-building approach (e.g. *(7)*). It splits the container by horizontal slices. When compared to the wall building approach, the layer building approach may produce more stable loads. The third sub-category is the stack-building approach (e.g. *(3)*). It constructs box stacks, so that the container loading problem becomes the two-dimensional problem of arranging the stacks on the container floor. The fourth sub-category is the block-building approach (e.g. *(13)*). It constructs blocks, and then, these blocks are placed in the container. A block is composed of one or two types of boxes that are tightly packed. The fifth sub-category is the guillotine-cut approach *(14)*. It splits the container into smaller pieces by guillotine cuts. The guillotine cut is a cut through an object until another guillotine cut is met or the object is cut through.

Second, the tree search heuristic algorithms are developed based on dynamic programming scheme and certain construction algorithm, employing upper and lower bounds. Morabito and Arenales *(14)* proposed the tree search where a tree is created by guillotine cutting. Eley *(13)* proposed the tree search method that is built upon the greedy block building procedure. Pisinger *(2)* proposed the tree search heuristics based on the wall building procedure. The tree search framework is used to determine the wall depths and strip widths, and only the most promising nodes are kept in the tree. Third, the metaheuristic algorithms for container loading problem include genetic algorithms (e.g. *(15)*,*(16)*), GRASP algorithms (e.g. *(17)*), tabu search algorithms (e.g. *(18)*) and hybrid algorithms (e.g. *(19)*).

The problem considered in this paper is the knapsack container loading problem. To fill each container, the total loaded volume is maximized. In this study, we extend Bischoff and Marriott *(6)*'s approach by considering various orthogonal rotations of container and incorporating Pisinger *(2)*'s box pairing procedure and Pisinger *(20)*'s dynamic programming algorithm for 0-1 knapsack problem. In the case studies, the proposed approach is compared to the manual approach and the tree search heuristic *(2)*.

MODIFIED WALL-BUILDING BASED COMPOUND APPROACH

George and Robinson *(12)*'s wall building algorithm fills the single container by building layers (walls) across the container depth. The layer depth is selected based on the rationale that a box with the largest size of the smallest dimension may be difficult to accommodate later in the packing procedure. As such, the ranking rule is set the layer depth equal to the largest size of the smallest dimensions of the unpacked boxes. Given a known layer depth, the horizontal strips are built across the container height. To fill a horizontal strip, the algorithm inserts the box with the largest size of the smallest dimension of an unpacked box. Bischoff and Marriott *(6)* proposed the compound approach that performs the wall-building algorithms with various ranking rules while recording the best solution found. In this paper, we modify the compound approach by considering six orthogonal rotations of the container, Pisinger *(2)*'s three ranking functions and two priority rules, Pisinger *(2)*'s box pairing procedure, and Pisinger *(20)*'s dynamic programming based algorithm for the exact solution of the 0-1 knapsack strip packing problem.

The dimensions $\overline{W}$, $\overline{H}$, and $\overline{D}$ are referred to the typical width, height and depth of the container, whereas the current width, height and depth of container (*W*, *H* and *D*) are referred to the dimensions considered in the modified wall-building algorithm along the x-axis, y-axis and z-axis, respectively, as shown in Figure 1.

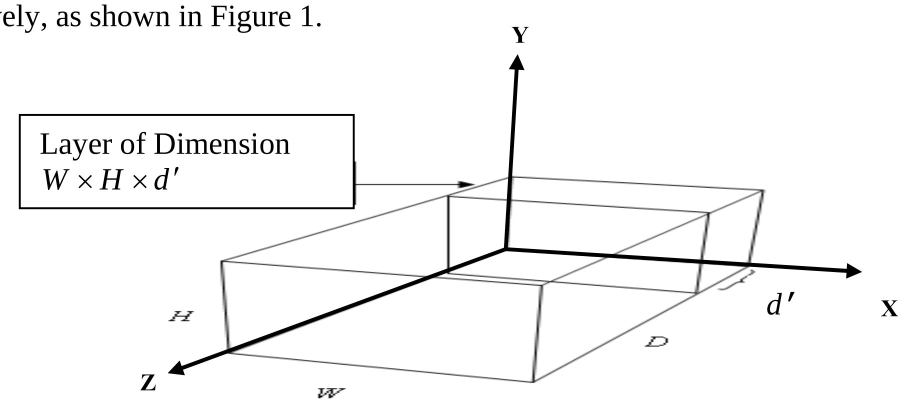
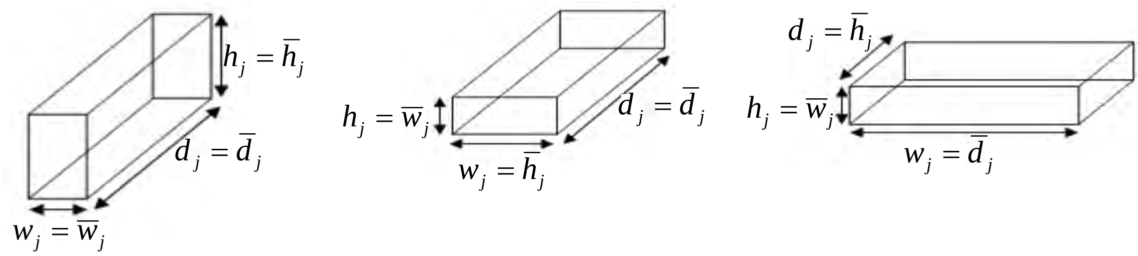
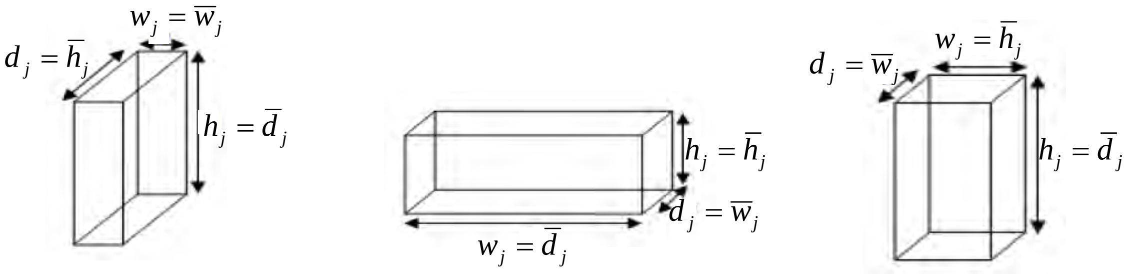


Figure 1. Three Dimensional Axes, Current Container Dimensions (*W*, *H*, *D*) and Current Layer Depth (d') *(2)*

