European Journal of Operational Research 236 (2014) 461–472

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

The nuclear medicine production and delivery problem

Jongsung Lee [a], Byung-In Kim [a,*], Andrew L. Johnson [b,c], Kiho Lee [a]

[a] Department of Industrial and Management Engineering, Pohang University of Science and Technology (POSTECH), Pohang, Kyungbuk 790-784, Republic of Korea
[b] School of Business, Aalto University, Helsinki 00101, Finland
[c] Department of Industrial and Systems Engineering, Texas A&M University, College Station, TX 77382, USA

ARTICLE INFO

Article history:
Received 14 May 2013
Accepted 15 December 2013
Available online 24 December 2013

Keywords:
Routing
Nuclear medicine
Half-life
Production and delivery
Vehicle routing
Large neighborhood search

ABSTRACT

Half-life is a unique characteristic of radioactive substances used in a variety of medical treatments. Radioisotope F-18 used for diagnosing and monitoring many types of cancers has a half-life of 110 minutes. As such, it requires careful coordination of production and delivery by manufacturers and medical end-users. To model this critical production and delivery problem, we develop a mixed integer program and propose a variant of a large neighborhood search algorithm with various improvement algorithms. We conduct several computational experiments to demonstrate the effectiveness of the proposed approach. The method when applied in a case study shows that improvement in terms of both time and cost is possible in the production and delivery of F-18.

1. Introduction

To our knowledge, no previous work has considered the nuclear medicine production and delivery problems (NMPDP) posed by F-18, an isotope that requires scrupulous determination of manufacturing levels as well as accounting for its rapid deterioration. In this paper we develop a mixed integer programming (MIP) model and propose a variant of a large neighborhood search (LNS) algorithm with various improvement algorithms. We conduct several computational experiments to demonstrate the effectiveness of the proposed approach. We conclude that the approach will help manufacturers and the medical community produce and deliver F-18 considering both time and cost.

The remainder of this paper is organized as follows. The detailed problem is described in Section 2. A literature review is presented in Section 3. The MIP model and our proposed approach are presented in Sections 4 and 5. Our experimental results are shown in Section 6. A case study is presented in Section 7 and our concluding remarks are presented in Section 8.

2. Problem description

The description below is based on a manufacturer that produces only radioisotope F-18 that we observed. Physicians order nuclear medicine, such as F-18, based on a specified level of radiation according to their injection plans for the medicine. The manufacture gathers a set of customer orders a day in advance, each of which corresponds to a hospital and perhaps multiple patients, and schedules delivery vehicles to visit each hospital. Each customer has a requested quantity of F-18 and time window for delivery. The end of time window corresponds to the medicine's usage time. F-18's half-life, 110 minutes, is the elapsed time until the number of atoms reduces to the half of its initial state. To meet the specified radiation level, the quantity of medicine produced is determined based on the production time and the usage time accounting for F-18's half-life. The production quantity must be equal to $d_i 2^{(l_i - f_i)/HL}$, where d_i, l_i, f_i and HL correspond to demand radiation quantity, use time, production termination time of order i, and the half-life, respectively.

The F-18 manufacturer batches customer orders in a production run. The factory has multiple cyclotrons, each capable of multiple production runs in a day. In this paper, we divide a cyclotron's production capacity into production runs with fixed start and end times. Management typically schedules the cyclotrons to cope with F-18's unpredictable low yield, which results from the process's stochastic nature. For this reason the manufacturer we observed prefers preset production runs although they may cause delivery vehicles to wait or excessive use of cyclotrons. We assume that when a production run is used, a fixed production cost occurs regardless of the production quantity, and that if no customer orders are assigned to a production run, there is no production cost for the run.

In the problem, vehicles with different capacities, i.e., heterogeneous vehicles, are available for delivery from the plant to the

[*] Corresponding author. Tel.: +82 54 279 2371; fax: +82 54 279 2870.
E-mail address: bkim@postech.ac.kr (B.-I. Kim).

0377-2217/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ejor.2013.12.024

customer stops. Time windows in which the vehicles are available are given and multiple trips per vehicle are allowed. Special packaging[1] protects workers and the public from radiation, and maintains product integrity. Split deliveries are not allowed and a single vehicle services a customer order.

We summarize the assumptions in the model to characterize the manufacturer's operations:

- The manufacturer has K different machines (cyclotrons) with different capacity.
- Each machine has a fixed number of scheduled production runs with fixed start and end times.
- One or many production runs can be skipped when appropriate.
- Vehicles start and end at the manufacturing plant.
- Multiple trips are possible for each vehicle.
- Demand quantities and delivery times are specified for each customer order.
- A customer order is serviced by a single vehicle (split delivery is not allowed).
- A single vehicle can deliver a product produced in separate production runs.
- The end of a customer's time window is the medicine usage time. A vehicle arriving before the customer's earliest allowed time waits until the time window begins. Late arrival after the time window is prohibited.
- Production quantities are determined accounting for the half-life of F-18 and the duration between production and delivery times.

The NMPDP also presents two sub-problems. Considering only the production scheduling component, it becomes a special type of bin packing problem, i.e., the production runs are modeled as bins and the customer orders are items. Since the production runs have different capacities, this problem becomes a variable-sized bin packing problem. Further, the size (production quantity) of items (customer orders) changes depending on the bin (production run) used. Thus, the production scheduling problem is a special case of the variable-sized bin packing problem (Pisinger & Sigurd, 2005).

The routing component, i.e., fixing the production schedule, becomes a vehicle routing problem with time windows (VRPTW). While vehicles leave the depot (production facility) at any time in a typical VRPTW, however, the embedded VRPTW in the problem specifies the ready times for the customer orders. The embedded VRPTW problem is similar to the pickup and delivery problem with time windows (PDPTW) (Dumas, Desrosiers, & Soumis, 1991; Lee, Ferdinand, Kim, & Ko, 2010).

3. Literature review

The nuclear medicine delivery problem resembles some perishable food and newspaper delivery problems. This section briefly reviews these studies and research efforts.

Tarantilis and Kiranoudis (2001) consider fresh milk distribution for a diary company in Athens, Greece, and propose a modified version of the threshold-acceptance algorithm of Dueck and Scheuer (1990), which is a deterministic version of the simulated annealing algorithm. Tarantilis and Kiranoudis (2001) treat the problem as a heterogeneous fixed fleet vehicle routing problem and do not consider customer time windows. Entrup, Gunther, Van Beek, Grunow, and Seiler (2005) develop three MIP models for production scheduling problems in yogurt production. They explicitly consider shelf-life in their models but do not consider delivery. Hsu, Hung, and Li (2007) and Osvald and Stirn (2008), who study the delivery of temperature-sensitive perishable food products, consider customer time windows, product value deterioration along delivery time, and time-dependent travel time, and formulate the problem as a mathematical model. Hsu et al. (2007) additionally consider time-varying temperature during the delivery day and propose a simple insertion heuristic to demonstrate the importance of the consideration of product value deterioration and energy cost in perishable product delivery.

Chen, Hsueh, and Chang (2009) study a production scheduling and vehicle routing problem with time windows for perishable food products but late delivery is allowed with a penalty. Each retailer has a stochastic demand with a known probability density function. The authors assume a single production line and that all products on the same vehicle are produced continuously as a single batch created on a particular production line. They present an integer nonlinear programming model for the problem, and propose a decomposition-based solution approach which uses the Nelder–Mead simplex algorithm (Nelder & Mead, 1965) for the production scheduling sub-problem and insertion and improvement algorithms for the vehicle routing sub-problem.

Garcia and Lozano (2004) study ready-mix concrete delivery. They classify the problem as scheduling with fixed start and end times. Raw materials at a plant are loaded into a revolving drum mounted on a vehicle which immediately delivers the mix to the customer site. The plant can mix up to a specified number (capacity) of customer orders at the same time. Since the plant has limited capacity, not all customer orders can be served. The authors consider two scenarios, arbitrary customer value and uniform customer value. They show that the first case corresponds to the fixed job scheduling problem and can be solved by a minimum cost flow algorithm. For the second case, they propose an exact graph-based algorithm and a branch-and-bound heuristic algorithm.

Mantel and Fontein (1993) develop a nonlinear mathematical programming model for designing a distribution network and routing solution for a Dutch regional newspaper. They use a location–allocation heuristic for determining the number of distribution centers and their locations and the savings algorithm of Clarke and Wright (1964) for truck route generation.