In the same way, the dimensions $\overline{w}_j$, $\overline{h}_j$ and $\overline{d}_j$ are the initial width, height and depth of box *j*, whereas the current dimensions of boxes (w_j, h_j and d_j) are referred to the dimensions considered in the algorithm along the three axes. All boxes can be rotated in six orthogonal directions (or rotations) as illustrated in Figure 2. The modified wall building based compound approach further considers six possible container rotations in the procedure. As shown in Table 1, the container rotation types 1, 2, 5, and 6 correspond to wall building approaches, whereas the container rotation types 3 and 4 correspond to floor building approaches. Container rotation type 1 (i.e. typical container rotation) builds layers (walls) across the container depth $\overline{D}$ as shown in Figure 1, and builds horizontal strips of length $\overline{W}$ across the container height $\overline{H}$ as shown in Figure 3. For container rotation types 2-6, the procedure rotates the container in the other five orthogonal directions, and performs the wall-building algorithm; these corresponds to either wall or floor building and either horizontal or vertical strip building. Specifically, container rotation type 2 corresponds to building layers (walls) across the container depth $\overline{D}$ and vertical strips of length $\overline{H}$ across the container width $\overline{W}$. Container rotation type 3 corresponds to building layers (floors) across the container height $\overline{H}$ and horizontal strips of length $\overline{D}$ across the container width $\overline{W}$. Container rotation type 4 corresponds to building layers (floors) across the container height $\overline{H}$ and horizontal strips of length $\overline{W}$ across the container depth $\overline{D}$.



a) Rotation Type 1 b) Rotation Type 2 c) Rotation Type 3



d) Rotation Type 4 e) Rotation Type 5 f) Rotation Type 6

Figure 2. Six Orthogonal Rotations

Table 1. Container Rotation Types and Associated Descriptions

Container Rotation Type	W	H	D	Description
1	$\overline{W}$	$\overline{H}$	$\overline{D}$	Wall Building: Layer building across the depth $\overline{D}$ Horizontal strip (strip length = $\overline{W}$) building across the height $\overline{H}$
2	$\overline{H}$	$\overline{W}$	$\overline{D}$	Wall Building: Layer building across the depth $\overline{D}$ Vertical strip (strip length = $\overline{H}$) building across the width $\overline{W}$
3	$\overline{D}$	$\overline{W}$	$\overline{H}$	Floor Building: Layer building across the height $\overline{H}$ Horizontal strip (strip length = $\overline{D}$) building across the width $\overline{W}$
4	$\overline{W}$	$\overline{D}$	$\overline{H}$	Floor Building: Layer building across the height $\overline{H}$ Horizontal strip (strip length = $\overline{W}$) across the depth $\overline{D}$
5	$\overline{D}$	$\overline{H}$	$\overline{W}$	Wall Building: Layer building across the width $\overline{W}$ Horizontal strip (strip length = $\overline{D}$) Across the height $\overline{H}$
6	$\overline{H}$	$\overline{D}$	$\overline{W}$	Wall Building: Layer building across the width $\overline{W}$ Vertical strip (strip length = $\overline{H}$) building across the depth $\overline{D}$

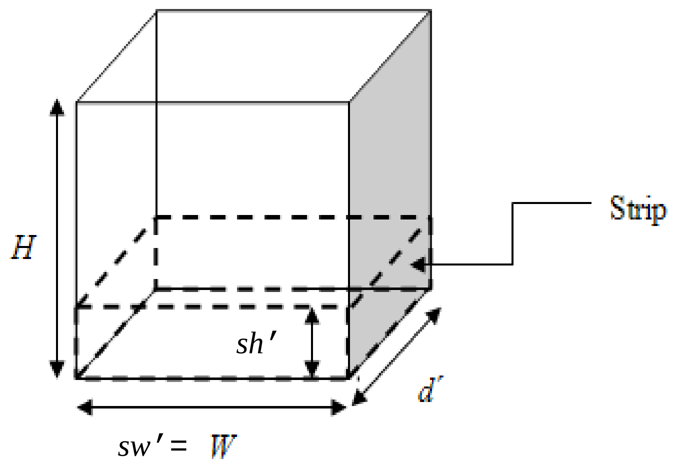


Figure 3. Current Strip Dimensions ($sw' \times sh' \times d'$)

Container rotation type 5 corresponds to building layers (walls) across the container width $\overline{W}$ and horizontal strips of length $\overline{D}$ across the container height $\overline{H}$. Container rotation type 6 corresponds to building layers (walls) across the container width $\overline{W}$ and vertical strips of length $\overline{H}$ across the container depth $\overline{D}$.

The notations used in the proposed procedure including the parameters and variables are first given. Then, the pseudo-code is described, followed by the descriptions of major components in the pseudo-code.

Notations

<u>Parameters</u>

$N = \{1,\ldots,n\}$ = set of boxes
$\overline{W}$ = typical container width
$\overline{H}$ = typical container height
$\overline{D}$ = typical container depth
$\overline{w}_j$ = initial width of box j
$\overline{h}_j$ = initial height of box j
$\overline{d}_j$ = initial depth of box j

<u>Variables</u>

N' = set of unloaded boxes
N'' = set of unloaded boxes that are feasible to fill the current layer
N''' = set of unloaded boxes that are feasible to fill the current strip
W = current container width in the algorithm
H = current container height in the algorithm
D = current container depth in the algorithm
w_j = current width of box j

h_j = current height of box j

d_j = current depth of box j

D_r = current residual container depth

H_r = current residual wall height

d' = current layer depth

sw' = current strip width

sh' = current strip height

strip_start_y = y-coordinate of the beginning of current strip

strip_end_y = y-coordinate of the ending of current strip

wall_start_z = z-coordinate of the beginning of current layer

wall_end_z = z-coordinate of the ending of current layer

(x_j, y_j, z_j) = (x,y,z)-coordinate of the referenced corner of loaded box j in the current solution

(dx_j, dy_j, dz_j) = (width, height, depth) of loaded box j in the current solution

(x_j^*, y_j^*, z_j^*) = (x,y,z)-coordinate of the referenced corner of loaded box j in the best solution found

(dx_j^*, dy_j^*, dz_j^*) = (width, height, depth) of loaded box j in the best solution found

Pseudo-code

For each heuristic method c,u,v (i.e. container rotation type c, ranking function u and priority rule v), the following steps are performed, given a rectangular-shaped container.

Step 0: Set con = 1.
Set $N' = N$.

Step 1: Initialize the following for the container con :

- set the current residual container depth to the current container depth: set D_r=D,
- set the current strip width to the current container width: set $sw' = W$, and
- Set *wall_end_z* = 0
- set (x_j, y_j, z_j) = (0,0,0) for all boxes j in N'
- set $(dx_j, dy_j, dz_j) = (\overline{w}_j, \overline{h}_j, \overline{d}_j)$for all boxes j in N'

Step 2: Determine the current layer depth d' based on the ranking function f^u and the priority rule v and D_r. If d' can be determined,

- update the current residual container depth: set $D_r = D_r - d'$
- set the current residual wall height to the current container height: set H_r=H
- set *wall_start_z* = *wall_end_z*
- set *wall_end_z* = *wall_start_z* + d'
- set *strip_end_y* = 0

Otherwise, go to Step 7.