Ree and Yoon (1996) propose a solution method between three main distribution centers and 250 local distribution centers in Korea. They use a generalized assignment problem algorithm to assign the main centers to local centers and develop a simulated annealing algorithm that allows split deliveries. Song, Lee, and Kim (2002) use a regret distance method to assign three printing plants to 400 local distribution agents, a sweep-based method for route generation, and an urgent route first rule for route dispatching as the solution algorithms.

Hurter and Van Buer (1996) and Van Buer, Woodruff, and Olson (1999), who study a newspaper production and delivery problem with a single press, propose a nonlinear mathematical programming model and compare the performance of various tabu searches and simulated annealing algorithms. They also demonstrate that allowing multiple trips for a vehicle reduces operating costs. Russell, Chiang, and Zepeda (2008) consider a newspaper delivery problem and use a simple sequencing method based on the distance of zones from a newspaper printing plant for synchronizing production and truck loading, and a parallel insertion and tabu search method for the delivery routing problem. They report

[1] International Atomic Energy Agency (IAEA) suggests Type A packaging, which seals F-18 with a fiberboard box, a wooden box, and a steel drum filled with supplementary radiation-blocking materials. (IAEA, 2012, Cyclotron produced radionuclides: Guidance on Facility Design and Production of [^{18}F]Fluorodeoxyglucose(FDG), IAEA Radioisotopes and Radiopharmaceuticals Series No. 3).

a significant reduction in the number of vehicles and travel times compared with the firm's current practice.

Bachouch, Guinet, and Hajri-Gabouj (2009) study drug delivery for a French firm providing homecare concerning both planned and on-call delivery. They develop a mathematical model to test four strategies of deliveries: scheduling a specified number of deliveries, starting a tour only if a defined distance is reached by the calculated tour, starting deliveries of each carrier when a fixed number of assigned deliveries are reached, and starting deliveries at fixed hours.

There is some literature on the integration of production and distribution planning. Chandra and Fisher (1994) conduct computational experiments and report a 3–20% reduction in total operating cost by coordinating production and distribution decisions. Chang and Lee (2003) show the difficulty in modeling and analyzing systems that integrate production scheduling and delivery. Investigating cases in which jobs require different amount of storage space during delivery, they prove the problems are NP-hard and propose simple heuristics with worst-case performance ratio analysis. Li, Vairaktarakis, and Lee (2005) extend the single machine scheduling problem of Chang and Lee (2003) to consider multiple customer delivery locations. They propose a dynamic programming approach considering only a single vehicle. Toth and Vigo (2002) study the vehicle routing problem (VRP) where all loads are ready to be delivered at time zero.

Averbakh (2010), who introduces a production and delivery problem in which jobs are released in real-time, develops the shortest remaining processing time (SRPT) based rule as the solution approach. Reviewing the scheduling problem literature considering same-time production and delivery decisions, Chen (2010) finds no research regarding production quantity adjustment according to production time.

As mentioned, no previous research has considered the NMPDP which combines the variable item size bin packing problem and the PDPTW. While the NMPDP is similar to the perishable food delivery problem, in that product value deteriorates quickly, the NMPDP prohibits late delivery whereas the perishable food delivery problem allows late delivery with a penalty. Although similar to newspaper delivery, i.e. the NMPDP also requires strict just-in-time delivery, production cannot occur in advance and production and delivery must be synchronized, NMPDP's novelty comes from the half-life of the nuclear medicine, which requires that production quantity be adjusted to meet the order quantity and that the timing of production be determined considering both production and delivery.

4. Mathematical model

This section introduces an MIP model for scheduling the production and delivery of radioisotope F-18 to closely match demand. This paper uses the following notation:

Indices

i, j customer stop index, $1, 2, \dots, n$; 0 = plant index (start location); $n + 1$ = plant index (final location)
v vehicle index, $1, 2, \dots, V$
h vehicle tours index, $1, 2, \dots, H$
k machine index, $1, 2, \dots, K$
p production run index, $1, 2, \dots, P$

Parameters

d_i customer i demanded quantity
$[e_i, l_i]$ time window of customer i; l_i = the exact time the medicine is used
c_k production capacity of each production run on machine k
f_{pk} completion time of production run p on machine k
b_{ipk} production quantity for customer i when it is produced in production run p on machine k; can be calculated by $d_i 2^{\frac{l_i - f_{pk}}{110}}$
t_{ij} travel time from location i to location j
s_i service time at location i
O_v capacity of vehicle v
R_k production cost of each run on machine k
F fixed cost for each vehicle
G travel cost per unit distance
M a sufficiently large number

Decision variables

z_{ipk} 1, if customer i's demand is satisfied from production run p on machine k; 0, otherwise
q_{pk} 1, if production run p on machine k is used; 0, otherwise
y_{ivh} 1, if customer i is visited by the hth tour of vehicle v; 0, otherwise
x_{ij} 1, if customer i is immediately followed by customer j during a vehicle's tour; 0, otherwise
w_{vh} departure time for vehicle v from production facility for tour h
a_i arrival time at customer i
g_i cumulative quantity of delivered products after a vehicle visits customer i

We construct the MIP model as follows:

$$\text{Min} \sum_{p=1}^{P}\sum_{k=1}^{K} R_k q_{pk} + F\sum_{i=1}^{n} x_{0i} + G\sum_{i=0}^{n}\sum_{j=1}^{n+1} x_{ij} t_{ij} \tag{1}$$

s.t.

$$\sum_{p=1}^{P}\sum_{k=1}^{K} z_{ipk} = 1, \ \forall i = 1,\dots,n \tag{2}$$

$$\sum_{i=1}^{n} b_{ipk} z_{ipk} \leqslant c_k, \quad \forall k = 1,\dots,K, \ p = 1,\dots,P \tag{3}$$

$$f_{pk} \leqslant w_{vh} + M(2 - z_{ipk} - y_{ivh}), \quad \forall i = 1,\dots,n, \\ k = 1,\dots,K, \ h = 1,\dots,H, \ p = 1,\dots,P, \ v = 1,\dots,V \tag{4}$$

$$w_{vh} + t_{oi} \leqslant a_i + M(2 - x_{0i} - y_{ivh}), \quad \forall i = 1,\dots,n, \\ v = 1,\dots,V, \ h = 1,\dots,H \tag{5}$$

$$a_i + s_i + t_{ij} \leqslant a_j + M(1 - x_{ij}), \quad \forall i = 0,\dots,n, \ j = 1,\dots,n \tag{6}$$

$$a_j + s_j + t_{j,n+1} \leqslant w_{v,h+1} + M(2 - x_{j,n+1} - y_{jvh}), \quad \forall j = 1,\dots,n, \\ h = 1,\dots,H, \ v = 1,\dots,V \tag{7}$$

$$e_i \leqslant a_i \leqslant l_i, \ \forall i = 1,\dots,n \tag{8}$$

$$z_{ipk} \leqslant q_{pk}, \quad \forall i = 1,\dots,n, \ k = 1,\dots,K, \ p = 1,\dots,P \tag{9}$$

$$\sum_{i=0, i\neq j}^{n} x_{ij} = 1, \ \forall j = 1,\dots,n+1 \tag{10}$$

$$\sum_{j=1, i\neq j}^{n+1} x_{ij} = 1, \ \forall i = 0,\dots,n \tag{11}$$

$$\sum_{v=1}^{V}\sum_{h=1}^{H} y_{ivh} = 1, \ \forall i = 1,\dots,n \tag{12}$$

$$y_{ivh} \geqslant x_{ij} + y_{jvh} - 1, \quad \forall i = 1,\dots,n, \ j = 1,\dots,n, \ h = 1,\dots,H, \\ v = 1,\dots,V \tag{13}$$

$$y_{jvh} \geqslant x_{ij} + y_{ivh} - 1, \quad \forall i = 1,\dots,n, \ j = 1,\dots,n, \ h = 1,\dots,H, \\ v = 1,\dots,V \tag{14}$$

$$\sum_{i=1}^{n} y_{ivh+1} \leqslant M\sum_{i=1}^{n} y_{ivh}, \quad \forall h=1,\ldots,H, \ v=1,\ldots,V \tag{15}$$

$$x_{0i}+x_{0j}+y_{ivh}+y_{jvh} \leqslant 3, \quad \forall i=1,\ldots,n, \ j=1,\ldots,n, \ i\neq j, \\ h=1,\ldots,H, \ v=1,\ldots,V \tag{16}$$

$$g_i+\sum_{p=1}^{P}\sum_{k=1}^{K} b_{jpk}z_{jpk} \leqslant g_j+M(1-x_{ij}), \quad \forall i=0,\ldots,n, \\ j=1,\ldots,n+1, \ i\neq j \tag{17}$$

$$\sum_{p=1}^{P}\sum_{k=1}^{K} b_{ipk}z_{ipk} \leqslant g_i \leqslant O_v+M\left(1-\sum_{h=1}^{H} y_{ivh}\right), \quad \forall i=1,\ldots,n, \\ v=1,\ldots,V \tag{18}$$

$$z_{ipk}\in\{0,1\}, \quad \forall i=1,\ldots,n, \ k=1,\ldots,K, \ p=1,\ldots,P \tag{19}$$

$$y_{ivh}\in\{0,1\}, \quad \forall i=1,\ldots,n, \ h=1,\ldots,H, \ v=1,\ldots,V \tag{20}$$

$$x_{ij}\in\{0,1\}, \quad \forall i=0,\ldots,n, \ j=1,\ldots,n+1 \tag{21}$$

$$q_{pk}\in\{0,1\}, \quad \forall k=1,\ldots,K, \ p=1,\ldots,P \tag{22}$$

The objective function (1) is to minimize the total cost including production costs, fixed vehicle costs and travel costs. The constraints are as follows:

(2) require that each customer request is assigned to a production run
(3) the production capacity constraints require that the production by each cyclotron for a particular production run does not exceed the machine's capacity.
(4) require that a tour starts only after the product, F-18, is manufactured. If a customer's request is assigned to production run p on machine k (z_{ipk}=1) and it is also assigned to vehicle v's hth tour (y_{ivh}=1), the vehicle's start time from the plant must occur after the production run's completion time.
(5) requires that the arrival time at the first customer stop in a tour occurs after the start time of the tour from the plant plus the travel time to the first stop.
(6) requires a relationship between the arrival times at consecutive stops in a tour.
(7) requires that a tour's start time is greater than or equal to the previous tour's arrival time at the plant.
(8) requires the time window restrictions at customer stops.
(9) requires that customers are assigned to a production run only if the production run is used.
(10) requires that customers are visited only once.
(11) requires that there is only one departure from each customer.
(12) requires that each customer is assigned to only one vehicle.
(13) require that if customer j follows customer i in a tour (x_{ij} = 1) and customer j is assigned to vehicle v (y_{ivh} = 1), then customer i is also assigned to vehicle v (y_{ivh} = 1).
(14) enforce that if x_{ij} = 1 and y_{ivh} = 1, then y_{jvh} = 1.
(15) ensure that a vehicle tour is activated only if the previous tour is also active constraints
(16) enforce that the first stop in a tour is the plant.
(17) and (18) ensure that the capacity of a vehicle used in a particular tour is not exceeded.
(19)–(22) are the binary variable constraints.

Since the embedded sub-problems are NP-hard, the integrated production and delivery problem is also an NP-hard problem. Therefore, exact solution algorithms are limited to small problem instances. For example we could only solve instances with less than 8 customers and 3 vehicles in one hour using CPLEX 11.1 (ILOG Inc., 2007). The next section describes our heuristic construction algorithms and an LNS-based improvement algorithm to solve larger problem instances.

5. Solution method

LNS, a state-of-the-art meta heuristics, has been shown to perform well on various combinatorial optimization problems including vehicle routing problems (Kim, Kim, & Park, 2012; Kim, Li, & Johnson, 2012; Pisinger & Ropke, 2010; Prescott-Gagnon, Desaulniers, & Rousseau, 2009; Ropke & Pisinger, 2006; Shaw, 1997; Wy & Kim, 2013). While local search heuristics attempt to improve solutions via incremental adjustments to current solutions, LNS perturbs a current solution significantly within a large neighborhood. Pisinger and Ropke (2007) present a detailed history of LNS and its relationship to a very large scale neighborhood search and a variable neighborhood search. Below, we present our five algorithms which are integrated together to solving the NMPDP. For each algorithm a brief summary is given followed by pseudocode for the algorithm and then an explanation of some of the more complicated steps in the algorithm.

5.1. Proposed algorithm

Algorithm 1 shown in Fig. 1 is the main algorithm which implements LNS search.

Line 1: The user-given parameters are number of iterations (I_{max}), maximum allowable computation time (T_{max}), the maximum proportion of deleting stops (p_{max}), and the number of maximum solutions (N_{sp}) in the solution pool S.

Lines 2–5: A local minimum solution is generated and inserted to the solution pool.

Lines 6–13: A solution in the solution pool is selected and improved according to the LNS scheme. Note that the function $f()$ returns the cost of a solution and the objective is to minimize the cost.

Line 14: The new solution replaces the worst solution if: (1) the solution pool is full, (2) the current solution is not present in the pool, and (3) the pool's worst solution is worse than the current solution.

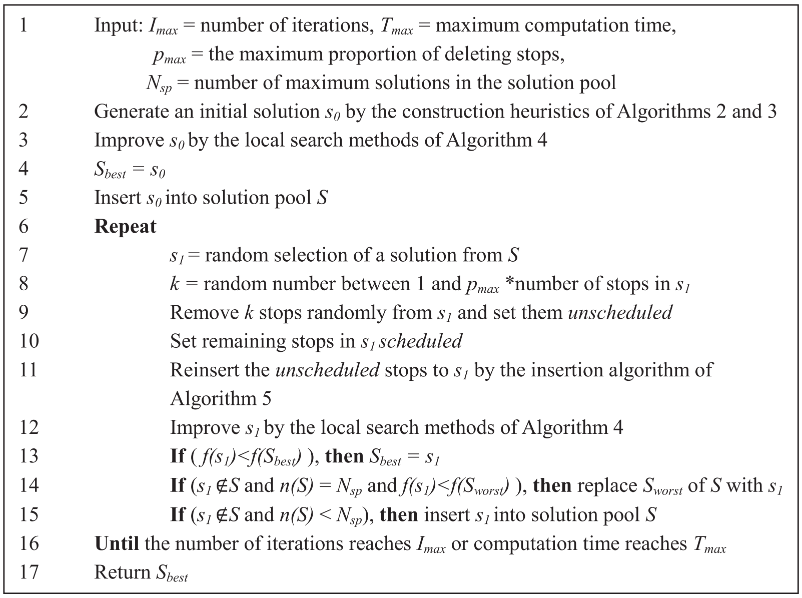
Line 15: the current solution is inserted into the solution pool if the solution pool is not yet full and does not contain the same solution.

5.2. Production run assignment algorithm

An initial solution is needed which we will incrementally improve with LNS with the additional algorithms we are proposing. The construction algorithm consists of two parts; production run assignment and route construction. Algorithm 2 shown in Fig. 2 provides a production run assignment method to develop a feasible solution.

Line 1: b_{ipk}, the production quantity for customer i when it is produced in production run p at machine k is calculated using the finish time of production run p at machine k (f_{pk}).

Line 2: The relative unit cost of each production run (RUCP) is calculated. The RUCP is calculated as the production run cost (R_k) divided by the relative production run capacity (RPC), which is calculated as the quantity produced and available at the end of a series of production runs. The outputs of early sub-production runs will be reduced due to deterioration between the time the subrun ends and the production run ends. An example of the command in line 2 is suppose there are three sub-runs A, B and C with a production capacity of 800 mCi. The time window of each production run is [0,10], [20,30], and [30,40], respectively and the half-life is 10. The production run ends at time 40. The RPC of the production run A is 100 mCi because 3 half-lives have elapsed between the end of sub-run A and the end of the run. In the same way, RPC of B and C is 400 mCi and 800 mCi, respectively.



```
Input: $I_{max}$ = number of iterations, $T_{max}$ = maximum computation time,
        $p_{max}$ = the maximum proportion of deleting stops,
       $N_{sp}$ = number of maximum solutions in the solution pool
Generate an initial solution $s_0$ by the construction heuristics of Algorithms 2 and 3
Improve $s_0$ by the local search methods of Algorithm 4
$S_{best} = s_0$
Insert $s_0$ into solution pool $S$
Repeat
        $s_1$ = random selection of a solution from $S$
        $k$ = random number between 1 and $p_{max}$ *number of stops in $s_1$
        Remove $k$ stops randomly from $s_1$ and set them unscheduled
        Set remaining stops in $s_1$ scheduled
        Reinsert the unscheduled stops to $s_1$ by the insertion algorithm of
        Algorithm 5
        Improve $s_1$ by the local search methods of Algorithm 4
        If ( $f(s_1)<f(S_{best})$ ), then $S_{best} = s_1$
        If ($s_1 \notin S$ and $n(S) = N_{sp}$ and $f(s_1)<f(S_{worst})$ ), then replace $S_{worst}$ of $S$ with $s_1$
        If ($s_1 \notin S$ and $n(S) < N_{sp}$), then insert $s_1$ into solution pool $S$
Until the number of iterations reaches $I_{max}$ or computation time reaches $T_{max}$
Return $S_{best}$
```

Fig. 1. Algorithm 1. Proposed computational algorithm.