Step 3: Perform the box pairing procedure to obtain the set of feasible unloaded boxes (N'') to fill the current wall. Update (dx_j, dy_j, dz_j) for all rotated boxes j.

Step 4: Determine the current strip height (sh') based on the ranking function f^u and the priority rule v, given H_r. If sh' can be determined,

- update the current residual wall height: set $H_r = H_r - sh'$
- determine the set of feasible unloaded boxes (N''') to fill the current strip, and update (dx_j , dy_j , dz_j) for all rotated boxes j.
- set *strip_start_y* = *strip_end_y*
- set *strip_end_y* = *strip_start_y* + sh'

Otherwise, go to Step 2.

Step 5: Perform the strip packing procedure to select boxes from N''' to fill the strip ($sw' \times sh' \times d'$) and update (x_j, y_j ,z_j) and (dx_j , dy_j , dz_j) of each loaded box j. Update the sets of unloaded boxes N' and N''.

Step 6: If there is an unloaded box (i.e. $N' \neq \{\}$), go to Step 4. Otherwise, go to Step 8.

Step 7: Calculate the fill percentage of the current container *con*: fill percentage = volume of loaded boxes / volume of container. Set *con* = *con* + 1. Go to Step 1.

Step 8: Terminate with the empty set of unloaded boxes ($N' = \phi$). Calculate the fill percentage of the current container *con*.

Step 9: Update the best solution found:

- If the current solution is better in terms of the number of containers,
  - set best heuristic method (c^*,u^*,v^*) = current heuristic method (c,u,v)
  - set (x_j^*, y_j^*, z_j^*) = (x_j, y_j , z_j) and set (dx_j^*, dy_j^*, dz_j^*) = (dx_j , dy_j , dz_j) for all j in N.
- If the current solution is as good as the best solution in terms of number of containers,
  - If the higher fill% in the current solution is higher than that in the best solution,
    - set best heuristic method (c^*,u^*,v^*) = current heuristic method (c,u,v)
    - set (x_j^*, y_j^*, z_j^*) = (x_j, y_j , z_j) and set (dx_j^*, dy_j^*, dz_j^*) = (dx_j , dy_j , dz_j) for all j in N.

Descriptions of Major Components

The major components of the proposed approach include the layer depth and strip height determinations, the box pairing procedure, and the strip packing problem.

Layer Depth and Strip Height Determinations

In Step 2, we employ the ranking functions in *(2)*, which are based on certain statistics of the dimensions of the unloaded boxes. Denote by α and β the smallest dimension and the largest dimension of the unloaded boxes, respectively. Three different ranking functions are considered:

$$f_k^1 = \sum_{i=1}^{n} 1_{(w_i = k \vee h_i = k \vee d_i = k)} \quad \forall k = \alpha, \alpha+1, ..., \beta-1, \beta \tag{1.1}$$

$$f_k^2 = \sum_{i=1}^{n} 1_{(\max\{w_i, h_i, d_i\} = k)} \quad \forall k = \alpha, \alpha+1, ..., \beta-1, \beta \tag{1.2}$$

$$f_k^3 = \sum_{i=1}^{n} 1_{(\min\{w_i, h_i, d_i\}=k)} \quad \forall k = \alpha, \alpha+1, ..., \beta-1, \beta \tag{1.3}$$

The type-1 ranking function, Eq.(1.1), determines the number of occurrences of each dimension from all dimensions w_i, h_i and d_i of the remaining boxes. The type-2 ranking function, Eq.(1.2), determines the number of occurrences of each dimension from the largest dimensions of the unloaded boxes. The type-3 ranking function, Eq.(1.3), determines the number of occurrences of each dimension from the smallest dimensions of the unloaded boxes.

In Step 4, when the layer depth and the set of feasible unloaded boxes (N'') to fill the current wall have been determined, the ranking functions only consider the current width and the current height of feasible unloaded boxes. Denote by α and β the respective smallest dimension and the largest dimension of the current widths and current heights of the feasible unloaded boxes. Three different ranking functions are:

$$f_k^1 = \sum_{i=1}^{n} 1_{(w_i = k \vee h_i = k)} \quad \forall k = \alpha, \alpha+1, ..., \beta-1, \beta \tag{2.1}$$

$$f_k^2 = \sum_{i=1}^{n} 1_{(\max\{w_i, h_i\}=k)} \quad \forall k = \alpha, \alpha+1, ..., \beta-1, \beta \tag{2.2}$$

$$f_k^3 = \sum_{i=1}^{n} 1_{(\min\{w_i, h_i\}=k)} \quad \forall k = \alpha, \alpha+1, ..., \beta-1, \beta \tag{2.3}$$

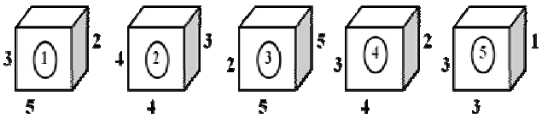
The type-1 ranking function, Eq.(2.1), determines the number of occurrences of each dimension from all dimensions w_i and h_i of the feasible unloaded boxes. The type-2 ranking function, Eq.(2.2), determines the number of occurrences of each dimension from the largest dimensions of w_i and h_i of the feasible unloaded boxes. The type-3 ranking function, Eq.(2.3), determines the number of occurrences of each dimension from the smallest dimensions w_i and h_i of the feasible unloaded boxes.

In Steps 2 and 4, we consider two priority rules *(2)*:

Priority Rule 1: the largest dimension with positive ranking function value is selected; i.e. largest dimension k with $f_k > 0$.

Priority Rule 2: the most frequent dimension is selected; i.e. dimension k with the largest value of f_k.

The motivation of priority rule 1 is that the largest dimension should be loaded early in the packing procedure; otherwise, it may be difficult to be packed later. The motivation of priority rule 2 is that a homogeneous layer or strip with may tightly be packed. It is noted that the procedure by *(12)* is equivalent to the type-3 ranking function and priority rule 1. Figure 4 illustrates the layer depth determination on a small problem by the three ranking functions and the two priority rules.



a) Five-Box Example

Dimension (k)	f_k^1
1	1
2	3
3	5
4	3
5	3

Priority rule 1: d'=5
Priority rule 2: d'=3

b) First Ranking Function and Two Priority Rules

Dimension (k)	f_k^2
1	0
2	0
3	1
4	2
5	2

Priority rule 1: d'=5
Priority rule 2: d'=5

c) Second Ranking Function and Two Priority Rules

Dimension (k)	f_k^3
1	1
2	3
3	1
4	0
5	0

Priority rule 1: d'=3
Priority rule 2: d'=2

d) Third Ranking Function and Two Priority Rules

Figure 4. Examples of Layer Depth Determinations

Box Pairing Procedure

After the layer depth d' is determined, in Step 3, we employ the box pairing procedure *(2)* to determine the set of feasible unloaded boxes (N'') to fill the current wall. *(2)* indicated that a box pairing procedure can be used to achieve an improved solution in his tree search heuristic, and this is also adopted in our proposed algorithm. The complexity of the box pairing procedure is $O(n^2)$, and it is executed only once for each layer depth d'. The box pairing procedure is described below:

Step 3.1: Set $N''=\{\}$. If the smallest dimension of each box i in the set N' is bigger than the layer depth d', box i is not inserted in N'' and is not considered in the box pairing procedure. Otherwise, Rotate each box i in the set N' such that its depth d_i is the largest dimension satisfying the constraint $d_i \le d'$. The filling ratio to pack box i in the layer with depth d' is $\mu(i)$:

$$\mu(i) = w_i h_i d_i / w_i h_i d' = d_i / d' \qquad (3)$$

Step 3.2: Pairing box i with other box j in the set N' where $j \ne i$. All orthogonal rotations of i and j are considered such that $d_i + d_j \le d'$ and the associated filling ratio, $\eta(i, j)$, is determined:

$$\eta(i, j) = \frac{w_i h_i d_i + w_j h_j d_j}{d' \cdot \max\{w_i, w_j\} \cdot \max\{h_i, h_j\}} \qquad (4)$$

If $\eta(i,j) \le \mu(i)$ for all boxes $j \ne i$ in N' and all orthogonal rotations of boxes *i* and *j*, then box *i* remains alone and is inserted in N''. Otherwise, box *j* and corresponding rotation with the largest value of $\eta(i,j)$ is selected to pair with boxes *i* in order to form a new box *k* with the following dimensions: $w_k = \max\{w_i, w_j\}$, $h_k = \max\{h_i, h_j\}$, and $d_k = d_i + d_j$ as illustrated in Figure 5. Then, box *k* is inserted in N''.

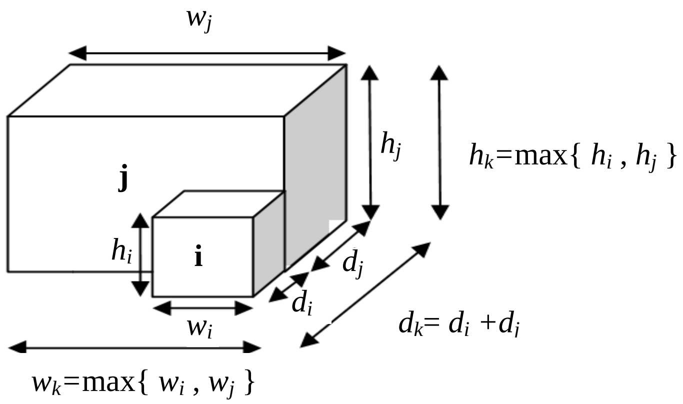


Figure 5. Paired Box Dimensions

Strip Packing Problem

In Step 5, strips are filled horizontally. The strip has the width equal to current container width $sw'=W$, the depth equal to current layer depth d' and the height sh'. The procedure first determines the set of feasible unloaded boxes (N''') to fill the current strip as follows. Set $N'''=\{\}$. Each box *j* in the set N'' is rotated in one of six directions such that w_j is minimized subject to $d_j \le d'$ and $h_j \le sh'$. If it is possible to fit box *j* in the current strip, box *j* is inserted in the set N'''. If it is not possible to fit box *j* within the current strip, then box *j* is not considered for the current strip packing. The strip packing problem can be formulated as a 0-1 knapsack problem as shown below *(2)*:

$$\max \sum_{j \in N'''} w_j \cdot h_j \cdot d_j \cdot s_j$$

Subject to

$$\sum_{j \in N'''} w_j \cdot s_j \le W$$

$$s_j \in \{0,1\} \quad \forall j \in N'''$$

where s_j is a binary decision variable; $s_j=1$ if box *j* is chosen to fill the current strip, and 0 otherwise. The 0-1 knapsack problem is an NP-hard problem *(21)*, so there does not exist an efficient algorithm to solve for an exact solution in polynomial time. It can be solved in pseudo-polynomial time by dynamic programming *(22)*. In this study, we employ the effective dynamic programming-based algorithm by *(20)*.

COMPUTATIONAL EXPERIENCES

The 36 modified wall building heuristics are implemented in C by modifying the callable C code in *(23,24)*. The box pairing procedure is taken from the callable C code by *(23)*, and the dynamic programming heuristic for 0-1 knapsack problem is taken from the callable C code by *(24)*. These heuristics run on a computer with 1.73 GHz Intel Core i7 processor and 4 GB of RAM, running under Windows 7. We use three real-world test problems from a furniture company in Thailand. The origin of the cargos is Thailand, and the destinations of the cargos in the three test problems are Brunei, Vietnam and Japan, respectively. The initial dimensions of boxes for the three test problems are shown in Tables 2a-2c. These three test problems are weakly heterogeneous, since the numbers of box types are less than 20 *(2)*. The standard container types are 40 HQ', 40' and 20' as shown in Table 2d. In this experiment, we employ the proposed algorithms to fill each container in the manual solutions in a descending order of the container size. When one container has a higher fill rate, it is more possible to use a smaller container (than that in the manual solutions) to pack the left boxes. As such, for the first test problem, all heuristics employ 40 HQ' containers. For the second test problem, all heuristics employ 40' containers. For the third test problem, all heuristics employ 40' container as the first container and 20' container as the second container.

Tables 3-5 show the computational results for test problems 1, 2 and 3, respectively. It is noted that the cargo weights on each container in the solutions do not exceed the allowable weight. The heuristic method c,u,v is referred to the container rotation type c, ranking function f^u and priority rule v. Apparently, there is not a winning heuristic that performs best on the three test problems. On the first test problem, there are 11 heuristics that yield two 40HQ' containers and 25 heuristics that yield three 40HQ' containers. The heuristic method c=6,u=2,v=2 performs best on the first test problem with two 40HQ' containers and the highest fill percentage (87.82%) of container number 1. On the second test problem, there are 34 heuristics that yield two 40'-containers and only 2 heuristics (c=3,u=2,v=1 and c=5,u=2,v=1) that yield three 40' containers. The heuristic method c=6,u=3,v=1 performs best on the second test problem with two 40'-containers and the highest fill percentage (80.75%) of container number 1. As can be seen in Table 5, in the best solution found, the fill percentage (20.50%) of container number 2 is equal to 15,908,120 cm^3 which can fill a 20' container. Thus, the best solution found becomes a 40'-container with 80.75% fill and a 20'-container with 41.03% fill. On the third test problem, there are 33 heuristics that yield two containers (40' and 20'). The three heuristics (c=1,u=1,v=2; c=3,u=1,v=2; and c=4,u=1,v=2) perform best on the third test problem, yielding a single 40'-container with the highest fill percentage (68.42%). Interestingly, the typical wall building algorithms that are associated with the container rotation type 1 (c=1) do not perform well; thus, this reiterates the improvement by considering the six orthogonal rotations of container.

Subsequently, the best solutions found on the three test problem by the proposed compound approach are compared to the manual solutions by the furniture company as well as the best solution found by the tree search heuristic by *(2)* as shown in Table 6. The manual solutions by the furniture company employ three 40HQ' containers for the first test problem, two 40'-containers for the second test problem and a 40'-container and a 20'-container for the third test problem. In terms of the number of containers, the proposed compound approach can save 33.33%, 25% and 33.33% on the three test problems, respectively. The highest fill percentages in the best solution founds are improved by 36.45%, 29.42% and 14.94% on test problems 1, 2 and 3, respectively, when compared with the manual solutions. Figure 6 illustrates the box layouts of manual solution and best solution found on test problem 3.

The tree search heuristic yields the same results in terms of number of containers as the proposed compound approach. However, the proposed approach yields improvement in highest fill percentage on test problems 1 and 2 by 6.52% and 1.31%, respectively. Especially, the total CPU times by the proposed approach are only 20.30%, 3.35% and 31.34% of those by the tree search heuristic.

Table 2. Initial Dimensions of Boxes for Three Test Problems and Dimensions of Standard Containers

a) Initial Dimensions (Centimeters) of Boxes for Test Problem 1 (223 Boxes and 16 Box Types)

Box Type	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Width ($\overline{w}_j$)	100	70	96	100	85	85	85	102	80	212	80	102	102	145	70	69
Depth ($\overline{d}_j$)	100	70	70	100	190	148	195	160	155	80	170	102	102	67	68	69
Height ($\overline{h}_j$)	80	80	53	60	100	100	100	90	90	90	65	77	57	83	100	92
Number of Boxes	1	3	2	3	3	4	3	3	3	4	3	6	1	16	11	157

*Total Box Volume = 129,402,900 Cubic Centimeters

b) Initial Dimensions (Centimeters) of Boxes for Test Problem 2 (113 Boxes and 14 Box Types)

Box Type	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Width ($\overline{w}_j$)	90	92	110	100	80	100	110	110	110	100	100	100	100	70
Depth ($\overline{d}_j$)	100	142	110	100	95	195	110	110	110	210	210	180	180	73
Height ($\overline{h}_j$)	101	101	75	115	92	70	90	55	45	55	60	60	26	92
Number of boxes	12	1	8	4	5	3	1	1	1	1	9	1	1	65

* Total box volume = 78,578,264 cubic centimeters

c) Initial Dimensions (Centimeters) of Boxes for Test Problem 3 (94 Boxes and 11 Box Types)

Box Type	1	2	3	4	5	6	7	8	9	10	11
Width ($\overline{w}_j$)	120	120	120	91	130	90	130	110	80	110	70
Depth ($\overline{d}_j$)	120	120	190	210	244	170	170	190	80	210	73
Height ($\overline{h}_j$)	70	40	70	70	80	70	70	45	70	60	92
Number of boxes	1	1	1	1	1	2	2	1	3	1	80

*Total Box Volume = 53,097,700 cubic centimeters

d) Typical Dimensions of Standard Container Types

Container Type	$\overline{W}$	$\overline{H}$	$\overline{D}$
40 HQ'	243 cm (8 ft)	292 cm (9.6 ft)	1219 cm (40 ft)
40'	243 cm (8 ft)	262 cm (8.6 ft)	1219 cm (40 ft)
20'	243 cm (8 ft)	262 cm (8.6 ft)	609 cm (20 ft)

Table 3. Computational Results for Test Problem 1
(All Containers are 40 HQ' with 86,495,364 Cubic Centimeters)

Heuristic Method c,u,v (CPU Time)	Cont. No.	Fill %
1, 1, 1 (3.88 sec)	1	77.03
	2	72.13
	3	0.45
2, 1, 1 (3.23 sec)	1	70.7
	2	77.04
	3	1.87
3, 1, 1 (0.39 sec)	1	55.39
	2	80.7
	3	13.51
4, 1, 1 (4.79 sec)	1	76.97
	2	72.64
5, 1, 1 (8.73 sec)	1	61.42
	2	53.81
	3	34.37
6, 1, 1 (1.06 sec)	1	67.5
	2	66.69
	3	15.42
1, 1, 2 (0.61 sec)	1	81.11
	2	68.49
2, 1, 2 (0.46 sec)	1	72.95
	2	73.67
	3	2.99
3, 1, 2 (0.10 sec)	1	79.04
	2	67.24
	3	3.33
4, 1, 2 (0.39 sec)	1	81.28
	2	68.33
5, 1, 2 (0.28 sec)	1	81.48
	2	64.3
	3	3.83
6, 1, 2 (1.38 sec)	1	81.9
	2	63.88
	3	3.83

Heuristic Method c,u,v (CPU Time)	Cont. No.	Fill %
1, 2, 1 (0.48 sec)	1	77.03
	2	72.13
	3	0.45
2, 2, 1 (0.55 sec)	1	71.36
	2	77.8
	3	0.45
3, 2, 1 (0.23 sec)	1	55.39
	2	80.7
	3	13.51
4, 2, 1 (6.19 sec)	1	76.97
	2	72.64
5, 2, 1 (8.64 sec)	1	61.42
	2	53.81
	3	34.37
6, 2, 1 (0.27 sec)	1	58.84
	2	70.58
	3	20.19
1, 2, 2 (0.45 sec)	1	79.51
	2	70.1
2, 2, 2 (0.37 sec)	1	79.5
	2	69.65
	3	0.45
3, 2, 2 (0.09 sec)	1	78.17
	2	52.61
	3	18.82
4, 2, 2 (0.31 sec)	1	77.99
	2	24.86
	3	46.76
5, 2, 2 (0.10 sec)	1	86.13
	2	59.94
	3	3.53
6, 2, 2 (0.22 sec)	**1**	**87.82**
	2	**61.78**

Heuristic Method c,u,v (CPU Time)	Cont. No.	Fill %
1, 3, 1 (1.58 sec)	1	78.24
	2	71.37
2, 3, 1 (0.63 sec)	1	68.99
	2	62.42
	3	18.19
3, 3, 1 (0.36 sec)	1	79.66
	2	69.95
4, 3, 1 (2.39 sec)	1	79.15
	2	70.46
5, 3, 1 (0.35 sec)	1	86.5
	2	63.11
6, 3, 1 (4.03 sec)	1	82.4
	2	67.21
1, 3, 2 (0.89 sec)	1	73.21
	2	68.15
	3	8.25
2, 3, 2 (0.60 sec)	1	80.21
	2	66.83
	3	2.56
3, 3, 2 (0.12 sec)	1	79.52
	2	66.06
	3	4.02
4, 3, 2 (0.57 sec)	1	68.75
	2	71.12
	3	9.73
5, 3, 2 (3.57 sec)	1	79.92
	2	61.28
	3	8.41
6, 3, 2 (15.85 sec)	1	78.91
	2	63.83
	3	6.87

Note: c=container rotation type; u = ranking function; v = priority rule.
Total CPU time = 74.14 seconds
The best solution found is bold and underlined.