Line 3: The production runs are sorted in ascending order of RUCP. Note that if all the production run costs are the same, then the production runs are sorted in descending order of their finish times.

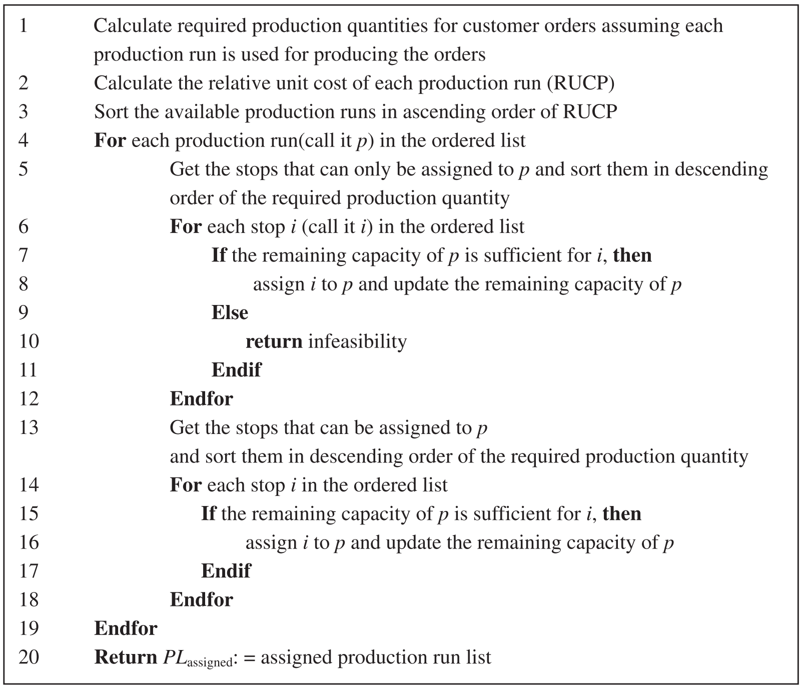
Lines 4–19: Then the production runs in the list are filled with the production quantities determined serially by the requirements of the customer requests.

Lines 5–12: For each production run *p*, if there are customers for which there is only one feasible production run, *p*, the customers are assigned to the run. If some customer orders can only be assigned to a particular production run *p* are not assigned to that run, then algorithm 2 returns infeasibility and stops. Depending on the problem instances, it may be difficult to find a feasible solution or there may be infeasible cases. However, in practice the capacities of production runs are sufficient to fulfill customer demands. The remaining feasible customer orders manufactured in production run *p* are sorted in descending order of the required production quantity. With this sorting method, we intend to pack the customer orders requiring large quantities first.

Lines 15 and 16: Customer orders from the sorted list are assigned one at a time in sequence to the current production run if feasible. The procedures are repeated for all production runs.

5.3. Route construction algorithm

Having assigned customer orders to production runs, the previous algorithm determines their ready times to departure from the plant. Algorithm 3 shown in Fig. 3 provides a route construction



```
1   Calculate required production quantities for customer orders assuming each
    production run is used for producing the orders
2   Calculate the relative unit cost of each production run (RUCP)
3   Sort the available production runs in ascending order of RUCP
4   For each production run(call it $p$) in the ordered list
5           Get the stops that can only be assigned to $p$ and sort them in descending
            order of the required production quantity
6           For each stop $i$ (call it $i$) in the ordered list
7                If the remaining capacity of $p$ is sufficient for $i$, then
8                     assign $i$ to $p$ and update the remaining capacity of $p$
9                Else
10                   return infeasibility
11               Endif
12          Endfor
13          Get the stops that can be assigned to $p$
            and sort them in descending order of the required production quantity
14          For each stop $i$ in the ordered list
15              If the remaining capacity of $p$ is sufficient for $i$, then
16                   assign $i$ to $p$ and update the remaining capacity of $p$
17              Endif
18          Endfor
19  Endfor
20  Return $PL_{assigned}$: = assigned production run list
```

Fig. 2. Algorithm 2. Production run assignment algorithm.

```
1   Repeat
2       Select the unscheduled stop $i$ that has the smallest ($l_i$- travel time from the
        plant to the stop)
3       Select a new vehicle $v$ to serve stop $i$
4       Insert stop $i$ into the route of vehicle $v$ and set stop $i$ scheduled
5       Repeat
6           Select the feasible stop $j$ that incurs the least increment to the route time
            of $v$
7           Insert stop $j$ into the route of vehicle $v$ and set stop $j$ scheduled
8       Until no more unscheduled stops can be inserted into the route of $v$
9   Until no unscheduled stops remain
10  For each vehicle $v$ in the routed vehicle list
11      For each vehicle $w \neq v$ in the routed vehicle list
12          If $c_w \leq c_v$ then
13              $ST(v)$:= start routing time of vehicle $v$
14              $FT(v)$:=end routing time of vehicle $v$
15              If $FT(v) \leq ST(w)$ then
16                  Merge all the trips of $w$ to the back of $v$
17              Else if $FT(w) \leq ST(v)$ then
18                  Merge all the trips of $w$ to the front of $v$
19              Endif
20          Endif
21      Endfor
22  Endfor
23  Return $VL_{routed}$:= routed vehicle list
```

Fig. 3. Algorithm 3. Route construction algorithm.

algorithm. When an order can be inserted into the route, all the possible insertion positions within the route are tested and the position that incurs the least route time increase is selected. At each iteration, among all the orders that can be inserted into the route, the order that incurs the least route time increase is selected and inserted. When there are multiple orders that increase the route time the same amount, the order having the least index is selected. Note that each vehicle has at most one trip and multiple trips are not considered at this stage. The procedures are repeated until all orders are routed.

Line 2: When a customer order is assigned to a vehicle, the departure time of the vehicle cannot be earlier than the ready time of the stop. When *unscheduled* customer orders remain, the order that has the smallest value for the difference (calculated as the exact time that the medicine is used at stop i (l_i) – travel time from the plant to the stop) is selected.

Line 3: A new vehicle for the order is selected.

Line 4: The order is inserted into the route of the vehicle.

Line 6 to 7: Then the *unscheduled* orders that can be inserted into the route are assigned to the route one at a time until no more orders can be inserted.

Lines 10–22: After all the orders are assigned to vehicles, the algorithm attempts to merge trips to reduce the number of used vehicles. Note that all the vehicles have only one trip. The merging procedure is conducted between two vehicles, v and w. Algorithm 3 selects the vehicle v and w sequentially from the routed vehicle list.

Lines 12–20: If the vehicle capacity of v is greater than or equal to w, the trips of w are tested to be moved to the other vehicle. The whole trips of vehicle w are inserted into vehicle v if there is no overlap between the trips. The testing of merge pairs of vehicles continues until all pairs are tested.

The route reduction heuristic (lines 10–22) can be replaced by an optimal procedure for cases in which the fleet is homogeneous. When the start and end time of each trip is fixed and the fleet is homogeneous, finding the minimum number of vehicles to cover the trips can be modeled as an assignment problem and can be solved in a polynomial time. To better understand the optimal procedure, we will describe an example. Let ST_i and ET_i be the fixed start and end time of trip i. Let α_{ij} be the cost of performing trip i and j with the same vehicle. Then, α_{ij} is set to the travel time (d_{ij}) from i to j if trip j can follow trip i, i.e., trips i and j can be covered by a single vehicle without violating the time constraint, i.e., $ET_i + d_{ij} \leqslant ST_j$. Note that since each trip starts and ends at the plant, α_{ij} becomes 0 if the traversal of i to j is feasible. If trip j cannot follow trip i because of timing constraint, α_{ij} is set to $M + 1$, where M is a big number. Finally, α_{ii} is set to M.

Then the route reduction problem can be modeled as an assignment problem as below. The decision variable u_{ij} is 1 if trip i is followed by trip j on the same vehicle in the optimal solution; 0, otherwise. Let T be the number of trips.

$$\min \sum_{i=1}^{T}\sum_{j=1}^{T}\alpha_{ij}u_{ij}, \tag{23}$$

s.t.

$$\sum_{i=1}^{T}u_{ij} = 1, \quad \forall\, j = 1,\ldots,T \tag{24}$$

$$\sum_{j=1}^{T}u_{ij} = 1, \quad \forall\, i = 1,\ldots,T \tag{25}$$

An optimum solution of the assignment problem denotes a consecutive enumeration of trips which can be covered by vehicles and is easily translated to a solution of NMPDP. If $u_{ij} = 1$ and $\alpha_{ij} = 0$, it means trip j can be placed in a vehicle immediately after trip i. If $u_{ii} = 1$, a vehicle serves trip i only. All the assignments i to j of cost $\alpha_{ij} \leqslant M$ are considered for the translation. For example, when $T = 6$, and the optimal solution is $u_{12} = u_{25} = u_{46} = u_{33} = u_{51} = u_{64} = 1$ where α_{12}, α_{25}, α_{46}, $\alpha_{33} \leqslant M$, then trips 1–2–5, 3, and 4–6 can be grouped and served by three vehicles. More details about the procedure are described in Kim, Kim, et al. (2012) and Kim, Li, et al. (2012).

5.4. Local route improvement algorithm

Fig. 4 shows our local search route improve algorithm. We use two route improvement methods, inter-route and intra-route improve, and a route reduction method. In the Route 1–0 improvement method, a stop from a route is moved into another route if it improves the solution quality. In the Route 1–1 improvement method, a stop from a route is exchanged with a stop from the other route. In the Production and Route 1–1 improvement method, a stop from a route and its corresponding production run is exchanged with a stop from the other route and production run. Note that when a stop from a route is exchanged with a stop from the other route, their positions may not be the same within the routes. In other words, two stops are removed from the two routes and inserted into other routes at the best positions. The intra-route improvement algorithm consists of the well-known 2-opt edge exchange method and Or-opt improvement method (Or, 1976).

In the route reduction algorithm (line 12), we attempt to reduce the number of routes by removing all the customer stops from a particular route and place them within other routes. All the customer stops in a routed vehicle are tested to be inserted to another vehicle and all the routed vehicles are tested. All the insertion positions are tested and the feasible position which gives the least routing cost increase is selected to move a stop from the origin vehicle. If it is possible to delete a route, i.e., all the stops in the route can be moved into other routes, we accept the movement. The improvement methods repeat until no further improvement is possible.

For example, assume that Veh #1, Veh #2 and Veh #3 cover stops A, B; C, D; and E, F, respectively. Algorithm 4 selects the vehicles one by one. If Veh #2 is selected, stops in Veh #2 are sequentially selected and inserted into other vehicles. First, stop C is tested to determine if it can be inserted between the plant and A, A and B, and B and the plant of Veh #1, and the plant and E, E and F, F and the plant of Veh #3. The tested insertion is feasible if the vehicle is able to load the medicine for the stop at the plant and can deliver it to the stop without preventing service of already existing stops. If there are multiple feasible inserting positions, stop C is inserted into the position where the cost increase is minimal. Similarly, stop D follows the same procedure. If both stops can be inserted into other routes, the reduction procedure is successful and Veh #2 is removed. Then the route reduction procedure starts again with the new solution. If it is not possible to move all of a vehicle's stops, the original solution is restored.

5.5. Insertion algorithm

Fig. 5 shows the insertion algorithm. Here, we reinsert the deleted customer stops from a solution in line 9 of Algorithm 1. The production schedule for the deleted and *unscheduled* customer stops can be generated by Algorithm 2. Then, for each *unscheduled* customer stop, the best vehicle route that incurs the least increment to its route time when the customer stop is inserted is selected and the stop is assigned to the route (lines 3–4). When combined with the deletion step, this insertion algorithm diversifies the solution search space. The insertion procedure is similar to the insertion procedure in algorithm 4. All feasible positions are tested and the best insertion position is selected. However, algorithm 5 considers adding a vehicle, which is not an option of the route reduction procedure in algorithm 4.

```
1   Input: $s_l$ = current solution
2   Repeat
3       For all combinations of the routes in $s_l$ (inter-route algorithm)
4           Do Route 1-0 improvement
5           Do Route 1-1 improvement
6           Do Production and Route 1-1 improvement
7       Endfor
8       For each route in the solution $s_l$ (intra-route algorithm)
9           Do 2-opt edge exchange improvement
10          Do Or-opt improvement
11      Endfor
12      Do route reduction improvement
13  Until the solution improves
14  Return improved $s_l$
```

Fig. 4. Algorithm 4. Local search route improvement algorithm.

```
1   Run Algorithm 2 for the production schedule generation of unscheduled stops
2   For each unscheduled stop $i$,
3       Select the vehicle $v$ including a new vehicle that incurs the least increment
        to its route time when stop $i$ can be feasibly inserted
4       Insert stop $i$ into the route of vehicle $v$ and set stop $i$ scheduled
5   Endfor
```

Fig. 5. Algorithm 5. Insertion algorithm.

6. Experimental results

We use the C++ language to implement the algorithms and an Intel Core(TM) i7 CPU 920 with 2.67 giga hertz, 6 gigabytes RAM running Windows 7 Enterprise for our eight scenarios. Because the previous research has not considered the NMPDP and there is no publicly available benchmark data, we develop 29 benchmark instances by extending the benchmark problem instances for the VRPTW (Solomon, 1987); we use the VRPTW benchmarks because VRPTW is an embedded sub-problem of the NMPDP and the optimal solutions for the benchmark problems are known. We use the same data set for the depot and the customer stops and requirements. We calculate the travel times between locations with one decimal point and truncation as suggested by Kohl and Madsen (1997). The vehicle capacity is $O_v = 200.0$ for any vehicle and for consistency with the VRPTW instances. Table 1 gives the production run data for case study.

We compare the results of our approach with those of Solomon's benchmark problem instances. For Scenarios 1–4, we use the production runs without considering their cost so that we can compare the total travel time and the number of vehicles of our approach with the known solutions of the VRPTW instances. Scenario 5 uses only the construction algorithm whereas Scenarios 6–8 use the LNS algorithm with various parameter settings. We compare the results of the construction algorithm and the LNS algorithm and bold the minimum cost for each problem instance in Table 3.

Table 2 shows the results of Scenarios 1–4. Columns 1 and 2 show the names of the VRPTW benchmark instances and the total demand quantities. For example, the total demand quantity of all of the customers in instance r101 is 1458. Columns 3 and 4 show the optimal solutions of the VRPTW instances; the values are taken from the literature and the VRPTW benchmark problem website (http://w.cba.neu.edu/~msolomon/problems.htm). Note that the optimal solutions for r108 and r112 have not been found yet. For those problems, we use the heuristic best known solutions obtained from the same website.

Columns 5–8 show the results of applying our proposed approach to the benchmark instances without considering F-18's half-life. Here, we assume zero production time for all customer orders such that production time does not limit the vehicles' start times. These scenarios are used to evaluate the vehicle routing solution quality of our approach. Scenario 1 shows the results with parameters $(I_{max}, T_{max}, p_{max}, N_{sp}) = (1500, 300\ \text{seconds}, 0.5, 20)$ and Scenario 2 $(4000, 600\ \text{seconds}, 0.5, 20)$. The proposed approach generates optimal solutions for several problems and near optimal solutions for the remaining problems. Note that the objective function for the optimal solution approach is minimizing the total travel time, but the objectives of our approach are to minimize the number of vehicles and the total travel time. Thus, our solution requires fewer vehicles in some instances. The travel distances presented in column 4 are lower bounds on the travel distances for all scenarios.