Table 4. Computational Results for Test Problem 2
(All Containers are 40' with 77,608,854 Cubic Centimeters)

Heuristic Method *c*,*u*,*v* (CPU Time)	Cont. No.	Fill %
1, 1, 1 (0.69 sec)	1	77.62
	2	23.62
2, 1, 1 (0.53 sec)	1	67.93
	2	33.32
3, 1, 1 (0.13 sec)	1	35.56
	2	65.69
4, 1, 1 (0.28 sec)	1	64.85
	2	36.4
5, 1, 1 (0.12 sec)	1	36.25
	2	65
6, 1, 1 (6.81 sec)	1	68.54
	2	32.71
1, 1, 2 (0.35 sec)	1	73.6
	2	27.65
2, 1, 2 (0.33 sec)	1	71.69
	2	29.56
3, 1, 2 (0.06 sec)	1	77.68
	2	23.57
4, 1, 2 (0.19 sec)	1	76.79
	2	24.46
5, 1, 2 (0.08 sec)	1	69.42
	2	31.82
6, 1, 2 (0.19 sec)	1	72.64
	2	28.61

Heuristic Method *c*,*u*,*v* (CPU Time)	Cont. No.	Fill %
1, 2, 1 (0.41 sec)	1	73.42
	2	27.82
2, 2, 1 (0.24 sec)	1	70.96
	2	30.29
3, 2, 1 (0.10 sec)	1	35.56
	2	63.99
	3	1.7
4, 2, 1 (0.12 sec)	1	61.22
	2	40.03
5, 2, 1 (0.08 sec)	1	34.85
	2	59.32
	3	7.07
6, 2, 1 (0.11 sec)	1	68.54
	2	32.71
1, 2, 2 (0.56 sec)	1	71.16
	2	30.09
2, 2, 2 (0.61 sec)	1	72.01
	2	29.24
3, 2, 2 (0.16 sec)	1	78.43
	2	22.82
4, 2, 2 (0.55 sec)	1	71.26
	2	29.99
5, 2, 2 (0.07 sec)	1	70
	2	31.24
6, 2, 2 (0.33 sec)	1	71.27
	2	29.98

Heuristic Method *c*,*u*,*v* (CPU Time)	Cont. No.	Fill %
1, 3, 1 (0.41 sec)	1	70.61
	2	30.64
2, 3, 1 (0.26 sec)	1	74.25
	2	27
3, 3, 1 (0.06 sec)	1	73.64
	2	27.61
4, 3, 1 (0.16 sec)	1	79.48
	2	21.77
5, 3, 1 (0.08 sec)	1	70.6
	2	30.65
<u>6, 3, 1</u> (0.11 sec)	**<u>1</u>**	**<u>80.75</u>**
	<u>2</u>	**<u>20.50</u>***
1, 3, 2 (0.31 sec)	1	80.74
	2	20.51
2, 3, 2 (0.36 sec)	1	64.43
	2	36.82
3, 3, 2 (0.08 sec)	1	68.18
	2	33.07
4, 3, 2 (0.19 sec)	1	77.18
	2	24.07
5, 3, 2 (0.08 sec)	1	68.27
	2	32.98
6, 3, 2 (0.18 sec)	1	70.61
	2	30.64

Note: * 20.50% of 40' container = 15,908,120 cm^3; this can fill a 20' container (38,772,594 cm^3)
c=container rotation type; *u* = ranking function; *v* = priority rule.
Total CPU time =15.38 seconds
The best solution found is bold and underlined.

Table 5. Computational Results for Test Problem 3
(Container No.1 and No.2 are 40' and 20' with 77,608,854 and 38,772,594 cm^3, respectively)

Heuristic Method *c,u,v* (CPU Time)	Cont. No.	Fill %
1, 1, 1 (0.65 sec)	1	67.26
	2	2.31
2, 1, 1 (0.8 sec)	1	65.45
	2	5.95
3, 1, 1 (0.5 sec)	1	30.34
	2	76.22
4, 1, 1 (0.15 sec)	1	58.2
	2	20.44
5, 1, 1 (0.8 sec)	1	30.92
	2	75.06
6, 1, 1 (0.14 sec)	1	58.78
	2	19.29
1, 1, 2 (0.25 sec)	**1**	**68.42**
2, 1, 2 (0.27 sec)	1	62.72
	2	11.4
3, 1, 2 (0.05 sec)	**1**	**68.42**
4, 1, 2 (0.11 sec)	**1**	**68.42**
5, 1, 2 (0.05 sec)	1	63.42
	2	10.01
6, 1, 2 (0.13 sec)	1	64.98
	2	6.88
1, 2, 1 (0.66 sec)	1	63.6
	2	9.64
2, 2, 1 (0.67 sec)	1	65.45
	2	5.95
3, 2, 1 (0.56 sec)	1	30.34
	2	76.22
4, 2, 1 (0.15 sec)	1	58.2
	2	20.44
5, 2, 1 (0.21 sec)	1	30.92
	2	75.06
6, 2, 1 (0.09 sec)	1	58.78
	2	19.29
1, 2, 2 (0.23 sec)	1	64.57
	2	7.7
2, 2, 2 (0.67 sec)	1	67.07
	2	2.7
3, 2, 2 (0.09 sec)	1	63.09
	2	10.66
4, 2, 2 (0.12 sec)	1	63.09
	2	10.66
5, 2, 2 (0.08 sec)	1	63.04
	2	10.76
6, 2, 2 (0.22 sec)	1	63.04
	2	10.76
1, 3, 1 (0.21 sec)	1	65.15
	2	6.54
2, 3, 1 (0.28 sec)	1	65.42
	2	6
3, 3, 1 (0.17 sec)	1	65.5
	2	5.84
4, 3, 1 (0.10 sec)	1	62.76
	2	11.33
5, 3, 1 (0.08 sec)	1	63.31
	2	10.22
6, 3, 1 (0.11 sec)	1	66.89
	2	3.06
1, 3, 2 (0.21 sec)	1	63.85
	2	9.14
2, 3, 2 (0.31 sec)	1	63.52
	2	9.8
3, 3, 2 (0.31 sec)	1	63.52
	2	9.8
4, 3, 2 (0.12 sec)	1	63.36
	2	10.12
5, 3, 2 (0.04 sec)	1	63.42
	2	10.01
6, 3, 2 (0.06 sec)	1	64.98
	2	6.88

Note: *c*=container rotation type; *u* = ranking function; *v* = priority rule.
Total CPU time = 9.65 seconds
The best solution found is bold and underlined.