Table 1
Production run data.

Production run ID	Machine	Capacity	Cost	Start time	End time
1	1	500	350	−20	0
2	2	500	700	−15	5
3	1	500	350	5	25
4	2	500	700	10	20
5	1	500	350	30	50
6	2	500	700	35	55
7	1	500	350	55	75
8	2	500	700	60	80
9	1	500	350	80	100
10	2	500	700	85	105

Columns 11–16 show the results of applying the proposed approach to the benchmark instances considering both production and routing. Column 9 shows the half-life length of each problem instance and Column 10 shows the lower bound on the production quantity measured in mCi. To extend Solomon's VRPTW test instance, we merge the production run data specific to our case study with Solomon's instances. To assure feasibility, we adjust the half-life lengths to make the problems feasible based on preliminary experiments. We calculate the lower bound on the production quantity, column 10, as follows:

(1) Find the latest production run for each customer demand and calculate the required production quantity.
(2) Calculate the total required production quantity for each production run.
(3) From the latest production run to the earliest run: if the required quantity does not exceed capacity, add the quantity to the lower bound quantity; otherwise, add the capacity to the lower bound quantity and add the surplus quantity to the next production run considering the half-life inflation.

Note that the total production quantity with consideration for the half-life is about 2.75 times the quantity without half-life consideration (4504.7/1640.6).

Scenario 3 shows the results of the instances with half-life with parameters $(I_{max}, T_{max}, p_{max}, N_{sp}) = (1500, 300\ \text{seconds}, 0.5, 20)$ and Scenario 4 $(4000, 600\ \text{seconds}, 0.5, 20)$. When the half-life is considered, more vehicles are required because the delivery quantities increase, but vehicle capacity $(O_v = 200)$ is the same. Compared to the lower bounds on the production quantity, Scenario 3 and Scenario 4 generate solutions that require 5.21% ((4739.6 − 4504.7)/4504.7) and 5.17% ((4737.4 − 4504.7)/4504.7) more production on average, respectively. Although Scenarios 3 and 4 generate almost the same solutions in terms of the production quantities, Scenario 4 generates better solutions in terms of the number of vehicles and the total travel time. We first attempt to improve the routing solutions and then reduce the production levels. With more computation time, Scenario 4 generates better solutions than Scenario 3. These findings suggest that the proposed approach generates better solutions when more iterations are allowed. This is in contrast to many alternative heuristic approaches which cannot improve their solution quality beyond a certain point even with greater computation time.

Table 3 shows the results of comparing the construction algorithm with LNS. Scenario 5 is the result of the construction algorithm and Scenarios 6–8 are the results of the full algorithm utilizing LNS. The objective of Scenario 5–8 is to minimize total cost, i.e., the sum of production run cost, vehicle cost, and travel time cost. We use the data in Table 1 for the production run cost; $1000 as the vehicle cost, and $10 for a unit travel time. For example, the cost of the solution using production runs 1 and 2 and three vehicles with total travel time 250 miles is $350 + 700 + 3 * 1000 + 10 * 250 = \$6550$. Scenarios 6–8 show 25.5%, 28.4% and 29.9% improvement, respectively, compared with the construction algorithm only. Recall that we increased the values of I_{max} and T_{max} in Scenarios 6–8 to test the improvement rate of our approach. Table 3 shows that the improvement rate decreases as computation time increases, i.e., the improvement gap is 2.9% between Scenarios 6 and 7, but only 1.5% between Scenarios 7 and 8. The next section reports the results of applying our model and solution methodology to the original problem.

Table 2

Performance of the LNS method under different parameter settings for the NMPDP instances developed base on Solomon's VRPTW test problems.

Problem		VRPTW optimal solution		Without half-life				With half-life							
				Scenario #1 $(I_{max}, T_{max}, p_{max}, N_{sp})$ = (1500, 300 seconds, 0.5, 20)		Scenario #2 $(I_{max}, T_{max}, p_{max}, N_{sp})$ = (4000, 600 seconds, 0.5, 20)		Half-life	Prod. amt. lower bound	Scenario #3 $(I_{max}, T_{max}, p_{max}, N_{sp})$ = (1500, 300 seconds, 0.5, 20)			Scenario #4 $(I_{max}, T_{max}, p_{max}, N_{sp})$ = (4000, 600 seconds, 0.5, 20)		
Name	Demand	Veh#	Dist	Veh#	Dist	Veh#	Dist			Veh#	Dist.	Prod.	Veh#	Dist	Prod.
r101	1458	20	1637.7	19	1648.5	19	1645.7	40	3420.6	25	2022.2	4465.3	25	1844.6	4512.9
r102	1458	18	1466.6	17	1491.7	17	1480.1	55	4419.4	21	1565.8	4772.6	21	1512.3	4779.1
r103	1458	14	1208.7	14	1219.5	14	1216.7	65	4471.9	18	1822.3	4638.6	18	1818.1	4631.1
r104	1458	11	971.5	10	993.0	10	993.0	75	4297.9	19	1329.4	4505.2	18	1426.8	4497.1
r105	1458	15	1355.3	15	1366.6	15	1366.2	40	4307.5	23	1736.3	4784.8	23	1717.2	4796.2
r106	1458	13	1234.6	13	1265.5	13	1257.0	60	3992.6	19	1563.6	4619.6	18	1506.1	4507.7
r107	1458	11	1064.6	11	1073.8	11	1072.0	65	4711.9	18	1352.2	4846.2	18	1342.2	4836.9
r108	1458	9[a]	960.9	10	948.4	10	948.4	75	4382.2	19	1423.2	4525.3	19	1419.2	4502.5
r109	1458	13	1146.9	12	1152.6	12	1149.4	45	4855.5	23	1866	4938.9	22	1859.4	4963.0
r110	1458	12	1068	11	1106.8	11	1090.8	55	4276.2	21	1708.9	4675.3	21	1701.5	4682.7
r111	1458	12	1048.7	11	1064.4	11	1064.4	65	4066.5	19	1482.2	4596.9	19	1474.7	4596.9
r112	1458	9[a]	982.1	10	1000.8	10	969.8	65	4334.6	22	1669.4	4489.5	22	1669.4	4489.5
c101	1810	10	827.3	10	827.3	10	827.3	340	4727.5	16	1643.1	4766.2	16	1605.5	4764.8
c102	1810	10	827.3	10	827.3	10	827.3	450	4712.8	13	1823.9	4739.8	13	1823.9	4739.8
c103	1810	10	826.3	10	826.3	10	826.3	570	4810.0	12	1457.8	4819.3	12	1453.8	4819.3
c104	1810	10	822.9	9	1014.5	9	969.6	660	4778.1	12	1156.4	4785.3	12	936.1	4794.6
c105	1810	10	827.3	10	827.3	10	827.3	350	4886.7	17	2328.1	4897.6	16	2015.3	4892.5
c106	1810	10	827.3	10	827.3	10	827.3	370	4756.1	15	2236	4765.9	15	2222.9	4769.5
c107	1810	10	827.3	10	827.3	10	827.3	370	4877.0	15	1892.4	4887.1	14	1724.2	4884.5
c108	1810	10	827.3	10	827.3	10	827.3	390	4874.9	15	1969.3	4891.3	15	1667.1	4886.8
c109	1810	10	827.3	9	1223.9	9	1208.8	430	4891.9	15	1855.1	4898.4	15	1702.7	4899.6
rc101	1724	15	1619.8	15	1727.5	15	1640.9	50	4463.1	27	2497.5	4858.2	27	2485.4	4858.2
rc102	1724	14	1457.4	14	1481.2	13	1503.3	65	4370.0	24	2232.5	4746.5	23	2223.3	4746.9
rc103	1724	11	1258	12	1314.4	12	1295.3	75	4417.4	21	2244.1	4668.8	21	2239.2	4684.1
rc104	1724	10[b]	1132.3	10	1194.9	10	1188.2	85	4602.9	22	2015	4755.3	22	1965.1	4789.8
rc105	1724	15	1513.7	15	1557.0	15	1546.3	60	4370.3	25	2289.8	4745.3	25	2249.7	4753.8
rc106	1724	13[c]	1401.2	13	1472.7	13	1410.7	60	4416.3	25	2439.3	4757.5	25	2316	4781.6
rc107	1724	12	1207.8	11	1246.2	11	1246.2	70	4449.4	24	2282.2	4759.2	24	2239.7	4723.4
rc108	1724	11	1114.2	11	1149.2	11	1140.0	75	4693.9	25	2187	4848.2	25	2166.3	4799.0
Avg.	1640.6	12.0	1113.5	11.8	1155.3	11.8	1144.6	178.4	4504.7	19.7	1865.2	4739.6	19.4	1804.4	4737.4

The values presented in columns 3 and 4 are taken from the literature and the VRPTW benchmark problem website (http://w.cba.neu.edu/~msolomon/problems.htm). Note that the optimal solutions for r108 and r112 have not been found yet. For those problems, we use the heuristic best known solutions obtained from the same website.