Table 6. Comparison of Manual Solutions, Best Solution Found by Tree Search Heuritic, and Best Solution Found by Proposed Approach on Three Test Problems

Cont. No.	Manual Solution		Best Solution by Tree Search Heuristic (Pisinger, 2002)		Best Solution Found by Modified Wall Building Based Compound Approach	
	Cont. Type	No. of Boxes Vol. of Boxes %Fill	Cont. Type	No. of Boxes Vol. of Boxes %Fill	Cont. Type	No. of Boxes Vol. of Boxes %Fill
Test Problem 1						
1	40 HQ'	82 44,440,856 cm^3 51.37%	40 HQ'	88 70,317,292 cm^3 81.30%	40 HQ'	162 75,962,452 cm^3 87.82%
2	40 HQ'	94 41,173,128 cm^3 47.60%	40 HQ'	135 59,085,608 cm^3 68.31%	40 HQ'	61 53,440,448 cm^3 61.78%
3	40 HQ'	47 43,788,916 cm^3 50.63%	-		-	
Total CPU Time	-		365.39 seconds		74.14 seconds	
Test Problem 2						
1	40'	39 39,838,464 cm^3 51.33%	40'	77 61,653,944 cm^3 79.44%	40'	98 62,670,144 cm^3 80.75%
2	40'	74 38,739,800 cm^3 49.92%	20'	36 16,454,200 cm^3 42.44%	20'	15 15,908,120 cm^3 41.03%
Total CPU Time	-		459.28 seconds		15.38 seconds	
Test Problem 3						
1	40'	85 41,508,600 cm^3 53.48%	40'	94 53,097,700 cm^3 68.42%	40'	94 53,097,700 cm^3 68.42%
2	20'	9 11,589,100 cm^3 29.89%	-		-	
Total CPU Time	-		30.79 seconds		9.65 seconds	

Note: Volume of 40HQ' Container = 86,495,364 cm^3
Volume of 40' Container = 77,608,854 cm^3
Volume of 20' Container = 38,772,594 cm^3

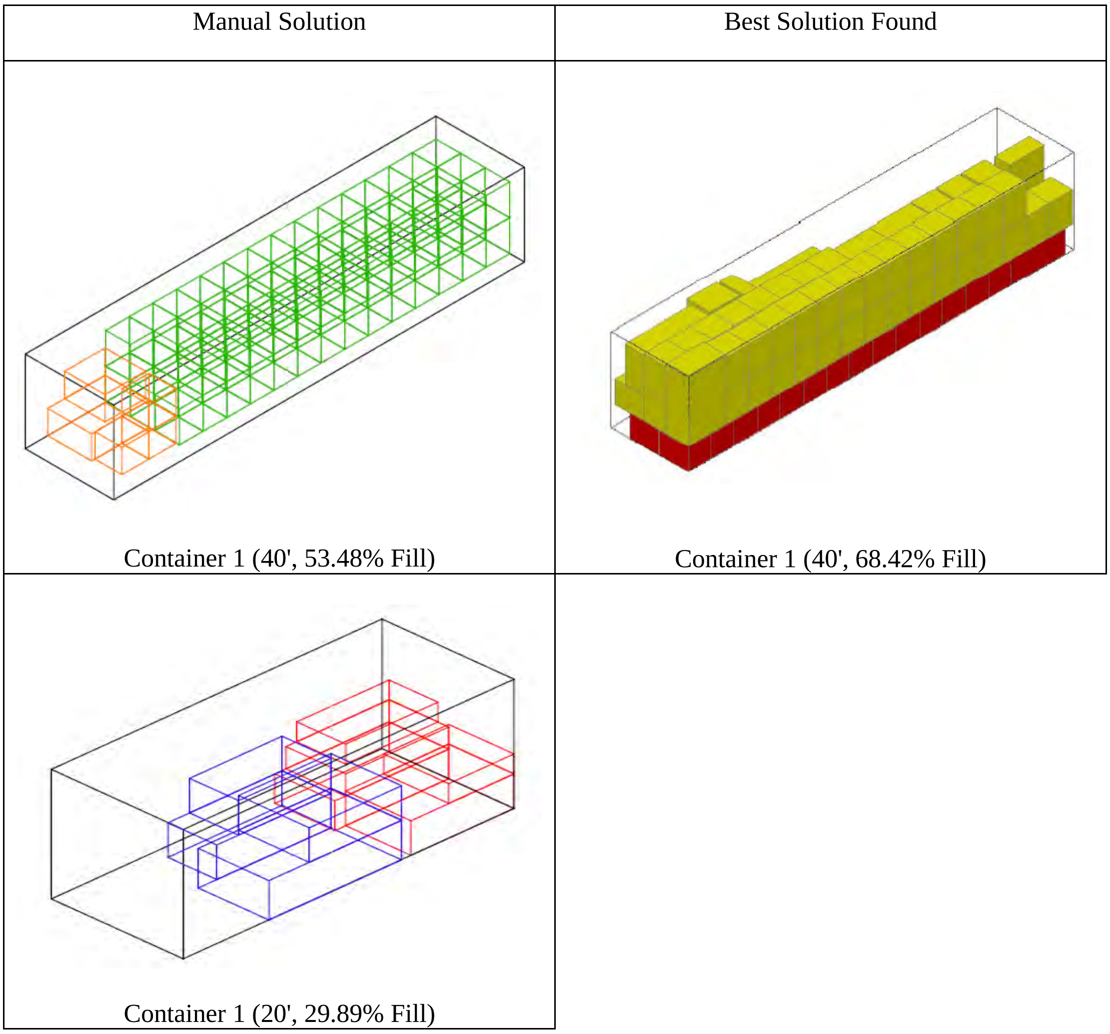


Figure 6. Comparison of Manual Solution and Best Solution Found on Test Problem 3

SUMMARY AND CONCLUSIONS

We consider the knapsack container loading problem where the rectangular-shaped cargo boxes with various sizes are to be packed in a given rectangular-shzped containers. The container is to be filled by selecting the best subset of boxes such that the total loaded volume is maximized. The boxes may be rotated in any orthogonal directions without load-related and positioning constraints. All boxes have the same origin-destination pair. The modified wall-building based compound approach performs 36 modified wall-building heuristics resulted from three existing ranking functions, two existing priority rules and six orthogonal rotations of container, while recording the best solution found. The modified wall building heuristics fill the container in a number of layers (wall) and fill the wall in a number of horizontal strips. The layer depths and strip heights are determined based on the dimensions of the remaining boxes, using three existing ranking functions*(2)* and two existing priority rules *(2)*. The existing box pairing procedure *(2)* is adopted in order to improve fill ratios of remaining boxes. The dynamic programming algorithm *(20)* is also adopted to exactly solve the 0-1 knapsack problem in order to fill each horizontal strip. We consider six orthogonal rotations of the container that correspond to the following: i) building walls across the container depth, and for each wall, building horizontal strips across the container height, ii) building walls across the container depth, and for each wall, building vertical strips across the container width, iii) building floors across the container height, and for each floor, building horizontal strips across the container width, iv) building floors across the container height, and for each floor, building horizontal strips across the container depth, v) building walls across the container width, and for each wall, building horizontal strips across the container height, and vi) building walls across the container width, and for each wall, building vertical strips across the container depth.

The proposed approach is performed on three real-world test problems, which are weakly heterogeneous with less than 20 box types, from a furniture company in Thailand. In the experiment, the proposed algorithms are employed to fill each container in the manual solutions in a descending order of the container size. When one container has a higher fill rate, it is more possible to use a smaller container (than that in the manual solutions) to pack the left boxes. There is not a winning heuristic that performs best on the three test problems. For the first test problem, the heuristic method with the container rotation type 6, ranking function f^2 and priority rule 2 performs best. For the second test problem, the heuristic method with the container rotation type 6, ranking function f^3 and priority rule 1 performs best. For the third test problem, three heuristics perform best. The typical wall building algorithms associated with the container rotation type 1 (c=1) do not perform well; thus, this shows significant improvement by considering the six orthogonal rotations of container in the modified approach.

The best solutions found on the three test problems are compared to the manual solutions by the furniture company. In terms of the number of containers, the proposed compound approach can save up to 33% on the three test problems. The highest fill percentages in the best solution founds are improved by up to 36% when compared with the manual solutions.

Moreover, the best solutions found by the proposed approach are compared to the best solutions found by the existing tree search heuristic. The proposed approach and the tree search heuristic yield the same results in terms of number of containers. However, the proposed approach outperforms the tree search heuristic in terms of solution quality and computational time. The best fill percentages by the proposed approach are up to 6% higher than those by the tree search heuristic. The total computational times by the proposed approach on the three test problems are up to 31% of those by the tree search heuristics. This paper may be extended to incorporate practical constraints such as multi-drop situations.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the support from The Thailand Research Fund under contract number MRG5380066, Office of the Higher Education Commission, Office of National Research Council of Thailand and King Mongkut's University of Technology Thonburi Research Fund.

REFERENCES

1. A. Bortfeldt and G. Wascher. Container Loading Problems-A State-of-the-Art Review, *Working Paper No. 7/2012, 2012*, Otto-von-Guericke Universitat Magdeburg.

2. D. Pisinger. Heuristics for the container loading problem. *European Journal of Operational Research*, Vol.141, 2002, 382-392.
3. P.C. Gilmore and R.E. Gomory. Multistage cutting stock problems of two and more dimensions, *Operations Research*, Vol.13, 1965, 94-120.
4. H. Dyckhoff. A typology of cutting and packing problems, *European Journal of Operational Research*, Vol.44, 1990, 145-159.
5. G. Wascher, H. Haubner, and H. Schumann. An improved typology of cutting and packing problems, *European Journal of Operational Research*, Vol.183 (3), 2007, 1109-1130.
6. E.E. Bischoff and M.D. Marriott. A comparative evaluation of heuristics for container loading. *European Journal of Operational Research*, Vol.44, 1990, 267-276.
7. E.E. Bischoff and M.S.W. Ratcliff. Loading multiple pallets, *Journal of the Operational Research Society*, Vol.46, 1995, 1322-1336.
8. H. Gehring, K. Menscher, M. Meyer. A computer-based heuristic for packing pooled shipment containers, *European Journal of Operational Research*, Vol.44, 1990, 277-288.
9. S. Martello, D. Pisinger, D. Vigo. The three-dimensional bin packing problem, *Operations Research*, Vol.48, 2000, 256-267.
10. C.S. Chen, S.M. Lee, Q.S. Shen. An analytical model for the container loading problem, *European Journal of Operational Research*, Vol.80, 1995, 68-76.
11. S.G. Christensen and D.M. Rousoe. Container loading with multi-drop constraints, *Thesis, Technical University of Denmark*, Kongens Lyngby, Denmark, 2007.
12. J.A. George, D.F. Robinson. A heuristic for packing boxes into a container. *Computers and Operations Research*, Vol.7, 1980, 147-156.
13. M. Eley. Solving container loading problems by block arrangement. *European Journal of Operational Research*, Vol.141, 2002, 393-409.
14. R. Morabito and M. Arenales. An AND/OR-graph approach to the container loading problem, *International Transactions in Operations Research*, Vol.1, 1994, 59-73.
15. H. Gehring and A. Bortfeldt. A genetic algorithm for solving the container loading problem, *International Transactions in Operations Research*, Vol.4, 1997, 401-418.
16. A. Bortfeldt and H. Gehring. A hybrid genetic algorithm for the container loading problem, *European Journal of Operations Research*, Vol.131, 2001, 143-161.
17. A. Moura and J.O. Oliviera. A GRASP approach to the container-loading problem, *Transportation and Logistics*, 2005, 50-57.
18. A. Bortfeldt, H. Gehring, and D. Mack. A parallel tabu search algorithm for solving the container loading problem, *Parallel Computing*, Vol.29, 2003, 641-662.
19. D. Mack, A. Bortfeldt, and H. Gehring. A parallel hybrid local search algorithm for the container loading problem, *International Transactions in Operations Research*, Vol.11, 2004, 511-533.
20. D. Pisinger. A minimal algorithm for the 0-1 Knapsack Problem, *Operations Research*, Vol.45, 1997, 758-767.
21. Gary, M.R. and D.S. Johnson, *Computers and Intractability: A Guide to theory of NP-Completeness*, Freeman, San Francisco, 1979.
22. Papadimitriou, C.H. On the complexity of integer programming, *Journal of ACM*, Vol.28, 1981, 765-768.
23. D. Pisinger. *A tree-search heuristic C code, University of Copenhagen*, Denmark, 1998, http://www.diku.dk/~pisinger/codes.html.

24. D. Pisinger. *The minknap algorithm callable C code*, University of Copenhagen, Denmark, 1993, http://www.diku.dk/~pisinger/codes.html.
25. T. G. Crainic, G. Perboli and R. Tadei. TS2PACK: A two-level tabu search for the three-dimensional bin packing problem, European Journal of Operational Research, Vol. 195 (3), 2009, 744-760.
26. A. Lodi, D. Pisinger and D. Vigo. Heuristic algorithms for the three-dimensional bin packing problem, European Journal of Operational Research, Vol.141, 2002, 410-420.
27. D. Martello, D. Pisinger and D. Vigo. The three-dimensional bin packing problem, Operations Research, Vol.48(2), 2000, 256-267.
28. P. Thapatsuwan, P. Pongcharoen, C. Hicks and W. Chainate. Development of a stochastic optimization tool for solving the multiple container packing problems, International Journal of Production Economics, 2011, doi:10.1016/j.ijpe.2011.05.012.