[a] Heuristic best known solution.

[b] Irnich and Villeneuve (2006).

[c] Salani (2005).

Table 3
Cost comparison of construction algorithm with LNS using various parameter settings.

Problem		Construction algorithm without LNS Scenario #5			With LNS											
					Scenario #6 $(I_{max}, T_{max}, p_{max}, N_{sp}) = (4000, 600 \text{ seconds}, 0.5, 20)$				Scenario #7 $(I_{max}, T_{max}, p_{max}, N_{sp}) = (15{,}000, 1800 \text{ seconds}, 0.5, 20)$				Scenario #8 $(I_{max}, T_{max}, p_{max}, N_{sp}) = (30{,}000, 3600 \text{ seconds}, 0.5, 20)$			
Name	Demand	Veh#	Dist	Cost	Veh#	Dist	Cost	Save (%)	Veh#	Dist	Cost	Save (%)	Veh#	Dist	Cost	Save(%)
r101	1458	38	2524.6	68,496	26	1798.5	49,235	28.12	26	1666.4	47,914	30.05	25	1699.1	**47,241**	31.03
r102	1458	32	2276.5	60,015	22	1515.4	42,404	29.34	20	1605.5	41,305	31.18	20	1458.1	**39,831**	33.63
r103	1458	27	1991.1	52,161	20	1537.2	40,622	22.12	19	1475.8	**39,008**	25.22	19	1475.8	**39,008**	25.22
r104	1458	25	1883.5	48,385	19	1314.6	37,396	22.71	18	1304	36,290	25.00	18	1297.2	**36,222**	25.14
r105	1458	39	2785.3	71,403	23	1790.8	46,158	35.36	23	1684	45,090	36.85	23	1677.3	**45,023**	36.95
r106	1458	27	2057.9	52,129	19	1498.5	39,235	24.73	18	1493.7	38,187	26.75	18	1295	**36,200**	30.56
r107	1458	27	2057.2	52,822	19	1632.4	40,574	23.19	19	1398.9	38,239	27.61	19	1309.2	**37,342**	29.31
r108	1458	25	1988.6	49,436	20	1348.8	38,738	21.64	20	1348.1	38,731	21.65	19	1364.8	**37,898**	23.34
r109	1458	32	2408	61,330	24	1959.8	48,848	20.35	22	1765.2	44,902	26.79	22	1682.1	**44,071**	28.14
r110	1458	29	2171.6	55,266	21	1637.7	42,627	22.87	21	1618.7	42,437	23.21	20	1538	**40,630**	26.48
r111	1458	24	1862.9	47,179	19	1543	39,680	15.89	19	1539.1	39,641	15.98	19	1406.6	**38,316**	18.79
r112	1458	25	1812.4	47,674	23	1666.1	44,911	5.80	23	1605.3	44,303	7.07	23	1556.9	**43,819**	8.09
c101	1810	26	2163.1	52,881	17	1704.5	39,295	25.69	16	1246.9	33,719	36.24	16	1111.7	**32,367**	38.79
c102	1810	25	2109.6	51,346	16	1358.6	34,836	32.15	15	1111.2	31,362	38.92	15	1097.3	**31,223**	39.19
c103	1810	26	2117.7	52,427	16	1112.4	32,374	38.25	15	1027.1	30,521	41.78	14	920.7	**28,457**	45.72
c104	1810	26	2016.5	51,415	13	777.4	26,024	49.38	13	707.4	25,324	50.75	13	684.2	**25,092**	51.20
c105	1810	26	2131.9	52,569	19	1612.3	40,373	23.20	18	1430.5	37,555	28.56	18	1424	**37,490**	28.68
c106	1810	25	2411.5	54,365	18	1570.6	38,956	28.34	18	1412.5	37,375	31.25	16	1314.2	**34,392**	36.74
c107	1810	26	2352.6	54,776	16	1512.7	36,377	33.59	15	1494	35,190	35.76	15	1465.9	**34,909**	36.27
c108	1810	26	2113.5	52,385	17	1451.6	36,766	29.82	15	1371	33,960	35.17	15	1301.6	**33,266**	36.50
c109	1810	26	2085.1	52,101	16	1291.8	34,168	34.42	16	1069.5	31,945	38.69	14	1162.4	**30,874**	40.74
rc101	1724	40	3303.5	78,285	27	2431.5	56,565	27.74	26	2304.7	54,297	30.64	26	2294.9	**54,199**	30.77
rc102	1724	33	3036.9	67,919	23	2271.1	50,961	24.97	23	2117.5	49,425	27.23	23	2055.7	**48,807**	28.14
rc103	1724	30	2810	62,650	22	2063.6	47,886	23.57	22	1847.9	**45,729**	27.01	22	1847.9	**45,729**	27.01
rc104	1724	27	2413.1	56,381	22	1859.7	45,847	18.68	22	1852.2	45,772	18.82	21	1888.9	**45,139**	19.94
rc105	1724	32	2877.4	66,024	25	2361.7	53,867	18.41	25	2261	52,860	19.94	24	2283.6	**52,086**	21.11
rc106	1724	36	3336.5	73,915	25	2397.3	54,223	26.64	24	2342.7	52,677	28.73	24	2232.2	**51,572**	30.23
rc107	1724	29	2847.6	62,026	23	2107.5	49,325	20.48	23	2032.7	48,577	21.68	22	2077.3	**48,023**	22.58
rc108	1724	28	2547.8	58,728	25	2172.9	51,979	11.49	24	2062.5	49,875	15.07	24	2036.8	**49,618**	15.51
Avg.		28.9	2361.9	57465.1	20.5	1700	42767.2	25.5	19.9	1593	41110.7	28.4	19.6	1550.3	**40,305**	29.9

Minimum cost in boldface.

Table 4
Case study results.

Problem			Current practice			Proposed approach $(I_{max}, T_{max}, p_{max}, N) = (500, 300\text{s}, 0.5, 20)$			Proposed approach $(I_{max}, T_{max}, p_{max}, N) = (1000, 600\text{s}, 0.5, 20)$		
Stop#	Total demand	Prod. amt. lower bound	Production quantity	Vehicle number	Travel dist. (mile)	Production quantity	Vehicle number	Travel dist. (mile)	Production quantity	Vehicle number	Travel dist. (mile)
277	3672.1	42242.9	47707.8	24	10,925	47736.7	22	11,026	47608.1	21	11,009

7. Case study

This problem is motivated by a company producing and delivering radioisotope F-18.[2] The problem instance has 277 customer stops and a total demand of 3672.1 mCi. The lower bound on the production quantity calculated by the method stated in the previous Section is 42242.9 mCi, which is more than 11.5 times the total demand. This large ratio is partially due to the long distances between the plant and the customers. While the current practice requires 47707.8 mCi of production quantity, 24 vehicles and 10,925 miles of travel distance, the results of the proposed approach with parameters $(I_{max}, T_{max}, p_{max}, N_{sp}) = (500, 300\text{ seconds}, 0.5, 20)$ are 47736.7 mCi, 22 vehicles and 11,026 miles; with parameters (1000, 600 seconds, 0.5, 20), they are 47608.1 mCi, 21 vehicles and 11,009 miles. Although the travel distance increases slightly (101 miles and 84 miles, respectively), the solutions actually save two or three vehicles. Table 4 summarizes the results.

8. Conclusions

This paper described a nuclear medicine production and delivery problem (NMPDP) in which the production quantity of the medicine, F-18, for a customer varied depending on when it was produced due to a half-life of 110 minutes, and the production time determines the vehicle's earliest departure time from the plant to the customer. F-18's half-life, a unique characteristic, makes this NMPDP different from other perishable product delivery, newspaper delivery, and integrated production and delivery problems.

We developed a MIP model, presented a benchmark data set, and proposed a variant of a large neighborhood search algorithm with various improvement algorithms. The results showed that the proposed approach performed well on the benchmark instances. When practically applied, the proposed approach improved current practice by reducing the number of vehicles.

This research suggests some further investigations. An efficient exact algorithm for the NMPDP can be pursued. Although we developed an MIP model for the problem, it could not be solved optimally for practical size problems nor could it give meaningful tight bounds. Branch-and-price-and-cut algorithms based with several valid inequalities give quite good solutions for the VRP families (Chabrier, 2006; Desaulniers, 2010; Jin, Liu, & Eksioglu, 2008). A similar approach can be pursued for the NMPDP. An efficient rescheduling algorithm is also needed. F-18's unstable yield rate, which results in different production quantities versus expectations, suggests the possibility of reassigning the demands to the production runs and generating route solutions whenever a production run completes and the production quantity is different than expected. Moreover, a more elaborately designed bin packing algorithm could improve the solution since FFD does not guarantee feasible assignment for an instance of NMPDP which is solvable. Various bounds and relaxations also can be tested to resolve the production run assignment.

[2] The data used is permutated slightly for anonymity.

Acknowledgments

This work was supported by the National Research Foundation of Korea Grant funded by the Korean Government (No. 2012R1A1A2005243).

References

Averbakh, I. (2010). On-line integrated production–distribution scheduling problems with capacitated deliveries. *European Journal of Operational Research, 200*(2), 377–384.

Bachouch, R. B., Guinet, A., & Hajri-Gabouj, S. (2009). A model for scheduling drug deliveries in a French homecare. In *The proceedings of international conference on industrial engineering and systems management*, May 13–15. Montreal, Canada.

Chabrier, A. (2006). Vehicle routing problem with elementary shortest path based column generation. *Computers & Operations Research, 33*(10), 2972–2990.

Chandra, P., & Fisher, M. L. (1994). Coordination of production and distribution planning. *European Journal of Operational Research, 72*, 503–517.

Chang, Y.-C., & Lee, C.-Y. (2003). Machine scheduling with job delivery coordination. *European Journal of Operational Research, 158*, 470–487.

Chen, Z.-L. (2010). Integrated production and outbound distribution scheduling: Review and extensions. *Operations Research, 58*(1), 130–148.

Chen, H.-K., Hsueh, C.-F., & Chang, M.-S. (2009). Production scheduling and vehicle routing with time windows for perishable food products. *Computers and Operations Research, 36*, 2311–2319.

Clarke, G., & Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research, 12*, 568–581.

Desaulniers, G. (2010). Branch-and-price-and-cut for the split-delivery vehicle routing problem with time windows. *Operations Research, 58*(1), 179–192.

Dueck, G., & Scheuer, T. (1990). Threshold accepting: A general purpose optimization algorithm appearing superior to simulated annealing. *Journal of Computational Physics, 90*(1), 161–175.

Dumas, Y., Desrosiers, J., & Soumis, F. (1991). The pickup and delivery problem with time windows. *European Journal of Operational Research, 54*, 7–22.

Entrup, M. L., Gunther, H.-O., Van Beek, P., Grunow, M., & Seiler, T. (2005). Mixed-integer linear programming approaches to shelf-life-integrated planning and scheduling in yoghurt production. *International Journal of Production Research, 43*(23), 5071–5100.

Garcia, J. M., & Lozano, S. (2004). Production and vehicle scheduling for ready-mix operations. *Computers and Industrial Engineering, 46*, 803–816.

Hsu, C.-I., Hung, S.-F., & Li, H.-C. (2007). Vehicle routing problem with time-windows for perishable food delivery. *Journal of Food Engineering, 80*, 465–475.

Hurter, A. P., & Van Buer, M. G. (1996). The newspaper production/distribution problem. *Journal of Business Logistics, 17*(1), 85–107.

ILOG Inc. (2007). ILOG CPLEX 11.0 Documentation.

Irnich, S., & Villeneuve, D. (2006). The shortest-path problem with resource constraints and *k*-cycle elimination for $k \geqslant 3$. *INFORMS Journal on Computing, 18*(3), 391–406.

Jin, M., Liu, K., & Eksioglu, B. (2008). A column generation approach for the split delivery vehicle routing problem. *Operations Research Letters, 36*(2), 265–270.

Kim, B., Kim, S., & Park, J. (2012). A school bus scheduling problem. *European Journal of Operational Research, 218*, 577–585.

Kim, B., Li, H., & Johnson, A. L. (2012). An augmented large neighborhood search method for solving the team orienteering problem. *Expert Systems with Application, 40*(8), 3065–3072.

Kohl, N., & Madsen, O. B. G. (1997). An optimization algorithm for the vehicle routing problem with time windows based on Lagrangean relaxation. *Operations Research, 45*, 395–406.

Lee, H., Ferdinand, F., Kim, T., & Ko, C. (2010). A genetic algorithm based approach to the profitable tour problem with pick-up and delivery. *Industrial Engineering & Management Systems, 9*(2), 80–87.

Li, C.-L., Vairaktarakis, G., & Lee, C.-Y. (2005). Machine scheduling with deliveries to multiple customer locations. *European Journal of Operational Research, 164*, 39–51.

Mantel, R. J., & Fontein, M. (1993). A practical solution to a newspaper distribution problem. *International Journal of Production Economics, 30–31*, 591–599.

Nelder, J. A., & Mead, R. A. (1965). A simplex for function minimization. *Computer Journal, 7*, 308–313.

Or, I. (1976). *Traveling salesman-type combinatorial problems and their relation to the logistics of regional blood banking*. Ph.D. Thesis, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL.

Osvald, A., & Stirn, L. Z. (2008). A vehicle routing algorithm for the distribution of fresh vegetables and similar perishable food. *Journal of Food Engineering, 85*, 285–295.

Pisinger, D., & Ropke, S. (2007). A general heuristic for vehicle routing problems. *Computers and Operations Research, 34*, 2403–2435.

Pisinger, D., & Ropke, S. (2010). Large neighborhood search. In *Handbook of metaheuristics* (pp. 399–419). US: Springer.

Pisinger, D., & Sigurd, M. (2005). The two-dimensional bin packing problem with variable bin sizes and costs. *Discrete Optimization, 2*(2), 154–167.

Prescott-Gagnon, E., Desaulniers, G., & Rousseau, L. M. (2009). A branch-and-price-based large neighborhood search algorithm for the vehicle routing problem with time windows. *Networks, 54*(4), 190–204.

Ree, A., & Yoon, B. S. (1996). A two-stage heuristic approach for the newspaper delivery problem. *Computers and Industrial Engineering, 30*(3), 501–509.

Ropke, S., & Pisinger, D. (2006). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science, 40*(4), 455–472.

Russell, R., Chiang, W.-C., & Zepeda, D. (2008). Integrating multi-product production and distribution in newspaper logistics. *Computers and Operations Research, 35*, 1576–1588.

Salani, M. (2005). Improved dynamic programming for the vehicle routing problem with time windows. In *Proceedings of the 16th Mini – EURO conference and 10th meeting of EWGT* (pp. 559–564).

Shaw, P. (1997). *A new local search algorithm providing high quality solutions to vehicle routing problems*, Technical report, Department of Computer Science, University of Strathclyde, Scotland.

Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research, 35*(2), 254–265.

Song, S. H., Lee, K. S., & Kim, G. S. (2002). A practical approach to solving a newspaper logistics problem using a digital map. *Computers and Industrial Engineering, 43*, 315–330.

Tarantilis, C. D., & Kiranoudis, C. T. (2001). A meta-heuristic algorithm for the efficient distribution of perishable foods. *Journal of Food Engineering, 50*, 1–9.

Toth, P., & Vigo, D. (2002). *The vehicle routing problem*. Philadelphia, PA: SIAM.

Van Buer, M. G., Woodruff, D. L., & Olson, R. T. (1999). Solving the medium newspaper production/distribution problem. *European Journal of Operational Research, 115*, 237–253.

Wy, J., & Kim, B. (2013). A hybrid metaheuristic approach for the rollon–rolloff problem. *Computers and Operations Research, 40*(8), 1947–1952.